Getting Started	1
Common Confusions	1
Template Development	1
Template Uploading	2
Document Assembly	3
Server Assembly Behavior Differs from Desktop	4
Known Issues	5
Quick Help Links	6
Starting HotDocs	6
Automating a Template	6
Assembling a Document	7
What's New in HotDocs Developer?	7
New and Enhanced Features of HotDocs 11	7
New and Enhanced Features of HotDocs 10	11
New and Enhanced Features of HotDocs 2009	
New and Enhanced Features of HotDocs 2008	15
New and Enhanced Features of HotDocs 2007	17
New and Enhanced Features of HotDocs 2006	
New and Enhanced Features of HotDocs 2005	
New and Enhanced Features of HotDocs 6.2	
New and Enhanced Features of HotDocs 6.1	
New and Enhanced Features of HotDocs 6	
Installing HotDocs	

	Installing HotDocs: Quick Start	38
	Becoming Familiar with HotDocs	38
	HotDocs Overview	38
	HotDocs Developer Overview	46
	List HotDocs File Name Extensions	49
	Using the Keyboard in HotDocs	51
	Use the Keyboard to Work in HotDocs	51
	Full List of Keyboard Shortcuts	53
	Template Development Workflow	53
	HotDocs Templates Overview	53
	Plan Your HotDocs Project	55
	Mark Up the Document Set	56
	Prepare to Automate the Templates	57
	Automate Your Templates	58
	Organize the Final Template Library	61
	Distribute the Templates	61
	Conclusion	62
	Appendix 1: Variable Properties	62
	Appendix 2: Template Instructions	64
U	ing Libraries	65
	At a Glance: The HotDocs Library Window	65
	Introduction: Use HotDocs Libraries	65
	At a Glance: The New Library dialog box	68
	Creating a Library	69

To create a library	69
Open a Library	69
To open a different HotDocs library	69
At a Glance: The Add Item dialog box	70
Adding Templates and Other Files to a Library	71
Adding a Single Item to a Library	71
Adding Multiple Items at Once	72
Adding Miscellaneous Files	72
Work with Templates and Other Files in a Library	72
To use the library to access the templates and clause libraries	73
At a Glance: The Remove Item Dialog Box	73
Remove Items from a Library	74
To remove items from a library	75
At a Glance: The Item Properties dialog box	75
Change the Properties of a Single Library Item	76
To modify a library item	76
At a Glance: The Multiple Item Properties dialog box	77
Change the Properties for Multiple Items in a Library	78
To change the properties of two or more items in a library folder	78
At a Glance: The Folder Sort Options Dialog Box	79
Sort Items in a Library	80
To sort library items	80
Organize the Contents of a Library	81
To organize the contents of a HotDocs library	81

Create a Favorites folder in the Template Library	
To add templates to a Favorites folder in the library	
Create a New Library by Exporting Part of an Existing Library	
To export part of an existing library	
At a Glance: The Import Library Dialog Box	
Import One Library Into Another	
To import one library into another	
Print a List of Items in a Library	
To print a HotDocs library	
Save a Library	
Save the Contents of a Library as a Text File	
To save the library as a text file	
Move Items Within a Library	
To drag library items to a new location	
To cut and paste several library items	
To copy and paste several library items within the library	
Move Items to New Locations on Disk	
To move template and clause library files	
Copy Templates to New Locations	
To copy the template to a different location	
Assembling Documents	
Completing an Interview	
Overview: Assemble a Text or Form Document	
At a Glance: The Assembly Window	

Adjust HotDocs Windows and Panes	
Update Your Interview Outline and Document	
Preview a Template Before Assembling It	
Search for a Specific Template in a Library	
Assemble a Text or Form Document	
At a Glance: The Answer File dialog box	
Select an Answer File for Assembly	
Assemble an Interview Template	
Assemble a Document Without Showing an Interview	
Overlay Answers in the Current Answer File	
Assemble Multiple Documents	
At a Glance: The Assembly Queue Dialog Box	
Use the Assembly Queue	
Assemble a Clause Library from the Template Library	
Navigate Through Answer-Gathering Dialogs	
View Resources During Assembly	
At a Glance: The Save Answer File Dialog Box	110
Save an Answer File During Assembly	
Switch Answer Files During Assembly	112
Use Repeated Dialogs During Assembly	112
Use Inserted Dialogs During Assembly	
View Individual Rows in a Spreadsheet	
Select Clauses During an Interview	
Answer Date Questions	

	View an Answer's Variable Name	117
	Use the End of Interview Dialog	117
	Spell Check Answers	119
W	/orking with Assembled Text Documents	120
	Preview the Assembled Text Document	120
	Edit Answers at the Document Preview Tab	121
	At a Glance: The Document Text Editor	123
	Edit Document Text at the Document Preview Tab	125
	View an Assembled Document in Markup View	126
	Send the Assembled Text Document to the Word Processor	127
	Open a Saved Text Document for Editing	128
	Print an Assembled Text Document	129
	Save an Assembled Text or Form Document	129
	Save Assembled Text Document as PDF	130
	Save Assembled Form Document as PDF	130
	At a Glance: The PDF Security Settings Dialog Box	131
	Assign Security Options to a PDF Document	133
W	/orking with Assembled Form Documents	134
	Introduction: Assemble a Form Document	134
	At a Glance: HotDocs Filler	135
	Enter Answers Directly at the Form Document Tab	136
	At a Glance: The Field Inactive Warning Dialog Box	138
	View the Assembled Form Document	139
	Assemble a Fillable PDF Document	139

Override an Answer in a Form	141
At a Glance: The Print PDF-Based Documents Dialog Box	141
Print an Assembled Form Document	143
At a Glance: The Printing Properties Dialog Box	144
Choose Default Form Printing Options for a Form	145
Edit a Saved Form Document	146
Create a Form Field	147
Select a Field	148
Create a Check-Box Field	149
Detect Borders to Create or Resize a Field	150
Move a Field on a Form	151
Align Two or More Fields	152
Change the Borders and Margins of a Field	153
Copy One or More Form Fields	154
Resize Form Fields	155
Change the Tab Order of Fields	156
Group Form Fields So Answers Can Flow From One Field to Another	157
Ungroup Form Fields	159
Change the Field Type	159
Change the Font Used for a Field	160
Format Lines and Paragraphs of Text in a Form Field	161
Rotate Answers in a Field	162
Keep Contents of Form Fields from Printing	163
Cross Out Static Text on the Form	

	Circle Static Text on a Form	165
	Insert a Graphic File in a Form	165
	Convert Answers on the Form to Bar Code Format	167
	Flow a Single Answer Across Two or More Pages in a Form	169
	At a Glance: The Overflow Status Dialog Box	169
	Check Form Fields for Answer Overflow	171
	Customize the Look of the Addendum	172
	Define Overflow Properties for a Regular Field	174
	Split a Multi-Line Answer Between the Form and the Addendum	176
	Attach a Text or Form Document to an E-mail Message	177
(Creating Template and Document Summaries	177
	Introduction: Question and Answer Summaries	177
	View a Question Summary	178
	View an Answer Summary	179
	View the Variable Sheet	180
S	aving Answer Files	181
	Introduction: Save Answers	181
	Create a New Answer File	181
	Create a Default Answer File	182
	Suggest an Answer File for Every Assembly	183
	Save an Answer File During Assembly	183
	Switch Answer Files During Assembly	184
	At a Glance: The Send Answers to Mail Recipient Dialog Box	185
	Attach an Answer File to an E-mail Message	186

Using Answer File Manager	
Introduction: Use Answer Management	
At a Glance: The Answer File Manager	
Open and Close Answer File Manager	
Add an Answer File to Answer File Manager	
Work with Answer File Manager	
Add, Modify, and Delete Folders in Answer File Manager	
Using Published and Web-Based Files	
Introduction: Use Published Files on the Internet	
Register HotDocs to Work with Web Browsers	
Add an Auto-Assemble File on the Internet to a HotDocs Library	
Download Auto-Assemble Files for Use Offline	
At a Glance: The Install Template Set dialog box	
Install a Template Set	
At a Glance: The Update Template Sets dialog box	
Check for Template Set Updates	
Upload Answers	
Add a Web Page to a HotDocs Library	
Assembling Documents Linked to a Database	
Introduction: Assemble Documents Linked to a Database	
Select Records During Document Assembly	
Sort Database Records During Assembly	201
Filter Database Records During Assembly	
Save Selected Records in an Answer File	

Setting HotDocs Options	
Introduction: HotDocs Options	
Interviews and Dialogs	
At a Glance: Interviews and Dialogs (HotDocs Options)	
Display Repeated Dialogs Using a Special Icon	
Control How Child Dialogs Appear in the Interview Outline	
Control the Tab Order of Buttons on the Navigation Bar	
Select Existing Answers When Tabbing Between Answer Fields	
Display Warning When Trying to Add a New Repetition	
Control the Functionality of the Enter Key During the Interview	210
Display Resource Buttons Next to Answer Fields	210
Warn When HotDocs Reformats Date and Number Answers	
Specify How HotDocs Should Process the Date Order	
Control How HotDocs Handles Two-Digit and Four-Digit Years	
Change the Way Dates Appear in Answer Fields	
At a Glance: Appearance (HotDocs Options)	
Customize the Look of the Dialog Pane	
At a Glance: End of Interview (HotDocs Options)	
Customize the End of Interview Dialog	
Control What Happens When You Finish an Interview	
Document Assembly	
At a Glance: Document Assembly (HotDocs Options)	
Make Assembly Window Toolbars Automatically Wrap	
Display Warning When Questions are Unanswered	

Prompt to Save the Assembled Document When Closing the Assembly Window	221
Warn When Printing Assembled Text Documents	222
Prompt to Select an Answer File Before Assembly	222
Specify a Default Date Format	222
Format Unanswered Variables in a Document	223
Set Properties for Viewing Answers and Editable Text in the Assembled Document	224
Warn When Adding Assemblies to Assembly Queue	224
Include List of Completed Assemblies in Assembly Queue	225
Document Markup	225
At a Glance: Document Markup (HotDocs Options)	225
Specify How Documents Should Be Marked Up	227
At a Glance: Field Colors (HotDocs Options)	229
Define Field Colors	230
HotDocs Models	232
At a Glance: HotDocs Models (HotDocs Options)	232
Allow Non-Standard Names in HotDocs Markup	233
Form Documents	233
At a Glance: Form Documents (HotDocs Options)	233
Change the Units of Measurement for Forms	235
Change Colors in HotDocs Forms	236
Send Assembled PDF Documents to Adobe	236
Control How Forms Appear When Opened	236
Control Warnings During Assembly	237
Save Printer Information for Printing Forms	238

Specify How PDF-based Forms Should Be Printed	238
Protect Forms By Backing Them Up	239
At a Glance: Properties for New Fields (HotDocs Options)	240
Set Properties for New Edit Fields	241
Set Properties for New check box Fields	242
At a Glance: Overflow Properties (HotDocs Options)	243
Set Rules for Handling Answer Overflow	245
Specify Addendum References and Labels for Answer Overflow	246
At a Glance: Properties for New Forms (HotDocs Options)	246
Automatically Generate Default Interviews for Form Templates	248
Set Paper Size and Paper Source Defaults for Printing Forms	248
At a Glance: Addendum Properties (HotDocs Options)	249
Customize the Appearance of Addenda	250
Question and Answer Summaries	251
At a Glance: Question and Answer Summaries (HotDocs Options)	251
Change the Way Question and Answer Summaries Appear	253
Control Which Variables are Asked in a Question Summary	254
Personal Information	254
At a Glance: Personal Information (HotDocs Options)	254
At a Glance: The Add/Edit Personal Information Dialog Box	255
Create Personal Information Variables	256
Edit Answers for Built-In Personal Information Variables	257
Spell Checking	257
At a Glance: Spell Checking (HotDocs Options)	258

Change Your Spell Checking Options	259
Template Development	
At a Glance: Template Development (HotDocs Options)	
Have HotDocs Automatically Complete Component Names	
Have HotDocs Automatically Generate Dialog Element Component Names	
Choose Icon Style for Component Lists	
Control When Returns Are Inserted After Instructions in Text Templates	
Enable Document Editing Options for All New Templates	
Enable All New Templates for Use with HotDocs Server	
Display HotDocs Warning Messages	
At a Glance: Script Editor (HotDocs Options)	
Change Script Editing Options	
At a Glance: Template Manager (HotDocs Options)	270
Set Template Manager Options	271
Word Processors	272
At a Glance: Word Processors (HotDocs Options)	272
Install Support for New Word Processors	274
Change Your Default Word Processor	275
At a Glance: Word Processor File Locations (HotDocs Options)	275
Change Word Processor File Locations	276
File Locations	277
At a Glance: File Locations (HotDocs Options)	277
Change HotDocs Program File Locations	279
Assign Reference Paths to HotDocs Files	279

File Management	
At a Glance: File Management (HotDocs Options)	281
Manage Answer Files	
Template Set Updates	
At a Glance: Template Set Updates (HotDocs Options)	
Determine How Frequently HotDocs Checks for Template Set Updates	
Plugins	
At a Glance: Plugins (HotDocs Options)	
Set Options for HotDocs Plugins	
Automating Text Templates	
Creating and Working with Template Files	
Introduction: Template and Component Files	
HotDocs Editing and Navigation Toolbar Buttons	290
At a Glance: The New Template Dialog Box	293
Creating a New Text Template File	295
Creating a Text Template Based on an Existing Template	297
Converting a Single Template to a New File Format	298
Edit a Template	
Add HotDocs Toolbar to WordPerfect Templates	
At a Glance: The Apply Colors dialog box	
Assign Colors to Fields and Instructions in Templates	
Move Between Fields in a Template	
At a Glance: The Label Fields dialog box	
Use Labels to Identify Instructions	

	At a Glance: The Go To Field dialog box	305
	Go to a Specific Field in the Template	306
	View the Template in Markup View	306
	At a Glance: The HotDocs Outliner dialog box	307
	View an Outline of Scripting in the Template	309
	Create a Foreign Language Template	311
	Automating Plain Text Templates	313
	Tips on Using Graphics in Word Templates	313
	How Does HotDocs Assemble DOCX Templates?	316
lr	nserting Variables in Text Templates	316
	Introduction: Create and Customize Variables	316
	At a Glance: The Variable Field dialog box	317
	Insert a Variable Field in a Text Template	319
	At a Glance: The Find and Replace dialog box	321
	Replace Other Instances of Selected Text with a Variable	322
	Create and Insert a Variable Using Component Manager	322
lr	nserting Instructions in Text Templates	323
	Introduction: Types of Insert Instructions	323
	Understand How Smart Returns are Inserted After Instructions	324
	Use Labels to Identify Instructions	325
	Match Opening Instructions with Closing Instructions	327
U	sing IF Instructions (Make Parts of the Template Conditional)	328
	Introduction: Make Parts of Templates Conditional	328
	At a Glance: The IF Field dialog box (IF True/False Variable)	329

	At a Glance: The IF Field dialog box (IF Expression)	330
	Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression	333
	Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions	335
	Edit an IF Instruction or Expression	336
	Nest IF Instructions	337
	Use Conditional Instructions in a Computation Script	338
	Tips on Using IF Instructions	338
ι	Ising REPEAT Instructions (Create a List of Answers)	340
	Repeats and Lists Overview	340
	At a Glance: The REPEAT Field dialog box	344
	Creating a REPEAT Instruction to Gather a List of Answers	346
	Editing a REPEAT Instruction	347
	Choosing a Presentation Style for the Repeated Dialog	347
	Customizing REPEAT Titles	348
	Punctuating a Sentence-Style List	349
	Sorting a List of Answers	350
	Filtering a List of Answers	350
	Controlling How Many Rows are Visible in a Spreadsheet Dialog	351
	Limiting the Number of Answers Allowed for a Repeated Dialog	352
	Getting Sum Totals for Repeated Number Variables	352
	Repeats in a Word Processor Table Overview	353
	Using a Word Processor Table to Display a List	355
	Making a List Appear as one or More Columns	357
	Counting the Number of Entries in a List	358

	Retrieving a Specific Answer from a List	359
	Retrieving Information Other Than Answers from a REPEAT Instruction	359
U	sing INSERT Instructions	360
	Overview: Insert the Contents of One Template into Another's Assembly Process	360
	At a Glance: The INSERT Field dialog box	362
	Inserting an Existing Template's Contents into Another Template's Document Assembly Process	364
	Making Part of a Template into an Inserted Template	365
	Inserted Templates and Shared Component Files	365
	Editing an INSERT Template Instruction	366
	Header and Footer Override Scenarios	366
	Overriding Headers and Footers in Inserted Word Templates	376
	Insert Image Files into a Template	377
	Inserting Image Files into a Template	379
	Specify File Names for Inserting Files During Assembly	380
	Inserted Template File Locations	381
	Starting a New Assembly From a Template	382
	At a Glance: The Other Field Dialog Box	383
	Differences Between INSERT and ASSEMBLE Instructions	384
U	sing SPAN Instructions (Marking Editable Text in a Template)	385
	Introduction: Use SPAN Instructions	385
	At a Glance: The Edit SPAN Field dialog box	385
	Allow Users to Edit the Text of an Assembled Document	386
	Allow Users to Enter New Text in the Assembled Document	387
U	sing Clauses and Clause Libraries	388

	Introduction: Use Clauses and Clause Libraries	388
	At a Glance: The Clause Library	389
	Understand How Clauses Work	392
	Open and Close a Clause Library	393
	Creating a Clause Library at the HotDocs Library	393
	Add a Folder to a Clause Library	394
	At a Glance: The Edit Clause dialog box	395
	Create Clauses at the Clause Library	396
	At a Glance: The Add Clause dialog box	397
	Add Existing Clauses to a Clause Library	398
	Customize a Clause Library	398
	Create Clauses at the Template and Add Them to the Library	400
	Create and Insert Clauses Directly in Template	401
	Insert a Clause Library into a Template	402
	Edit an Existing Clause	403
	Group Clause Components in One Custom Dialog	404
Aut	omating Form Templates	405
C	reating and Working with Forms	405
	HotDocs Automator Overview	405
	At a Glance: The Automator Window	406
	HotDocs Automator Toolbar	408
	Create a Form Template	410
	Save Existing HFT Files as PDF-based Form Templates	411
	At a Glance: The Find Dialog Box	412

	Move Around in a Form	413
	Edit a Form Template	414
	At a Glance: The Insert Pages dialog box	414
	Add Pages to a Form Template	415
	At a Glance: The Delete Pages Dialog Box	416
	Remove Pages from a Form Template	417
	At a Glance: The Replace Static Content dialog box	417
	Update the Underlying Text in a Form Template	417
	View the Form in More than One Window	418
	Globally Change Properties in a Template or Group of Templates	419
	Save and Close a Form Template	420
	Define the Interview for a Form Template	421
C	reating Form Fields	422
	Create a Form Field	422
	Select a Field	423
	Create a check box Field	424
	Attach a Variable to a Field	425
	Attach an ASK Instruction to a Field	425
	Make a Field Conditional	426
	Attach an INSERT Instruction to a Field	427
	Detect Borders to Create or Resize a Field	428
	Move a Field on a Form	429
	At a Glance: The Align Fields dialog box	430
	Align Two or More Fields	430

	Change the Borders and Margins of a Field	431
	Copy One or More Form Fields	432
	Resize Form Fields	433
	Change the Tab Order of Fields	434
	Group Form Fields So Answers Can Flow From One Field to Another	435
	Attach a Variable to Single-Character Boxes	437
	Attach a Multiple Choice Variable to a Group of check boxes	438
	Ungroup Form Fields	439
	At a Glance: The Answer Wizard Field dialog box	439
	Attach an Answer Wizard to a Field	441
C	hanging Field Properties	442
	At a Glance: The Field Properties dialog box (Type Tab)	442
	At a Glance: The Field Properties Dialog Box (Layout Tab)	444
	At a Glance: The Field Properties dialog box (Overflow Tab)	446
	At a Glance: The Field Properties Dialog Box (Additional Tab)	447
	At a Glance: The Field Properties dialog box (Bar Code Tab)	449
	Preview the Formatting of Answers in a Form Field	450
	Show and Hide Fields	450
	Change the Field Type	451
	Change the Font Used for a Field	452
	Format Lines and Paragraphs of Text in a Form Field	453
	Rotate Answers in a Field	454
	Keep Contents of Form Fields from Printing	455
	Create a Field for Behind-the-Scenes Tasks	455

	Create a Resource Link on a Form	456
	Cross Out Static Text on the Form	457
	Circle Static Text on a Form	458
	Insert a Graphic File in a Form	459
	Convert Answers on the Form to Bar Code Format	460
	Merge a List of Answers in a Standard Table	462
	Merge a List of Answers in a Nonstandard Table	463
	Flow a Single Answer Across Two or More Pages in a Form	465
D	efining Form Overflow Properties	465
	Define Overflow Properties for a Regular Field	466
	Define Overflow Properties for Run-on Fields	467
	Define Overflow Properties for a Table	468
	At a Glance: The Addendum Properties dialog box	470
	Customize the Look of the Addendum	471
	Split a Multi-Line Answer Between the Form and the Addendum	473
	Create Independent Addendums for Forms	474
	Use Answer Overflow and Addendum Text Codes	475
С	reating and Using Fillable PDF Form Templates	476
	Introduction: Use Fillable PDFs as Form Templates	476
	Creating a Form Template Using a Fillable PDF	477
	At a Glance: The Check Fillable Fields dialog box	479
	Check Fillable Fields in a PDF Template	481
	Attach Variables to Fillable Fields	482
	Tips for Working with Fillable Fields	484

Understanding Variables	
Using Variable Types	
Introduction: Create and Customize Variables	
At a Glance: The Text Variable Editor	
Customize a Text Variable	
At a Glance: The Number Variable Editor	491
Customize a Number Variable	
At a Glance: The Date Variable Editor	
Customize a Date Variable	
At a Glance: The True/False Variable Editor	
Customize a True/False Variable	
At a Glance: The Multiple Choice Variable Editor	
Customize a Multiple Choice Variable	
At a Glance: The Computation Editor	
Customize a Computation Variable	
Insert a Personal Information Variable	
Create a Date Variable that Inserts the Current Date	
Merge the Name of the Answer File in the Document	
Use Automatic Paragraph Numbering	510
Editing Variables	510
Edit a Variable	510
Format the Variable	511
Create a Prompt for a Variable	513
At a Glance: The Advanced tab (Variable Editor)	514

	Control How HotDocs Processes a Variable	515
	Specify the Width of Answer Fields in the Interview	516
	Control How Answers Appear in the Assembled Document	517
	At a Glance: The Resource tab (Variable/Dialog Editor)	519
	Add Resource Information to a Variable or Dialog	520
	At a Glance: The Used In tab (Variable/Dialog Editor)	521
	View Relationship Between the Current Component and Other Components	522
	At a Glance: The Notes tab (Variable/Dialog Editor)	523
	Add Notes to Components	524
	Add a Comment to a Variable or Instruction Field	525
	Format an Answer as the Time of Day	526
	Number Variable Formats, Percentages, and Fractions	526
	Work with Multiple Choice Options	527
	Specify Merge Text Options as Default or Field-Specific Properties	528
	Copy and Paste Columns in a Multiple Choice Variable Spreadsheet	529
	Name a Group of Merge Text Values	530
	Work with Variables in Headers, Footers, Footnotes, and Text Boxes	530
	Use Pattern Codes to Create a Custom Pattern	531
	Use Variables and Scripts in Prompts, Dialog Element Text, and Resources	532
	At a Glance: The Go To Dialog Box	533
	Use the Script Editor	534
	At a Glance: The Locals tab (Variable/Dialog Editor)	536
Т	ïps on Using and Editing Variables	538
	Tips on Naming Your Variables	538

How Example Formats are Interpreted	540
Additional Date Formats	544
Understand the HotDocs Scripting Language	546
Understand How Component Titles and Prompts are Used	548
Using Dialogs	551
Creating and Using Dialogs	551
Introduction: Create Your Own Dialogs	551
At a Glance: The Dialog Editor (Properties Tab)	554
At a Glance: The Dialog Editor (Options Tab)	556
At a Glance: The Dialog Editor (Script Tab)	558
At a Glance: The Dialog Editor (Layout Tab)	560
Gather Questions into a Custom Dialog	561
Edit a Custom Dialog	564
Change a Dialog's Options	564
Add Resource Information to a Variable or Dialog	567
Change the Layout of Variables in a Dialog	568
Add Notes to Components	569
View Relationship Between the Current Component and Other Components	570
Use the Same Variable or Clause in Two or More Dialogs	570
Control Whether Dialogs are Asked Automatically	571
Inserting Dialog Elements in Dialogs	572
At a Glance: The Dialog Element Editor	572
Add Text to Your Dialogs	575
Add the Ability to Launch an Application from a Dialog	

Add a Link to a Dialog That Runs a Computation	577
Add a Hyperlink to a Dialog	578
Add Spacing and Separator Lines to a Dialog	579
Add a Graphic File to a Dialog	580
Using Scripts to Make Dialogs Dynamic	581
Use Scripts to Add Power to Your Dialogs	581
Special Instructions for Dialog Scripts	582
Assigning Answer Sources to Dialogs	583
Suggest an Answer Source for Dialogs	583
Tips for Working with Answer Sources	584
At a Glance: The Variable Mapping dialog box	585
Link a Dialog to a List Gathered by a Repeated Dialog	586
Link an Existing Answer Source to Other Dialogs	586
Let Users Retrieve Answers from an Outlook Contacts List	587
Let Users Retrieve Answers from a Time Matters Record	588
Inserting Dialogs into other Dialogs	589
Insert Dialogs Into Dialogs	589
Group Child Dialogs in a Parent Dialog	590
Using ASK Instructions to Control Dialog Order	590
At a Glance: The ASK Field dialog box	590
Control When Your Dialogs Appear	591
Put ASK Instructions in a Computation Variable	592
Use ASK Instructions to Control the Interview	593
Defining Interviews and Interview Templates	595

HotDocs Interviews Overview	595
Interviews Defined	
Choose an Interview Type	595
Understand the Interview Generation Workflow	595
Have HotDocs Disable Irrelevant Variables in Interviews	597
Interview Templates Defined	597
Generating a Default Interview	597
To generate a default interview for a template	598
Combine Default Dialogs	598
To generate combine default dialogs in an interview	598
Improve Interview Speed Using a Custom Interview	599
Generate a Default Interview	599
Generate a Custom Interview	600
Defining a Custom Interview Component	600
To create a custom interview component	600
Tell HotDocs to use the custom interview component when generating the interview	601
Disabling Irrelevant Variables in Interviews	601
To mark how HotDocs should treat variables if they're not needed in the document	602
To automatically exclude dialogs that contain only irrelevant variables	602
To automatically disable unused variables in the interview	602
Creating an Interview Template	603
To create an interview template	603
Using Lists of Answers	605
Using Computations to Create Lists	605

Create a REPEAT Instruction Using a Computation Variable	605
Punctuate a List Using a Computation Variable	606
Sort a List Using a Computation Variable	606
Filter a List Using a Computation Variable	607
Tips on Using Computations to Create Lists	608
Creating Lists within Lists	609
Introduction: Create Lists Within a List	609
Create a List Within a List	610
Display a Child Dialog Directly on the Parent Dialog	611
Nest REPEAT Instructions Using a Computation Variable	612
Use a Particular Answer from a Sublist	613
Managing Components and Templates	615
Using Component Manager	615
At a Glance: The Component Manager	615
Open and Close Component Manager	617
Open Component Manager at the Template Library	618
Use Component Manager to Work with Components	618
Create and Edit Multiple Components Simultaneously	621
At a Glance: The Component File Properties dialog box	622
At a Glance: Additional Templates and Additional Files (Component File Properties)	623
At a Glance: Interview Tab (Component File Properties)	625
At a Glance: Assembly Tab (Component File Properties)	627
At a Glance: Hidden Data Tab (Component File Properties)	629
At a Glance: HotDocs Server Tab (Component File Properties)	631

At a Glance: Answer Upload Tab (Component File Properties)	633
Change Component File Properties	635
Delete a Component from the Current Component File	636
Copy Components From One File to Another	636
Rename Components in a Single Template	638
At a Glance: The Duplicate Variables dialog Box	639
Make a Duplicate Copy of a Variable	641
Import or Export Computation Variables	641
Edit Formats, Merge Text, Dialog Elements, and Patterns	642
At a Glance: The Find In Components dialog box	643
At a Glance: The Find and Replace Components dialog box	644
Search Component Files	646
At a Glance: The Spell Check Components dialog box	647
Spell Check Components	649
At a Glance: The Print Components dialog box	650
Print a List of Components	651
Using One Component File for Multiple Templates	651
Insert a PLAY Instruction in a Template	
Specify Whether Component File Properties are Shared	654
Make Templates Stop Sharing Component Files	655
Control Whether Users See the End of Interview Dialog	
Position the Cursor in the Assembled Document	656
Specify a Word Template for Storing Post-Assembly Macros	657
At a Glance: The Hidden Data Remover dialog box	657

Remove Hidden Data from Assembled Documents	659
Using the Hidden Data Remover dialog box	660
Remove Hidden Data from Word Templates	662
Work with Variables in Headers, Footers, Footnotes, and Text Boxes	663
Using Template Manager	664
Introduction: Use Template Manager	664
At a Glance: The Template Manager	666
Open and Close Template Manager	668
Understand Parts of Template Manager Window	669
Template Manager Toolbar	671
Export Contents of a Template Manager Window to a Spreadsheet	673
At a Glance: The Convert Files dialog box	673
Convert Multiple Templates to Work with HotDocs 11	675
Convert Templates and Clauses to Microsoft DOCX	676
At a Glance: The Rename Files dialog box	678
Rename Templates Using Template Manager	678
At a Glance: The Copy Components dialog box	679
At a Glance: The Paste Components dialog box	680
Copy and Paste Components Across Multiple Component Files	681
At a Glance: The Rename Components dialog box	682
Rename Components Across Multiple Component Files	682
Delete Components from Multiple Component Files	683
Modify Component File Properties Across Multiple Files	684
Using Component Manager	685

Using HotDocs Scripting	685
Scripting Overview	685
Using Instruction Models	700
Using Expression Models	727
Setting Values for Variables	783
At a Glance: The Component Manager	786
Open and Close Component Manager	789
To open Component Manager	789
Open Component Manager at the Template Library	790
To open Component Manager at the template library	790
Use Component Manager to Work with Components	790
To work with individual components in a component file	790
Create and Edit Multiple Components Simultaneously	792
To edit multiple components simultaneously	793
At a Glance: The Component File Properties dialog box	793
At a Glance: Additional Templates and Additional Files (Component File Properties)	795
At a Glance: Interview Tab (Component File Properties)	797
At a Glance: Assembly Tab (Component File Properties)	799
At a Glance: HotDocs Server Tab (Component File Properties)	801
At a Glance: Hidden Data Tab (Component File Properties)	803
Change Component File Properties	805
To change component file properties	806
Delete a Component from the Current Component File	806
To permanently delete a variable, format, pattern, or merge text group	807

Copy Components From One File to Another	807
To copy components from one component file to another	807
Rename Components in a Single Template	808
To rename a component while editing the template	809
To rename a single component using Component Manager	809
To rename multiple components using Component Manager	809
At a Glance: The Duplicate Variables dialog Box	810
Make a Duplicate Copy of a Variable	811
To duplicate a single variable	811
To duplicate multiple variables at once	811
Import or Export Computation Variables	812
Export a Computation to the Clipboard	812
Import a Computation from the Clipboard	812
Edit Formats, Merge Text, Dialog Elements, and Patterns	813
To edit supplemental components	813
Using One Component File for Multiple Templates	813
To point the current component file to a shared component file	814
Insert a PLAY Instruction in a Template	815
To insert a PLAY instruction in a template	815
Specify Whether Component File Properties are Shared	816
To choose which component file properties should be used for a pointed template	816
Make Templates Stop Sharing Component Files	817
To unpoint a component file	817
Control Whether Users See the End of Interview Dialog	817

To hide the End of Interview dialog	818
Position the Cursor in the Assembled Document	818
To insert a TypeHere bookmark	818
Specify a Word Template for Storing Post-Assembly Macros	819
To use a post-assembly macro template	819
Remove Hidden Data from Assembled Documents	820
To choose which properties will be removed from an assembled Word document	820
Work with Variables in Headers, Footers, Footnotes, and Text Boxes	821
Using HotDocs Scripting	823
Scripting Overview	823
Introduction: Understand the HotDocs Scripting Language	823
Use Operators when Scripting	825
Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts	828
Use the Script Editor	830
Local Variables	832
Parameters	835
Using Instruction Models	838
Full List of Instruction Models	838
ADD TEXT TO MULT_CHOICE_VAR; CLEAR MULT_CHOICE_VAR	840
ASCEND VAR; DESCEND VAR	842
ASK DATABASE	842
ASK DIALOG	843
ASK VAR	843
ASSEMBLE "FILENAME"	

	CONCEAL VAR	846
	DEBUG	846
	DEFAULT VAR TO VALUE	847
	ERASE VAR; ERASE DIALOG	848
	FILTER COMPUTATION_VAR	849
	FORMAT "LIST_FORMAT"	850
	GRAY ALL; GRAY VAR; UNGRAY ALL; UNGRAY VAR	850
	HIDE ALL; HIDE VAR; SHOW ALL; SHOW VAR	851
	IF EXPRESSION; ELSE IF; ELSE; END IF	852
	INCREMENT NUM_VAR; DECREMENT NUM_VAR	854
	INSERT "FILENAME"	854
	LANGUAGE CODE	855
	LIMIT NUM	856
	OMIT VAR	857
	PLAY "MACRO"	858
	QUIT	859
	REPEAT DATABASE; END REPEAT	860
	REPEAT DIALOG; END REPEAT	860
	REQUIRE ALL; REQUIRE VAR	862
	SET VAR TO VALUE	862
	WHILE EXPRESSION; END WHILE	863
ι	Ising Expression Models	865
	Full List of Expression Models	866
	Enter a Date	871

Enter a Number	
Enter some Text	
Enter True or False	
ABSOLUTE VALUE(NUM)	
AGE(DATE)	
ANSWERED(DIALOG)	
ANSWERED(VAR)	
CEILING(NUM)	
COUNT(DIALOG)	
COUNT(MULT_CHOICE_VAR)	
COUNTER	
DATE - NUM DAYS	
DATE - NUM MONTHS	
DATE - NUM YEARS	
DATE + NUM DAYS	
DATE + NUM MONTHS	
DATE + NUM YEARS	
DATE OF(NUM, NUM, NUM)	
DAY OF(DATE)	
DAY OF WEEK(DATE)	
DAYS FROM(DATE, DATE)	
ERRORTEXT	
EXPONENTIAL(NUM)	
FIRST(TEXT, NUM)	
FLOOR(NUM)	
---------------------------------------------------------	--
FORMAT(VALUE, "EXAMPLE")	
FUTURE VALUE (Rate, Term, Payment, Present Value, Type)	
INTEGER(TEXT)	
LAST(TEXT, NUM)	
LENGTH(TEXT)	
LOGARITHM(NUM)	
MAX(NUM, NUM)	
MID(TEXT, NUM, NUM)	
MIN(NUM, NUM)	
MONTH OF(DATE)	
MONTHS FROM(DATE, DATE)	
MULT_CHOICE=TEXT; MULT_CHOICE!=TEXT	
NATURAL LOGARITHM(NUM)	
NOT TRUE_FALSE	
OTHER(MULT_CHOICE_VAR)	
PAYMENT (Rate, Term, Present Value, Future Value, Type)	
POSITION(TEXT, TEXT, TRUE_FALSE)	
POWER(NUM, NUM)	
PRESENT VALUE (Rate, Term, Payment, Future Value, Type)	
RATE (Term, Payment, Present Value, Future Value, Type)	
REMAINDER(NUM, NUM)	
REPLACE(TEXT, TEXT, NUM)	
RESULT	

	ROUND(NUM, NUM)	906
	SELECTION(MULT_CHOICE_VAR, NUM)	907
	SPACE(TEXT, TEXT)	908
	SQUARE ROOT(NUM)	909
	STRIP(TEXT, TEXT, TRUE_FALSE, TRUE_FALSE)	910
	SUM(COMPUTATION_VAR)	911
	SUM(NUM_VAR)	911
	TERM (Rate, Payment, Present Value, Future Value, Type)	912
	TEXT CONTAINS TEXT	913
	TEXT ENDS WITH TEXT	914
	TEXT STARTS WITH TEXT	915
	TODAY	915
	TRIM(TEXT)	916
	TRUNCATE(NUM, NUM)	916
	UNANSWERED	917
	UNION(MULT_CHOICE, MULT_CHOICE)	918
	VALUE(VAR, EXPRESSION)	919
	YEAR OF(DATE)	920
	YEARS FROM(DATE, DATE)	920
	ZERO(NUM_VAR)	921
S	etting Values for Variables	921
	Automatically Assign Answers to a Variable	921
	Values in SET Instructions	922
	Use SET Instructions to Help Write Computations	923

Differences Between SET and DEFAULT Instructions	
Linking Templates to a Database	
Introduction: Link Templates to a Database	
Use Supported Databases	
Supported Databases	
Providers.ini	
Understand Database Connectivity	
At a Glance: The Database Editor	
At a Glance: The Data Link Properties dialog box	
Create a Database Component Using ADO	
To create a database component	
Create a Database Component Using ODBC	
To create an ODBC data source and link it to a database component	
Save ADO Connection Information in a Separate File	
To save an ADO connection string	
Connect to Microsoft Excel using the Database Component	
To connect to Microsoft Excel using the Database Component	
Edit a Database Component	
To edit a database component	
Link Variables to Database Fields	
To create links between variables and database fields	
Remember Selected Records	
To identify a field as a key field	
Save Changed Answers Back to the Database	

To save changes back to a database record	943
Change Database Component Properties	943
To change database component properties	944
Use a Custom Command to Retrieve Data from a Database	945
To create a database component using a custom command	946
Choose Which Database Fields Appear During Assembly	947
To display only selected fields	947
Cause HotDocs to Update Selected Records Between Assemblies	947
To update selected records between assemblies	948
Limit the Number of Database Records from Which a User Chooses	948
To filter certain records from the database table	948
Specify a User Name and Password for a Database	949
To specify a user name and password at the Options tab of the Database Editor	950
Sort Records in a Database Table	950
To sort the records for the user	950
Generate a List of Records in an Assembled Document	951
To assemble a single document that contains a list of records	951
Tips on Filtering and Sorting Repeated Database Components	951
Filtering Repeated Database Components	952
Sorting Repeated Database Components	952
Merge Filtered Records into the Document Automatically	952
To merge all filtered records into the document during assembly	952
Ask Variables Already Answered by the Database	953

Select Related Database Records	954
To select related records in another table	954
Control When Your Record Selection Screen Appears	
To insert an ASK Database instruction	955
Choose a Database Cursor	955
Client Cursors	956
Server Cursors	
To specify which type of cursor you will use	957
Using Dot Codes to Format Text and Insert Special Characters	959
Introduction: Dot Codes	959
Working with Plain Text	959
Punctuating Non-Repeated Lists	959
Full List of Dot Codes	959
Change Font Properties of Text	962
To apply formatting to plain text	962
Insert a Web Link	964
To include a Web address in a prompt or dialog text element	964
To include a Web address in the document text	964
Add Punctuation and Capitalization to Sentences	965
To add punctuation and capitalization dot codes	965
Insert Characters in Text Strings	966
To insert special characters in plain text	966
Punctuate Non-Repeated Lists of Answers	968
To punctuate a non-repeated list	

Testing Automation	971
Testing HotDocs Automation	
Introduction: Test HotDocs Templates	
Test Individual Variables	
Test a Computation Script	
Test a Custom Dialog	
Test Assemble a Document	
Use a Test Answer File	
Simultaneously Edit a Template and Test Assemble a Document	
Understand and Resolve Syntax Errors in a Template or Script	
Using HotDocs Debugger	
Introduction: Debugging Templates	
At a Glance Debugger Window	
Insert Debugging Instructions in Templates and Scripts	
Step Through a Template or Script	
At a Glance: The Add or Remove Variables dialog box	
Add Variables to the Debugger Watch List	
Using the Text Panel	
Introduction: Use the Test Panel	
Validate Variable Usage in a Template	
At a Glance: The Test Panel Window	
Check Interviews for Improper Scripting	
Move From the Document to the Template During Testing	
Publishing Templates	

Publishing HotDocs Templates	
Introduction: Publish HotDocs Templates	
At a Glance: Publishing Options (The Publishing Wizard)	
At a Glance: Published Template Set (The Publishing Wizard)	
Publish a Template as a Standard File	
Publish Templates as Auto-Assemble Files (HDA)	
At a Glance: Auto Install Target Library (The Publishing Wizard)	
Publish Templates as Auto-Install Files (HDI)	
At a Glance: File Security (The Publishing Wizard)	
At a Glance: Component File Expiration (The Publishing Wizard)	
Protect Published Files	
Publish and Register Templates for Use With HotDocs Player	1005
At a Glance: Target Library Short-Cut (The Publishing Wizard)	1005
Create a Shortcut for an Auto-Install Library	1007
Create an HDI File for a CD-based Template Set	1007
At a Glance: Additional Files (The Publishing Wizard)	
Add Files to a Published Set	1009
Save Publishing Settings	
Use a Custom Library Splash Screen	
Automatically Update Published Template Sets	
Publishing HotDocs Server Templates	1013
HotDocs Server Overview	1013
Template Features Not Allowed in HotDocs Server or Cloud Services Interviews	1015

Featur	es that Work Differently in Desktop Interviews than in Server and Cloud Services Intervie	ews 1016
Enable	Templates for Use With HotDocs Server	1019
Viewin	g an Interview in a Web Browser	1020
Publisł	n Templates For Use With HotDocs Server	1022
What i	s a Publishing Form Page?	1023
What i	s a Publishing Destination Page?	1024
Publishin	g Templates for HotDocs Cloud Services	1025
Overvi	ew: Upload HotDocs Files to a HotDocs Cloud Services host application	1025
Enablir	ng a Template for Uploading to Cloud Services	1026
Adding	g an Expiration Date to a Template for Use with Cloud Services	1026
Publisł	ning and Uploading Templates and Other Files for Use with Cloud Services	1028
Using	the Upload Plug-in to Save Template Packages Locally	1029
Uploadin	g Published Files	1031
Introdu	uction: Upload HotDocs Files to a Web Server	1031
Cloud	Services Overview	1031
Create	Custom HTML Documents for Template Uploading	1033
At a Gl	lance: The Web Destination dialog box	1034
Upload	d Templates to a Web Server	1035
Uploadin	g Published Files to a Server	1035
Introdu	uction: Upload HotDocs Answer Files	1035
Specify	y Options to Upload Answers	1036
Handle	e Data in Uploaded Answers	1037
HotDocs M	odels	1041

	Introduction: Create HotDocs Models	1041
	Creating a Simple HotDocs Model	1041
	Basic Markup	1041
	Mark Variable Text	1042
	Mark Conditional Text	1043
	Mark Repeated Text	1045
	Mark Special Characters	1045
	Hide Markup Clutter	1046
	Apply Color to Markup	1047
	Creating Full HotDocs Models	1048
	1. Preparing Your Documents	1048
	2. Markup the Document	1051
	3. Changing the Appearance of Questions and Dialogs in the Interview	1080
	4. Using the HotDocs Model in HotDocs	1085
	5. Including HotDocs Scripting In a HotDocs Model	1089
	Example HotDocs Models	1097
	Simple Markup Example (Contract)	1097
	Simple Markup Example (Agreement)	1098
	Complex Markup Example with Tables (Last Will and Testament)	1099
	Quick Reference Guides	1109
	Quick Reference—Field Types	1109
	Quick Reference—Field Properties	1111
ι	Ising Command Line Options	1127
	Introduction: Command-Line Options	1127

See Also	1127
Use Command-Line Options When Starting HotDocs	1127
To use command-line options when starting HotDocs	1127
Use Command-Line Options when Using a Shortcut to Start HotDocs	1128
To use command-line options when using a shortcut to start HotDocs	1128
Use Command-Line Options within ASSEMBLE Instructions	1128
To use command-line options within ASSEMBLE instructions	1128
Use Command-Line Options at File Properties	1128
To add command-line options to a library item	1129
Full List of Command-Line Options	1129
Application Control Switches	1129
Template Type Switches	1130
Answer Initialization Switches	1130
Interview Behavior Switches	1131
Answer Disposition Switches	1132
Document Disposition Switches	1133
Automator/Filler Command Line Switches	1133
Answer File	1135
Answer Summary	1135
Clause Name	1136
Default Answer File	1136
Discard Answers	1136
Don't Brag	1137
Edit Template	1137

Exit HotDocs	1137
Finish Interview Action	1137
Hide Library	1138
HotDocs Auto-Assemble File	1138
HotDocs Auto-Install File	1138
HotDocs Model	1139
Interview Scope	1139
Keep Interview Group	
Library File	
Lock Answer File	
Lock Library	
New Answer File	
No Assembly Window	
No Interview	
Output File	
Limitations When Using /of with /nw	
Overlay Answer File	
Paper Size	
Paper Tray	1145
Print	
Print Answers Only	
Print Both	
Print Copies	
Print Duplex	1147

Print Form Only	
Print Without Dialogs	
Question Summary	
Save Answers	
Save Answers Prompt	
Send to Plugin	
Send to Word Processor	
Show Library	
Start Interview Group	
Suggest Save	
Suggest Save New	
Suppress Unanswered Warning	
Template File	
Troubleshooting	
Template Development	
Converting to HotDocs 11.1 Produces Compo	onent Name is Invalid Error 1153
Can't Dynamically Add Merge Text to a Multi	ple Choice Variable 1153
Changing the Order Questions are Asked in a	in Interview 1153
Checking the Spelling of Library Entries	
Child Dialog Always Appearing in Interview C	Outline Even When it Shouldn't
Choosing Which Word Processor Will Be Use	d By Default 1155
Comparing the Contents of Two or More Cor	nponent Files 1155
Converting to HotDocs 11.1 Produces Compo	onent Name is Invalid Error1156
Converting WordPerfect Templates to Word.	

'The Component File Cannot Be Used With This Version of HotDocs' Message Appearing	1156
Dot Code Formatting Doesn't Seem to Work in True/False and Multiple Choice Options	1157
Editing Answer Source Records Causes Problems	1157
Enabling Multiple Templates for Use with HotDocs Server At Once	1158
Entered Answers Getting Overwritten	1159
Entering Multiple Choice Option Prompts and Merge Text	1159
Fixing Warnings Listed at Warnings Tab	1160
Foreign Characters Are Appearing as Boxes or Question Marks in HotDocs	1161
Getting Missing END IF or Missing END REPEAT Errors	1162
Hiding the End of Interview Dialog During the Interview	1163
'HotDocs Cannot Play the Macro HD6EditTemplate.Main' Message Appearing	1163
HotDocs is Creating Database File with Library File	1164
Incorrect formatting of multi-level list that includes HotDocs Instructions	1165
Keeping Users from Editing Templates	1165
Limiting Visible Rows but Not Limiting Number of Answers	1166
Matching Pairs of Instructions in a Computation Script	1166
Merging check boxes in a Text Template	1168
Merging Just a Portion of a Date Instead of the Whole Date	1169
Optional Variables Showing Up as Unanswered in Interview and Document	1169
Punctuating a Repeated List of Answers	1170
Receiving 'Invalid Variable Name' Errors When Creating Components	1170
Reducing Template File Size	1172
Renaming Components Correctly	1172
Repeating a Single Variable in a Dialog	1173

Repeating the Contents of a Single Cell in a Table	
SET Instructions Using Dialog Elements No Longer Working	
Test Assembling Template with ASSEMBLE Instructions	
Text Not Displaying Properly in Documents Assembled from Word 2007	۲emplates 1174
True/False Variables Default to False on Spreadsheet Dialogs	
Trying to Save an Invalid Computation Script	
Understanding the Difference Between Multiple Choice Options and Mer	ge Text 1176
Using DEFAULT Instruction Causes Extra Repetition	
Using Headers and Footers of Inserted Templates	
Using INSERT Instructions in a Table	
Using One Variable in Two Different Dialogs	
Variables Not Working in Word Text Boxes	
Watermark Disappearing	
Document Assembly	
Can't Attach Answer File or Document to Outlook 2007 E-mail Message	
Changing Unanswered Text Markers	
Discerning Between Template Preview and Actual Template	
Downloading HDIs and HDAs as ZIP Files	
Editing Answers in Assembled Document	
Entering a Two-Digit Year	
HotDocs Skips Questions When Navigating from Child Dialog	
HotDocs Skips Questions When Navigating from Child Dialog	1183 s Server 1183
HotDocs Skips Questions When Navigating from Child Dialog Images Appear Incorrectly After Converting an RTF to PDF Using HotDoc Increasing Font Size	

	Moving Between Questions in Interview Slow	. 1185
	Moving to the Next Dialog After Entering a Date or Number	. 1185
	Organizing Files in the Library vs. on the Disk	. 1186
	Overriding an Answer in a Form	. 1186
	Printing Form Document is Off-Center	. 1187
	Printing Form Document Takes Incredibly Long Time	. 1187
	Receiving Error Messages When Launching HDAs, HDIs	. 1188
	Restoring System Defaults to HotDocs	. 1189
	Saving Answers During Interview	. 1189
	Specifying Printing Options When Printing Text Documents	. 1190
	Text Not Displaying Properly in Documents Assembled from Word 2007 Templates	. 1190
	Understanding What Happens When Deleting Templates from the Library	. 1191
	Updating Changes Made in Word Processor Document in the Answer File, Template	. 1192
	Updating Cross-References in an Assembled Document	. 1192
	Viewing File Names Instead of Template Titles at Library	. 1192
F	lotDocs Automator	. 1193
	Double-Clicking Doesn't Create Field as Expected	. 1193
	Fixing Page Rotation Problems in PDF Templates	. 1194
	Getting Errors When Automating Table in Form	. 1194
	Keeping Control Field Text From Appearing on Form	. 1194
	Modifying a Single Field in a Group	. 1195
	Moving Form Fields Using Keyboard Takes Too Long	. 1195
	Needing Characters in Answer to Appear in Individual Boxes	. 1196
	No Interview When Assembling Form Document	. 1196

Setting Form Properties Not Taking Effect in Existing Templates	
Tabbing Through Fields Sequentially in Form	1197
Want check box to Use a Tick in the form check box	1197
Legal Notices	1199
Software License Agreement:	1199
Third-Party Software Acknowledgements	1202
Colosseum Builders, Inc	1202
Contact HotDocs Sales and Support	1203
HotDocs Technical Support	1203
For the United States, Canada, and Mexico:	1203
For Europe, Africa, Asia, Australia, and South America:	1203
HotDocs Sales Support	1204
For the United States, Canada, and Mexico:	1204
For Europe, Africa, Asia, Australia, and South America:	1205
Corporate Offices	1205
Head Office U.S.	1205
Head Office U.K	1205
Documentation Feedback	1207
Glossary	
Index	1227

Getting Started

Common Confusions

In HotDocs there are a some features which, while working as expected, may at first seem counter intuitive. The following is a list of common confusions and known issues:

See the Troubleshooting section of this help file to resolve other issues you encounter.

Template Development

Microsoft Word Features that Cannot Be Used with HotDocs

There are a couple of Microsoft Word features that are not supported in HotDocs templates. To avoid errors, we recommend you remove the following features from your documents before converting them into templates:

- Mail merge
- Smart quotes

Microsoft Word 2016 Right-click Menu Restrictions

Due to a change by Microsoft, custom right-click menus are no longer possible in Microsoft Word 2016. As a result, right-clicking on HotDocs fields within Word templates will no longer present a custom menu allowing you to edit the field or related component. This is unfortunate, but expected, behavior. HotDocs fields can still be edited by double-clicking in Word, and fields and components cat still be edited using the HotDocs ribbon in Word as well.

Using Smart Returns

We recommend becoming familiar with how HotDocs uses smart returns after inserting an instruction, before you develop any complex templates with multi-level lists. The way you set the smart returns options can alter the formatting in your assembled document. You can set your default options for smart returns at the HotDocs Options dialog box.

Clauses and Clause Libraries

HotDocs Clauses and Clause Libraries are an old feature which we do not recommend relying upon for your template functionality.

Dialog Element Image Display Issues in Browser Interviews

When you develop a template for browser interviews, we recommend you thoroughly test how images display when included as dialog elements because Silverlight and JavaScript handle image display in slightly different ways. If an image is too big to fit in the allotted space, Silverlight interviews clip the image, while JavaScript interviews resize the image. To avoid any display issues, in such cases, we recommend you resize the image so that it displays equally well in both Silverlight and JavaScript.

Using Multiple Choice Merge Text

If you are using merge text in a computation variable, make sure you remember to use quotations around the multiple choice variable name otherwise you will be using the option text. For example using «MCVariable» will use the option text in the assembled document but using "«MCVariable»" will use the merge text.

Apparent Issues with Dot Codes

Sometimes, it can appear that special characters are being considered as punctuation, so that the assembled document omits (for example) the period.

The behavior of the sentence punctuation dot codes—the dot codes for period (<<.>>), comma, semicolon, colon, exclamation point, and question mark—insert punctuation only if the preceding character is not another sentence punctuation of the kind that HotDocs inserts. For example, the period dot code will insert a period after a closing parenthesis, but will not do so if the dot code is preceded by an exclamation point.

Ampersands in Variable Prompts

You can type an ampersand (&) immediately before a character in a variable prompt to make that character an accelerator key. During the interview, an accelerator key enables the user to move the cursor to that answer field by pressing the **Alt** key while also pressing the underlined character in the prompt. This functionality can create an issue if you need to include a word in your prompt that has an ampersand immediately preceding a character. To do this without creating an accelerator key, you must type two ampersands (**&&**) where you want the ampersand to appear as a single ampersand. For example, to have "**R&D**" as your variable prompt you would need to type "**R&&D**".

ASSEMBLE instructions must be included in the template as well as in the interview script

To ensure that templates are added to, and remain in, the assembly queue, you should include identical ASSEMBLE instructions in both your interview and template scripts. This way, no matter which script HotDocs processes last, it will put the desired templates in the assembly queue.

Template Uploading

Template ID Conflicts

When uploading multiple templates, you may encounter an issue with duplicate template IDs. If two or more templates share the same template ID, HotDocs generates an error message alerting you to the problem, and specifying the templates that are in conflict.

A conflict of this sort most often occurs in cases where you copy a template (whether using Widows Explorer, or HotDocs native copy a template feature) to avoid recreating all the components in the accompanying component file. Since HotDocs assigns a unique ID to each template, merely copying a template to another location in this way, and even renaming that copy, does not change the template ID. In such cases, you end up with two templates, each with the same ID number. The proper method for creating a copy is to use HotDocs to create a new template based on an existing template.

You can fix the problem that caused the error message by following the steps below:

To change a template ID number

- 1. In the HotDocs Developer library, select the template whose ID you want to change.
- 2. On the right side of the toolbar at the top of the library, click **Component Manager**; then just below the Save icon, click **Component File Properties**.
- 3. On the General tab, to the right of the Template Identifier box, click **New**. This updates the template ID.
- 4. Repeat until you resolve all ID conflicts; then try uploading again.

Document Assembly

Problems with Text Pushing Too Far to the Left Side of the Assembled Document

Templates developed in Word 2007 with smart tags enabled can display text incorrectly. This results from a bug in Microsoft's Open XML Format SDK. If you have templates automated from documents developed in Word 2007, be sure to remove the smart tags in your templates or ask your template developer to resolve this problem.

Enabling Update table of contents, references, fields etc in HotDocs Server templates

On the HotDocs Server tab of Component File Properties, if you enable **Update table of contents**, **references**, **fields**, **etc** option, you may find that it does not produce exactly the same results as the desktop option of the same name on the Assembly tab. Unlike in desktop HotDocs, using this feature with server is known to occasionally alter or lose advanced formatting features in the assembled document. Not enabling this feature removes the chance of this error occurring but you loose the ability to dynamically update the table of contents, references, fields etc in the interview. If you choose to use this feature when converting a template for use with HotDocs server, we recommend testing all features that previously relied on the desktop version of this option.

Using the /nw Command-line Switch with the /of Switch Does Not Support Converting a Document to PDF

Normally, the file extension on the file specified in the /of command line switch instructs HotDocs to convert the assembled document to a file of that type after assembly. However, when converting to PDF, this only works when the assembly window is visible. When you hide the assembly window (by using the /nw switch), conversion to a PDF document is not supported. HotDocs does, however, save the assembled document to a file with a .PDF extension as instructed by the /of command line switch. This can be confusing if you specify "tf=template.docx /of=document.pdf" in the command line after using a /nw switch. In this case, HotDocs assembles a DOCX document from the DOCX template, but saves the DOCX document iin the file system with the filename specified by the /of switch (document.pdf). If you then double-click that file in Windows Explorer, the application mapped to the PDF file extension (usually Adobe Reader) tries to open the file but fails because the file contains a DOCX document, not a PDF document (as the file extension claims).

In cases where the assembly window is hidden by specifying the /nw switch, only native word processor file types are supported for the /of commandline switch, as follows:

Template type	Allowed output file
DOCX	DOCX
RTF	RTF, DOCX
DOT	RTF, DOCX

Work Around

A possible work around is to use Microsoft Word's PDF conversion feature to accomplish the same result, using either Word's graphical user or programming interface.

Server Assembly Behavior Differs from Desktop

Using a SET instruction in a repeated dialog script can result in unanswered repetitions in the assembled document

In a dialog script, if you SET a variable to an answer in a repeated dialog script without using a condition on the SET instruction, HotDocs sets that variable every time you create a new iteration of the dialog. As a result, your answer file may contain more answers than what you intended. (This is especially true if the variable in the SET instruction is hidden on the dialog.) Although you may not see that repetition in the interview, it exists in the answer file.

During a desktop assembly, HotDocs keeps track of how many repetitions of the dialog are visible and then uses that information when assembling the document to prevent seemingly unanswered repetitions from appearing in your document. However, if you perform the assembly silently (without showing the

assembly window), or you perform the assembly using HotDocs Server, HotDocs does not track the number of repetitions shown in the interview and must assemble the document based only on what is available in the answer file.

You can avoid the situation where these extra, unanswered repetitions appear in your assembled document by adding an IF instruction to the dialog script so that the script only SETs the variable if some other variable in the repetition is answered. Likewise, you can add a FILTER on the REPEAT instruction in your template to filter out any unwanted repetitions in which the only answered values are the ones that are hidden.

The following is an example of a dialog script that only SETs a variable if you answer another variable:

HIDE Repeat Variable IF ANSWERED(Test2) SET Repeat Variable TO "AutomaticallySetValue" ELSE SET Repeat Variable TO UNANSWERED END IF

Special characters that cannot be used in Cloud Services package IDs

'/', '\', and '?' cannot be included in the template name when you are creating a template package for use in a HotDocs Cloud Services direct interview.

Known Issues

The following issues are known and are either not addressable by HotDocs as they originate in external software or are planned to be resolved the future releases. In the meantime you can use the workarounds listed below:

• Multiple choice variables populated using a SET instruction do not differentiate between upper and lower case when you edit an answer in the interview. For example, the multiple choice variable won't honour changing the answer "JOHN DOE" to "John Doe". It continues to display the uppercase answer until you make a character change, only then does it recognise you have typed in a new answer. To avoid this problem you need to add an ERASE VAR instruction to the beginning of your computation, rather than using CLEAR VAR. For example, a computation like this would honour the changes between upper and lower case:

ERASE MCVariable ADD Txt1 TO MCVariable SET MCVariable TO Txt1

- Double ampersand is not escaped correctly in the dialog title when the Combine Default Dialogs option is selected. When the Combine Default Dialogs option is on, HotDocs takes the variable title, for example "R&&D", and instead of correctly escaping the first ampersand, so it appears in the interview correctly as "R&D", it appears incorrectly as "RD". To avoid this issue it is best to not use Combine Default Dialogs if you want to use ampersands in your dialog titles and instead create your own dialogs using the component file.
- When installing HotDocs Developer 10 or 11, users very occasionally see this error message: "Error 2738- could not access VBScript run time for custom action". If this occurs you need to download the fix from Microsoft, which removes any corrupt registry settings and then re-register the correct ones. You may need to restart your computer to see the change.
- On Windows Vista, if you try to assemble a .dot template on a new installation of HotDocs Developer, Windows is occasionally unable to open Microsoft Word correctly to assemble the document. To avoid this problem HotDocs recommends assembling an .rtf or .docx template after installing a new version of HotDocs Developer. If you do assemble a .dot template first, and experience this issue, you need to remove the template from your library, restart HotDocs Developer, then add the template again. That should solve the issue.

Browser Interviews

• In a Javascript interview, selecting "None of the above" on a list style multiple choice question in the interview, occasionally does not clear any other selected options. You need to hold down Ctrl and click the selected options to clear them and then you can select "None of the above" only.

Quick Help Links

Using the links below, you can quickly find topics of interest, particularly if you are a new user.

- New and Enhanced Features of HotDocs 11
- Complete the HotDocs Tutorials
- Get Help Using HotDocs

Starting HotDocs

- HotDocs Overview
- Start HotDocs
- Introduction: Use HotDocs Libraries
- Create a Library
- Work with Templates and Other Files in a Library

Automating a Template

- Introduction: Template and Component Files
- Create a New Text Template File
- Use Component Manager to Work with Components
- Insert a Variable Field in a Text Template
- Gather Questions into a Custom Dialog
- Create a REPEAT Instruction to Gather a List of Answers
- Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression

Assembling a Document

- Assemble a Text or Form Document
- Select an Answer File for Assembly
- Navigate Through Answer-Gathering Dialogs
- Preview the Assembled Text Document
- Save an Assembled Text or Form Document
- Save an Answer File During Assembly

What's New in HotDocs Developer?

New and Enhanced Features of HotDocs 11

New and Enhanced Features of HotDocs 11.2

Compatibility with Microsoft Word 2016

You can now build and use templates compatible with the latest version of Microsoft Word.

HotDocs 11.2 is a paid upgrade and license codes from earlier versions are not compatible. If you want to upgrade to HotDocs 11.2 you need to contact your HotDocs representative.

New and Enhanced Features of HotDocs 11.1

More flexible dialog validation with the new ERRORTEXT key word

You can use the new ERRORTEXT keyword in dialog scripts to create custom error messages for your end users, as well as to prevent users from proceeding through the interview until they satisfy custom validation rules.

Automatic grouping of variables into default dialogs

Use the new **Combine default dialogs** component file property to improve your end user's experience. This option groups individual variables (those not otherwise grouped into custom dialogs) into default dialogs in the interview. This keeps your end user from needing to advance through multiple variables one at a time, as part of an interview.

In the 11.1 release, this feature is valid only for templates published to the desktop environment.

Easier creation of shared component files

When creating templates, the New Template Dialog box now enables you to designate a shared component file from among those in the target directory.

More flexible repeats within table cells

Previously, the only way to repeat information inside a single table cell was by using a computation variable. Now you can also embed a REPEAT/END REPEAT block directly in a table cell. This is in addition to the existing use of REPEAT fields to repeat entire table rows.

Improved Update table of contents, references, fields, etc. for server-based assembly

The server-side code that updates tables of contents, cross references and other types of fields after assembly is improved, and now more closely resemble the behavior of Microsoft Word. However, serverbased field updates are still only an approximation of the behavior of Word on the desktop. Consequently, the former component file property to control whether such field updates should occur or not has been replaced by two new component file properties: one which governs behavior when assembling in desktop HotDocs, and a new check box on the HotDocs Server tab of the Component File Properties dialog that governs behavior when assembly is taking place on a server. When converting a template for use with HotDocs server, we recommend testing all features that previously relied on the desktop version of this option.

Support for INSERT instructions embedded in computation variables

Templates containing INSERT instructions embedded within computation variables (rather than included in a template's main content) now work on both HotDocs Server and HotDocs Cloud Services. This change requires you to re-publish any existing affected templates.

WordPerfect X7 Support

You can now build and use templates compatible with the latest release of WordPerfect. Note, though, that HotDocs no longer supports WordPerfect 12.

New license model

HotDocs Developer, LE and User now require a valid license key to run. If you do not provide a license, the regular HotDocs Developer installer automatically installs HotDocs with a 30-day evaluation license.

The evaluation period begins with the date of first use rather than the date of install.

Support for Inserting images into tables in DOCX templates

DOCX templates now join RTF templates in supporting INSERT FILENAME IMAGE fields within table cells.

New currency symbols added to the Number Variable Editor

The Yen (¥) and Pound (£) symbols are now available in the Currency Symbol drop-down list in the Number Variable Editor, regardless of system locale settings.

Maximum Field Size Increased

The maximum size for fields in Microsoft Word templates has been increased from 256 to 512.

New and Enhanced Features of HotDocs 11

Word processor support

HotDocs 11 now supports Microsoft Word 2013 and WordPerfect X6.

It no longer supports Microsoft Word 2000 and XP or WordPerfect 8,9,10, and 11.

New support for DOCX file format

HotDocs now supports DOCX format templates with no file conversion throughout, allowing HotDocs to better utilize the native properties of Microsoft Word, reducing any unexpected errors in assembly and avoiding unmanageably large file sizes. HotDocs will still support all RTF templates.

New support for Plain Text Templates

In addition to word processor based templates, HotDocs now supports Plain Text Templates (.ttx files). These templates can be automated using a text editor and when assembled the finished document can be sent to your default text editor.

Parameters and Local Variables

A new feature in HotDocs Scripting is the use of Parameters and Local Variables for use in more complex scripting tasks. Parameters can be re-used throughout a template to save creating multiple computation variables with the same use, while local variables only exist within a particular computation. Neither of them appear in the Component Manager and they can both be set from the Computation Variable Editor.

New Expressions

New Math Expression Models: HotDocs now includes the following 6 new math expressions:

- LOGARITHM(NUM): Find the common logarithm of a number variable.
- NATURAL LOGARITHM(NUM): Find the natural logarithm of a number variable.
- EXPONENTIAL(NUM): Find the exponential value of a number variable.

- SQUARE ROOT(NUM): Find the square root of a number variable.
- FLOOR(NUM): Find the floor value of a number variable.
- CEILING(NUM): Find the ceiling value of a number variable.

New Financial Expression Models: HotDocs now includes the following 5 new financial expressions, all of them utilize the same formula but differ on which value you are looking to generate :

- FUTURE VALUE (RATE, TERM, PAYMENT, PRESENT VALUE, TYPE)
- PAYMENT (RATE, TERM, PRESENT VALUE, FUTURE VALUE, TYPE)
- PRESENT VALUE (RATE, TERM, PAYMENT, FUTURE VALUE, TYPE)
- RATE (TERM, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE)
- TERM (RATE, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE)

New Text Expression Models: The following text expressions have been modified from existing expressions:

- POSITION(TEXT, TEXT, TRUE_FALSE): This function used to be POSITION(TEXT, TEXT) but it now
 contains an optional true/false variable placeholder where false (or nothing) indicates HotDocs
 should begin from the left hand side of the text and true indicates HotDocs should begin from
 the right.
- TRIM(TEXT): A new expression for removing white space that does the equivalent function of STRIP(TEXT, "\t\", TRUE, TRUE).
- TEXT STARTS WITH TEXT: This is a new variable based on TEXT CONTAINS TEXT but HotDocs checks if a text variable starts with the selected text.
- TEXT ENDS WITH TEXT: This is a new variable based on TEXT CONTAINS TEXT but HotDocs checks if a text variable ends with the selected text.

Help System Re-structure

Removal of the What's This help feature: The **What's This** help buttons have been removed from HotDocs windows and dialogs.

Window Specific Help File Pages: The **What's This** help buttons have been replaced with a new buttons and multiple which will open the relevant *At a Glance* window or dialog box page from the HotDocs help file.

Re-structure of the HotDocs Help File: The HotDocs Help File has undergone a change to the style and structure of the table of contents. There has been improvements made to the search function, index and glossary as well as the introduction of new *At a Glance* pages, where you can see the main features of HotDocs dialog boxes and windows. In the new help file *Introduction* pages are also now listed as *Introduction* pages.

HotDocs Models

Model Documents have been re-named to HotDocs Models for the release of HotDocs Developer 11. Model Document Markup Language (MDML) will also now be referred to as Markup to avoid any confusion with other programing languages. This name change is accompanied by an updated toolbar in Microsoft Word and a more comprehensive help system. Information for Developers can now be found in this help file and information for HotDocs Model users can be found on our help website.

Inserting Instructions into template

Changes to INSERT instructions: INSERT instructions will now work in Headers and Footers.

Using SPAN instructions in DOCX templates: SPAN sections will be stored in an RTF format during assembly and silently converted to DOCX when the document is opened or sent for print.

Import or Export Computation Variables

You can now export Computation variables in a HotDocs readable syntax, allowing you to share computations with other HotDocs users. (See Import or Export Computation Variables)

Template IDs

HotDocs now generates a unique template ID number for every new template. You can see this at the **Component File Properties** dialog box.

Additional Improvements

In addition to the new features above, this release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 10

New and Enhanced Features of HotDocs 10.2

Upload Plugin for HotDocs Cloud Services

HotDocs Developer now provides a plugin for uploading templates to web applications that use HotDocs Cloud Services for document assembly.

Additional Improvements

In addition to the new features above, this release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 10.1

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 10 HF3

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 10 HF2

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 10 HF1

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 10

Edition Name Changes

HotDocs is now available in four different editions:

New Name	Previous Name
Developer	Professional
Developer LE	Standard
User	
Player	Player

Word Processor Support

Word 2010 support: HotDocs now supports the use of Word 2010.

WordPerfect X5 support: HotDocs now supports the use of WordPerfect X5.

(See Install Support for New Word Processors for information on installing support for new word processors.)

Silverlight Interview

Templates intended for use with HotDocs Server can now be test-assembled using the Microsoft Silverlight web application framework. Unlike the traditional JavaScript-based interviews, where the interview logic is written in an interpretive language (JavaScript), Silverlight-based interviews are compiled into machine code, which makes them much quicker. This is especially noticeable in long, complex interviews.

(See View an Interview in a Web Browser for more information.)

Desktop & JavaScript Interviews

Desktop and JavaScript interviews have been updated and improved for HotDocs 10.

PDF Support

The functionality previously provided by HotDocs PDF Advantage is now integrated with HotDocs 10.

Changes to Files, Folders and Registry Keys

HotDocs Files

New Filename	Previous Filename
hotdocs.exe	hotdocs6.exe
hotdocs.dot	hotdocs6.dot
HotDocs.dotx	HotDocs6.dotx
hd_api.dll	hd6api.dll
hd_dispatch.exe	hd6dispatch.exe

HotDocs Default Installation Folder

New Folder	Previous Folder
C:\Program Files\HotDocs	C:\Program Files\HotDocs 6

Registry Keys

New Key	Previous Key
HKEY_CURRENT_USER > Software > HotDocs >	HKEY_CURRENT_USER > Software > LexisNexis >
HotDocs	HotDocs 6
HKEY_LOCAL_MACHINE > Software > HotDocs	HKEY_LOCAL_MACHINE > Software > LexisNexis >
> HotDocs	HotDocs 6

New and Enhanced Features of HotDocs 2009

New and Enhanced Features of HotDocs 2009 HF2

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 2009 HF1

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 2009

File Formats

Save component files, library files (both template and answer), and answer files in XML format: HotDocs now allows you to save component files, library files, and answer files in either XML format (required to use HotDocs 2009 features) or binary format (required to be compatible with HotDocs 2008 and earlier).

Please see Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs for a complete description of this format change.

Unicode Support

Automate templates using international languages: You can now automate your templates using any left-to-right reading languages that are supported in the Unicode character set. See Use International Characters in HotDocs for details.

Template Development

Use INSERT instructions to include images: You can now use an INSERT instruction to insert JPGs, PNGs, and GIFs in a template. Additionally, you can insert RTF documents that contain image files. INSERT image instructions are appended with the IMAGE keyword.

Use variable INSERTs to specify inserted file names during assembly: Template developers can now use a variable INSERT in a template, which allows the file path and name of a template file or graphic file to be specified at the time of assembly, rather than during template development. You must use a Text variable, a Multiple Choice variable, or a Computation variable that results in text with a variable INSERT.

Navigate repeated dialogs using new button: When completing an interview, end users now add new repetitions to a repeated series list by clicking the Add Another button, which is located both after all questions in the dialog as well as in the dialog's title bar. (Users can no longer use the previously available methods for adding new repetitions to a dialog.) You can customize the title of this new button on a perdialog basis by specifying it at the Dialog Editor.

Define a default behavior when automatically disabling irrelevant variables: At **Component File Properties**, you can specify a default behavior for irrelevant variables you automatically disable during the interview.

Enter comments about components at the Notes tab: Component editors now include a **Notes** tab where you can enter notes about the component, such as an explanation about why the component was created or how it should function in the interview.

Hide editing buttons on spreadsheet dialogs: You can now keep HotDocs from showing the **Edit Row**, **Insert Row**, and **Delete Row** buttons associated with a spreadsheet dialog. (The options are still available via buttons and menus, however.)

Publish templates so they're compatible with HotDocs 2005-2008: HotDocs lets you publish templates created in HotDocs 2009 so they will still be compatible with HotDocs 2005-2008.

Document Assembly

Install auto-install file libraries to any location: HotDocs now allows you to install a template set library to the same folder as you install the templates sets.

Use new method for adding repetitions to a list of answers: When completing an interview, users now add new repetitions to a repeated series list by clicking the **Add Another** button, which is located both after all questions in the dialog as well as in the dialog's title bar. (Users can no longer use the previously available methods for adding new repetitions to a dialog.)

HotDocs Automator/Filler

Create PDF templates from fillable PDFs: You can create PDF templates based on fillable PDF documents. When you create the template, HotDocs first presents you with several options for working with existing fillable fields—you can have HotDocs create mirror fields that match the existing fillable fields. You can also have HotDocs remove all existing fillable fields. Finally, if the PDF document contains annotations (like sticky notes, stamps, and so forth), you can remove them from the newly created template. Once you create a template that contains fillable fields, you can attach HotDocs variables to those fields using an automated wizard.

Save assembled forms as fillable PDFs: You can assemble a document from a template that was created from a fillable PDF document and save it once again as a fillable PDF document.

New and Enhanced Features of HotDocs 2008

New and Enhanced Features of HotDocs 2008 SR 2

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 2008 SR 1

Word Processor Support

WordPerfect X4 support: HotDocs is now supported for use with WordPerfect X4. (See Install Support for New Word Processors for information on installing support for it.)

New HotDocs Help Format

The look and feel of the HotDocs Help file has changed. The Help is contained in a single window, with new navigation options along the top of the window. Searching the help file is much more intuitive and useful. For details, see Get Help Using HotDocs and Form a Help Search Query

Document Assembly

Send assembled PDF documents to Adobe: If you're assembling PDF-based form documents, you can select a HotDocs option that will send assembled documents to Adobe Acrobat, rather than HotDocs Filler. See Send Assembled PDF Documents to Adobe for details.

HotDocs Automator and Filler

Create independent addendums: You can create multiple, independent addendums for forms where each page of the form represents a duplicate. See Create Independent Addendums for Forms.

New and Enhanced Features of HotDocs 2008

Model Documents

Create model documents, independent of HotDocs, and use them to assemble custom documents: Using Microsoft Word, you can create model documents. A model document is an exemplar document used as the basis for drafting documents of the same type for specific clients. Using a defined markup language, you identify text that varies from one client to another. This markup provides an unambiguous description of how a model document should be used to draft client-specific documents. Once you have a model document, you can use it with HotDocs User to generate custom documents for your clients.

Template Development

New Go to Field button added to HotDocs toolbar: When you receive scripting error messages in HotDocs, they often include the field number for the field where the error is located. To help you quickly find this field in a text template, HotDocs now includes a **≣**+Go To Field button in the HotDocs navigation toolbar.

New Test Panel added: HotDocs now includes a Test Panel, which you can access during a test assembly. Features of this test panel include the following tabs:

- **Variable usage:** This tab shows how variables are being used in the template. For example, you can review a list of variables that are asked during the interview, but they aren't used in the document. You can also review which variables are used in the document but don't appear in the interview. These two lists can help you make sure the questions you are asking the user are relevant to the document, which ultimately improves the accuracy of the document.
- **Warnings:** In previous versions of HotDocs, the **Warnings** tab was included directly in the assembly window. This tab has now been moved to the **Test Panel**. It includes information about scripting in the template that may cause unexpected results in the interview or assembled

document. For example, the **Warnings** tab can report when SET instructions have been used incorrectly. It can also identify when variables have been asked or referred to incorrectly.

• **Go to Template:** Using the **Go To Template** command, you can place your cursor in the **Document Preview** text and have HotDocs take you to the same place in the template. This lets you review the automation in that particular section of the template, perhaps to understand the conditions under which the text was inserted in the document.

Publishing licenses now include annual expiration: Licenses for publishing and registering templates for use with HotDocs Player now expire annually.

Have HotDocs automatically exclude irrelevant questions from an interview: Rather than write dialog scripts, you can have HotDocs automatically gray or hide variables that aren't relevant to the document.

Set HotDocs Server interviews so all dialogs in the interview appear in the same view: You can design your HotDocs Server interviews so that users can view all of the dialogs in an interview at once, on a single Web page.

Document Assembly

Entering Non-English answers during an interview: During a HotDocs interview, you can enter answers in left-to-right reading languages other than English.

Create a Favorites folder in the template library and add templates to it: You can group the templates you use most frequently in a *Favorites* folder of your library.

Database Connection

All installations of HotDocs now include the HotDocs Database Connection by default. You do not need to install it separately.

HotDocs Compare

HotDocs Compare now available separately for purchase: To use HotDocs Compare, you must now purchase a separate license for it. For details, contact your HotDocs sales representative. NOTE: Starting with the release of HotDocs 2009, HotDocs Compare is no longer available.

New and Enhanced Features of HotDocs 2007

New and Enhanced Features of HotDocs 2007 SP1

This release of HotDocs contains software corrections and minor enhancements to existing features. It also contains the following new features:

New Web link dot code added: You can now use the Web link dot code to insert hyperlinks in your prompts, plain-text resources, and dialog text elements.

New and Enhanced Features of HotDocs 2007

HotDocs supported on Windows Vista: HotDocs is now supported for use on the new Microsoft Windows Vista operating system.

Support for Microsoft Word 2007 added: HotDocs now supports Microsoft Word 2007.

View document markup in assembled documents: You can now change the formatting of an assembled document to **Markup View** and then submit it to an attorney or peer for revisions or corrections.

New command-line options: HotDocs now includes two command-line options—**Start Interview Group** and **Keep Interview Group** (**/sig** and **/kig**). These options are used to control which questions are asked when assembling a group of related documents; specifically, they keep questions that are already answered in one interview from being asked in subsequent interviews.

Access HotDocs Help directly from Expression and Instruction Model lists: You can now select a model in either the Expression models list or the Instruction models list and press Ctrl+F1 to view the help topic for that specific model.

New and Enhanced Features of HotDocs 2006

New and Enhanced Features of HotDocs 2006 SP1

Usability

Better keyboard accessibility: In an effort to make HotDocs compliant with Section 508, you can now access most commands and elements in various HotDocs windows and dialog boxes using the keyboard. Additionally, HotDocs is now more compatible with screen readers.

HotDocs Automator and Filler

End-user option of splitting an overflowing answer between the form and the addendum: Form users can now decide whether multi-line answers should be split between an answer field and the addendum.

New and Enhanced Features of HotDocs 2006

Template Development

Allow users to edit assembled document text while viewing the Document Preview tab: Sometimes users may need to edit the text of a document once it has been assembled. As the developer, you can control editing by marking sections of a template using a SPAN instruction. This allows users to edit that section of text at the **Document Preview** tab of the assembly window. Changes made during assembly can be saved to the answer file so that users can reassemble the same document later and have their changes reapplied.

Improved implementation for adding additional items to a dialog: HotDocs now provides a new Dialog Element component that lets you more easily add additional text, hyperlinks, buttons, graphics, lines, and spacing to dialogs. This implementation includes greater control over how these items appear, including control over whether they appear in desktop interviews, HotDocs Server interviews, or both. (In previous versions of HotDocs, many of these elements were added to a dialog using the @ command. This new feature replaces the need to use this command.)

Remove hidden data from assembled Microsoft Word templates and documents: Often, when working in Microsoft Word templates and documents, hidden data is saved to the file that may compromise the security of the document or increase the document's file size. You can choose which of these properties should be removed from a template. Additionally, you can have this data removed from a document after it is assembled.

Use HotDocs dot codes: HotDocs 2006 includes a new feature called dot codes, which allow you to 1) format text results derived from computation scripts, 2) insert special characters in plain text and template text, 3) format variable prompts and additional text, and 4) punctuate non-repeated lists of answers.

View an outline of scripting in a Word template: Using the HotDocs Outliner, you can generate an outline of scripting in the template. This outline can include just a list of instructions used in the template, or it can include variables as well. This gives you a more condensed view of instructions used in the template.

New schemes added for coloring fields in a template: Two new color schemes were added to help you better identify instructions in a template. They are **Nested** and **Sequential**. Nested marks each level of IF and REPEAT instructions using a custom color. (For example, all first-level IF instructions will use a specific color, while all second-level instructions will use a different color, and so on.) Sequential marks each IF and REPEAT instruction field using a custom color. (For example, the first instruction in a template will be marked using one color, while the next instruction will be marked using a different color, and so on.)

Display Word templates in Markup View: You can now display a Word template in Markup View, which allows you to provide non-HotDocs users with easy-to-understand versions of your templates.

Generate default interviews or specify custom interviews for all template types: When automating a template, you can now select whether HotDocs should generate a default interview for the template or whether HotDocs should use a custom-scripted interview. If choosing to use a custom interview, you can assign any name to the interview component. (In previous versions, all interview scripts were required to be named INTERVIEW.) You now designate the name of the interview component at the **Component File Properties** dialog box.

Choose which component file provides properties when pointing templates: In past versions of HotDocs, when you pointed a component file to a shared file, most of the properties for the pointed template would be specified by the shared component file -not the pointed component file. Now, if you need the pointed template to use its own component file properties, you make that designation at the component file.

Control how HotDocs inserts returns after inserting instructions in a template: When you insert an instruction in a text template, by default, HotDocs inserts a return after the instruction. (This return is removed during assembly.) You can select a HotDocs option that inserts these returns only when working with entire paragraphs of text, otherwise HotDocs won't insert a return character. (This keeps instructions within a paragraph from breaking up the paragraph text.)

Use new expression models to manipulate text answers: HotDocs includes four new expression models:

- **REPLACE** lets you search a string of text for a given character string and replace the results with new text.
- **SPACE** tests whether the variable is answered. If it is, it merges the answer, followed by a space character. If the variable is unanswered, it merges nothing ("").
- **STRIP** removes a specified character or characters from the beginning or end of a text answer.
- **VALUE** returns a default value for the variable type if the variable is unanswered. If the variable is answered, the value is the answer the user specifies.

Use new expression models to manipulate multiple choice answers: HotDocs includes two new expression models:

- **SELECTION** returns a specific selected option of a Multiple Choice answer.
- **UNION** combines all selected (and unique) options from two Multiple Choice variables.

Use new instruction models to erase answers in repeated dialogs: HotDocs includes two new instruction models:

- **ERASE VAR** clears all answers for a specific variable in a repeated dialog.
- **ERASE DIALOG** clears all answers in a repeated dialog.

Create 'Used In' lists for components: You can view which other components in the template use the component you are currently editing. This information is displayed in the new **Used In** tab of the component editor. (This tab was formerly the **Asked In** tab, which showed only the dialogs that used the variable.)

Store PLAY macros for RTF templates in a Word template and reference the template from the component file: You can store your post-assembly macros in a template specifically designed for HotDocs. You specify the name of this file at the **Component File Properties** dialog box for the template.

Update tables of contents, cross references, indexes, and fields after sending assembled documents to word processor: You can select a component file property that automatically updates all cross-
references (including the table of contents and index) in an assembled document once the document is sent to the word processor.

New Word menu option for inserting variables, instructions in a Word template: Word users can now more easily insert instructions such as ASSEMBLE, ANSWER FILE NAME, LANGUAGE, and DEBUG in their templates. The option for doing so can be found in the new HotDocs menu in the HotDocs toolbar. (Users can also insert other HotDocs field types (such as variables, REPEAT instructions, and so forth) using this new menu.) Finally, most of these options are also now available in the Word shortcut menu.

Multiple Choice options included in auto-complete lists: When scripting with Multiple Choice variables, you can use the auto-complete feature to access lists of options for the Multiple Choice variable. This keeps you from having to manually enter the options.

Scripting supported in plain-text resources: You can now include variables, IF instructions, and REPEAT instructions in plain-text resources. This allows you to customize the resource text users see in the interview, based on their answers to questions.

Save button added to component editors: As you edit components, you can now save your work without closing the component editor.

Document Assembly

Install support for new word processors without reinstalling HotDocs: In previous versions of HotDocs, if you installed a new word processor, you had to reinstall HotDocs in order to integrate the two products. Starting with HotDocs 2006, you can install support for new word processors at HotDocs Options.

Control font properties of dialog text: You can now specify the font face, size, and color of text used in dialogs. You can also change the font used for items in the interview outline.

Edit the document while viewing the Document Preview tab (Word users only): If the template provider has allowed it, you can edit the text of a document while viewing the **Document Preview** tab. This allows you to make changes to the document and be able to save those changes in the answer file so they can be reapplied if you ever reassemble the document.

View variable names for answers in interview: Sometimes you may need to communicate with a template provider about a specific question in the interview. While answering questions in the interview, you can place your cursor in an answer field, right-click and choose **Variable Name** from the shortcut menu.

Simultaneously copy an answer file to the Answers folder and add it to Answer File Manager: Frequently you receive answer files from other users. You can now open Answer File Manager and choose the **Import** command to copy an answer file to the **Answers** folder and add it to the answer library. You can also double-click on the answer file in Windows Explorer and the file will be added to the library and copied to the **Answers** folder. **Mark required answers in a dialog:** In addition to marking dialogs that contain required questions in the interview outline, you can now mark the actual questions in the dialog. Specifically, you can designate the color used for prompts of required questions as well as control whether an asterisk is used to indicate this.

Follow dialogs in interview outline when moving through interview: A new navigation option, Next Dialog Follows Outline, has been added. Selecting this option will move you to the next dialog in the interview outline when you click Next. For example, if you are in a main-level dialog, clicking Next will move you to any inserted dialogs in main dialog. Clicking Next again will move you to the next main-level dialog, rather than back to the current main-level dialog.

Update published template sets: If you are using a published template set, HotDocs can check for updates to the set at regular intervals and notify you when updates are available.

HotDocs Automator / Filler

HotDocs HFT Driver no longer supported: When creating a HotDocs form template (.HFT) file in previous versions of HotDocs, you had to use the HotDocs HFT driver, which was only supported for use with Windows 98 and Windows Me. Starting with the release of HotDocs 2006, these operating systems are no longer supported. Now, when you create new form templates, you must create them in PDF format.

Look and feel of HotDocs Automator / Filler updated: Several HotDocs Automator and Filler dialog boxes and windows have been updated.

Keep Field Properties dialog box open as you modify field properties: When applying field properties to form fields, you can keep the **Field Properties** dialog box open as you move between fields in the form. This allows you to set properties for a field and then test to see how the properties affect the underlying field.

Circle static text on a form document: You can now create a field that circles preprinted options on a form.

Send just the part of an answer that overflows to addendum (rather than the entire answer): When specifying overflow options for a multi-line edit field, you can select an option that sends just the part of the answer that doesn't fit in the field to the addendum.

New and Enhanced Features of HotDocs 2005

New and Enhanced Features of HotDocs 2005 SP3

Support for WordPerfect 13 added: HotDocs version 2005 SP3 is supported for use with WordPerfect 13.

New and Enhanced Features of HotDocs 2005 SP2

Template Development

Rename multiple components simultaneously: You can now select multiple components at Component Manager and rename them all at once.

Limit the number of rows in a spreadsheet: When a dialog is repeated as a spreadsheet, you can control the number of rows that are displayed in the dialog. (This option simply controls the number of rows that are displayed—users are still able to enter as many answers as they need.)

Map Text variables to Multiple Choice values (and vice versa) in an answer source: When mapping an answer source, you can map a Multiple Choice variable to a text value. You can also map a Text variable to a multiple choice value.

Create an answer source that links a dialog to a Time Matters Contacts or Matters record: You can now link an answer source to a Time Matters Contacts or Matters record. This allows you to retrieve information you're already storing in Time Matters and use that information in your documents. When creating the answer source, you can also designate whether changes to answers can be written back to Time Matters. (To create this answer source, you must be using Time Matters 7.0 or later.)

Document Assembly

Expand and collapse all folders at once in the library: You can now expand or collapse all subfolders in the template library. This makes it easier to quickly view or hide template lists in the library. (This command is available in the library **View** menu.)

Choose a color for additional text: When customizing the appearance of dialogs in an interview, you can now assign a different color to additional text. This may be useful if you want to distinguish additional text from other text in the dialog. You do this at the **Dialog Appearance** folder of HotDocs Options.

Open Windows Explorer from the template library: You can click on a reference to a template in a library and choose **Go To** from the **Template** menu. This opens the folder where the template file is saved.

View resource buttons for all answer fields at once: When viewing a dialog with answer field resources, you can now choose to display the resource button for all answer fields at once, or you can choose to display the button only when you are viewing that specific answer field. (This new functionality changes the available settings at **Tools > Options > Interviews and Dialogs > Show answer field resource button**.)

New and Enhanced Features of HotDocs 2005 SP1

Template Development

Save component files in HotDocs 6 or HotDocs 2005 format: When HotDocs 2005 was released, you had to upgrade your component file to the latest version. You can now designate a component file

property that saves the component file either in version 6 format (so you can continue to use the template with HotDocs 6.x) or in version 2005 format (so you can use all of the functionality of HotDocs 2005).

If you want to use HotDocs 2005 to develop your templates but you want them to be compatible with HotDocs 6, do not use any HotDocs 2005-specific features (such as non-breaking spaces) or the features implemented in this release (described in this topic). If you do, you may see unexpected results.

Changes to Component File Properties dialog box: The **Component File Properties** dialog box has been redesigned and now includes tabs to differentiate between the different types of properties.

Add titles to variables: You can now specify a title, or alternate name, for a variable. Sometimes you use variable names that are useful for you during template development, but not useful for users during an interview. For example, maybe you name your variables with some type of variable notation, like *Client name TE*. To a user, this name, if seen in an interview, may not make much sense. Here, a title (*Client's Name*) can be used in place of the variable name.

Assign a label to the entire repeated series: You can now assign a label to the entire repeated series, which lets you customize the dialog title for each individual repetition in the list.

Copy Multiple Choice options, prompts, and merge text and paste them into other spreadsheets: You can copy data from other spreadsheets or tables and paste it into the Multiple Choice Variable Editor spreadsheet. Similarly, you can copy the contents of a Multiple Choice Variable Editor spreadsheet and paste it into other tables or spreadsheets (including other Multiple Choice variables).

Automatically set repeat style when creating a REPEAT field: When you create a REPEAT instruction in the template (using the REPEAT Field command), HotDocs will automatically suggest **Repeated Series** as the dialog **Style**. Previously, you had to manually specify a repeat style.

Specify merge text on as-needed basis: You can now include merge text for Multiple Choice options on an as-needed basis. Previously, if you included merge text for one option, you had to provide it for all options. However, now you can provide it for only those options that require it.

New repeated list format added: When you create a new template, HotDocs includes **a**, **b** in the example format lists for repeated answers. This will format a series of answers as *apples, oranges, cherries* (excluding any conjunction). To use the format in existing templates, simply type it in the **Format** field, either at the **Multiple Choice Variable Editor** or at the **REPEAT Field** dialog box.

Document Assembly

Sort spreadsheets (including answer sources) during the interview: You can now sort the contents of a spreadsheet in alphanumeric order.

Print contents of the Preview tab at library window: You can select a text or form template at the template library and print a blank copy of it.

View file names in library's item list: You can now view either the template's title in the library list, or you can view the template's file name. Additionally, when searching for a specific template, the search is performed on the title, description, and file name. (Previously, HotDocs would search just the title and description.)

HotDocs Filler

Skip conditioned-out fields during direct-fill assembly: During direct-fill assembly of a form document, HotDocs will now skip fields that are conditioned (and resolved as false).

New and Enhanced Features of HotDocs 2005

Template Development

Database Connection: HotDocs Database Connection is now included with all licenses for HotDocs Developer. Database Connection allows you to retrieve answers from a database.

Changes to HotDocs Server templates and interviews: HotDocs Server 2005 has been redesigned to display browser-based interviews that more closely match desktop interviews. When creating templates for use with HotDocs Server, you must first enable them to be used on the server. (You do this at the **Component File Properties** dialog box.) Once templates are enabled, they will be checked for features that are incompatible with the server or Web browser. Additionally, enabling the template will let you test the template in a simulated browser environment.

To use the new style of interviews, you must regenerate the JavaScript (.JS) files and HotDocs Variable Collection (.HVC) files. You can do this by republishing your templates. (First make sure they are enabled for HotDocs Server.)

Override parent template headers and footers with those in inserted templates: When you insert templates and clauses, you can now select options that allow you to keep the headers and footers in the inserted files, rather than have them overwritten by the parent template's headers and footers. This is useful if you either want to create a master document containing several subdocuments, or you want the headers/footers defined in one of the inserted templates to be the headers/footers for the entire document.

Use symbol fonts for answers in text templates: In text templates, can now assign symbolic character fonts (as well as standard character fonts) for answers in your document. This is useful if users' answers must appear in a non-standard font, such as a bar code.

Use non-breaking spaces and hyphens in variable formats: Often, you need a user's answer to stay on a single line in the assembled document. Now you can assign an example format that will keep answers from breaking at the end of a line of text.

Access a Microsoft Outlook Contacts list: You can now link an answer source to your address book in Microsoft Outlook. This allows you to retrieve information you're already storing in Outlook and use that information in your documents.

Document Assembly

Edit answers directly at the Document Preview tab: When previewing an assembled document, you can choose to have HotDocs identify the answers you have entered. Once you are viewing these answers, you can edit the answers (by double-clicking on the field). (This feature is available to Microsoft Word users only. Additionally, it is only enabled if the template provider has enabled it.)

Navigate through answer fields in the document: If you are viewing either an assembled Word document or an assembled form document, you can use the new Navigation Bar to move between answers fields on the document. Specifically, you can move between all answers, unanswered questions only, or matching answers (for example, just those answers that are the same). Additionally, you can have HotDocs take you to the dialog in the interview where the question is asked. (Moving through fields at the **Document Preview** tab is available to Microsoft Word users only. Additionally, it is only enabled if the template provider has enabled it.)

Compare different versions of a document at the assembly window: When HotDocs Compare is installed, you can answer questions in the interview one way, save a snapshot of the document, and then answer questions differently so you can compare versions of the document. Additionally, you can save multiple versions of a snapshot to use in the comparisons you are doing. (HotDocs Compare is available to Microsoft Word users only.) NOTE: Starting with the release of HotDocs 2009, HotDocs Compare is no longer available.

Copy templates to new locations on disk: In addition to moving templates using the **Move Templates** command, you can copy templates to new locations using the new **Copy Templates** command. When you copy templates, you create a copy of the template in a new location. Additionally, you can designate that the copied file be marked as read-only.

New and Enhanced Features of HotDocs 6.2

New and Enhanced Features of HotDocs 6.2 SP1

This release of HotDocs contains software corrections and minor enhancements to existing features.

New and Enhanced Features of HotDocs 6.2

Default File Locations

Several changes were made to default file locations HotDocs uses.

Template Development

New word processor support: HotDocs version 6.2 is now supported for use with WordPerfect 12.

Troubleshoot problems in templates or scripts: You can now insert a DEBUG instruction either in your scripts or in your templates to help diagnose problems with automation. The HotDocs Debugger lets you step through a template field by field, or through a script line by line. It also lets you track variable answers as they change during the interview. Finally, it lets you view the sequential list of templates and components you are processing. This can help you understand the path of execution.

Repeatedly loop through an answer or set of answers: The WHILE EXPRESSION instruction allows you to repeatedly process (or loop through) an answer or set of answers until a certain condition is met, such as a certain answer is found or a limit is reached.

Increment temporary counters using new instruction: You can use the INCREMENT and DECREMENT instructions in a HotDocs script to increment or decrement a counter. Previously, this was done using SET instructions such as *SET Num TO Num* + 1. These instructions are useful when incrementing counter variables used in WHILE instructions.

Component Explorer redesigned and renamed to Template Manager: The purpose of Component Explorer was expanded to include template management features, such as template conversion and renaming (see below). This necessitated a change to the name of the tool. It is now called Template Manager.

Convert templates to HotDocs 6 format or to RTF: Use Template Manager to convert large batches of templates or clauses either from HotDocs 5 format to HotDocs 6 format. You can also convert WordPerfect or Microsoft DOT templates to the faster RTF format.

Rename templates: Use Template Manager to rename templates in a library. When renaming templates using this method, all associated files (such as component files, clause libraries, and so forth) will be renamed, as well. References to the renamed file will be updated throughout the library.

New template navigation toolbar added to text templates: A new toolbar has been added to word processor templates that allows you to more quickly work with fields in the template. Specifically:

- HotDocs now assigns different colors to the different types of fields in a template using the Apply Colors button. Specific fields include variable fields, IF instructions, REPEAT instructions, ASK instructions, and comments. You can customize the colors used at HotDocs Options.
- You can use the new **CLabel Fields** button to label matching IF and END IF instructions as well as matching REPEAT and END REPEAT instructions. Additionally, where you've nested these instructions, you can identify the level of nesting. Finally, you can assign the word processor's hidden text property to these comments and labels so that you can show and hide them during automation.
- You can use the new Add the set of the set

• You can use the new **Next Field** and **Previous Field** buttons to move from one field to the next. This is particularly useful when you have several pages in your template that do not have any automation and you don't want to scroll through these pages manually.

Double-click in Word fields to edit them: You can now double-click in a Word field to bring up the **Variable Field** dialog box for that variable. (This requires Word 2000 and above.)

Control whether dialogs are asked automatically: The Dialog Editor now includes an **Ask automatically** option, which, when cleared, causes HotDocs not to ask the dialog when it is used in a REPEAT instruction or in an expression that refers to the dialog's True/False status. This eliminates much of the need for ASK NONE instructions.

Add comments to template fields: You can now add comments to the end of fields. After creating the field in your template, insert your cursor after the component name (or instruction), type two forward slashes, and then type your comment, for example, *«Employee Name //comment»*.

Keep the End of Interview dialog from appearing in users' interviews: You can select a component file property that keeps the End of Interview dialog from appearing in the interview. When the user clicks
Next at the last dialog in the interview, HotDocs will either send the assembled document to the word processor or it will display the **Document** tab, depending on which End of Interview action the user has defined at HotDocs Options.

Sort components in Component Manager: You can now sort lists of components by type as well as alphabetically. In Component Manager, sorting is done using the new 2 **Sort Components** button. In other component lists, sorting options are found on the shortcut menu.

Edit components in second component list: If you have two component files open in Component Manager, you can now edit components in both component files. Previously, you could only modify components in the left pane.

Online test a template from the template—not the library: You can online test templates directly from the template. (Previously, you had to close the template and test from the template library.) To do this, press the **Shift** key as you click the **Test Assemble** button.

New unanswered variable placeholder added: A new unanswered variable placeholder has been added. It inserts the variable name between square brackets, for example **[Variable]**. You can choose this option either at the **Component File Properties** dialog box or at **HotDocs Options**.

Generate templates from marked up documents: Markup editors can mark up a document for automation using the HotDocs Markup Tool. Once marked up, you can use the Template Set Generator to convert the documents to HotDocs templates. (Contact your HotDocs sales representative for details on purchasing a license for the Markup Tool and Template Set Generator.) (With the release of HotDocs 2008, support for the Markup Tool and Template Set Generator was removed.)

Document Assembly

Rearrange entries in a repeated list, and remove or add entries any place in the list: When entering sets of answers in a repeated dialog, you can now move entries around in the list as well as add or remove entries.

Changes to *End of Interview* **dialog:** The following changes have been made to improve the End of Interview dialog:

- You can choose which buttons appear in the *End of Interview* dialog. You make your selections at HotDocs Options.
- You can choose to keep the *End of Interview* dialog from being displayed during the interview. (When hidden, clicking **Next** at the last dialog in the interview either sends the document to the word processor or Filler, or switches the view to the **Document** tab. You choose which option you prefer at HotDocs Options.)
- Buttons for each option have been enlarged and arranged in the order they are most frequently used.
- A new button was added that, when clicked, pastes the assembled document into the current word processor. (This is useful when assembling clause documents.)
- A new button was added that, when clicked after assembling a form document, moves you to the Form Document tab of the assembly window. (This allows you to view the assembled form document in the assembly window where changes you make will be reflected in the answer file.) Before clicking this button, you can choose to have HotDocs check fields in the form for any answers that overflow.
- A new button was added that, when clicked, lets you choose which *End of Interview* buttons you want in the dialog.

Finish Interview button added to navigation bar: A new button was added to the navigation bar which allows you to skip any remaining questions in the interview and view the assembled document. You can choose whether to view the document at the **Document** tab or whether to send the document to the word processor or HotDocs Filler. HotDocs will perform this same action when the *End of Interview* dialog is hidden and you navigate past the last dialog in the interview.

Send assembled documents to the word processor that matches the template type: You now have two options for sending the assembled document to the word processor—you can either send to the default word processor always or send to the word processor that matches the template type. You select your preference at HotDocs Options.

New unanswered variable placeholder added: A new unanswered variable placeholder has been added. It inserts the variable name between square brackets, for example **[Variable]**. You can choose this option at HotDocs Options. All variables that do not have a specific placeholder assigned will use this placeholder.

Save As option added to Save Answers dialog box: Now, when you close the assembly window, you have the option of saving existing answers to either the current answer file or a new answer file.

Choose where assembly window tabs are placed: You can now choose to have the assembly window tabs (Interview, Document Preview, etc.) displayed either along the top or along the bottom of the assembly window.

HotDocs Automator and Filler

Span a text field across multiple form pages: Often, a text answer needs to flow from one field to another. In most situations, you can group these related fields as a run-on group and when the answer won't fit in the first line, it will flow to the next line. However, when the fields span across two or more pages, HotDocs will not allow you to group them. You can, however, use the **Named Group** overflow property to span a text field across multiple pages.

Control when form field answers are updated: You can now use the **Instant Update** button to control when HotDocs updates answers in form fields. When selected, HotDocs updates all fields whenever an answer changes. When cleared, HotDocs updates a field only when it is tabbed to or out of.

New PDF printing options added to HotDocs print function: When printing a nonstandard-sized PDF form using the HotDocs print function, you can choose the paper size that most closely matches the page size, or you can scale the content of the page to fit the printable area.

New and Enhanced Features of HotDocs 6.1

New and Enhanced Features of HotDocs 6.1 SP1

Template Development

Auto complete while typing component names: As you are creating components, HotDocs can now automatically complete the component name if what you are typing matches the name of an existing component. This was implemented to help you distinguish between new and existing components. (You can disable this feature at **Tools > HotDocs Options > Template Development**.)

Default Multiple Choice variable to multiple values: Multiple Choice variables that have the **Select All That Apply** property set can now be defaulted to multiple values using the **Automatically select this option** check box at the **Options** tab.

SET Multiple Choice variable to multiple values: Multiple Choice variables that have the **Select All That Apply** property set can now be SET to multiple values using an instruction (for example, *SET MC Variable TO "Option 1|Option 2|Option 3"*). (You must use the vertical bar character as the separator between options.)

LIMIT allows numeric expression: The LIMIT instruction now accepts a numeric expression. (Previously you had to use a single variable to specify the limit.)

Force a paragraph break in text answers: Text variables now include a property called Enter key in multi-line answers inserts new paragraph mark (1). This allows you to control what kind of break is inserted into multiple-line text when the user presses Enter. Because of this change, the option Enter key action in multiple-line fields was removed from HotDocs Options.

Drag variables from Component Manager to Automator: You can now drag variables from Component Manager and drop them directly on a form field in HotDocs Automator. If no field exists where the dropping occurs, a default field will be created.

Document Assembly

Bypass assembly window: You can now select an option at the **Answer File** dialog box at the beginning of an interview that allows you to skip the assembly window and immediately view the assembled document.

Check spelling of answers: You can now spell check answers at the **Interview** and **Form Document** tabs of the assembly window.

New and Enhanced Features of HotDocs 6.1

Template Development

New word processor support: HotDocs version 6.1 is now supported for use with Microsoft Word 2003 and WordPerfect 11.

Use the new script editor: Several enhancements were made to the way you write computation, dialog, and expression scripts. These include customizable color coding for the different keywords and placeholders in a script, as well as **Auto Complete** functionality that allows you quicker access to instruction and expression keywords and component names. Additionally, you can more easily 'comment' blocks of script, and match IFs with END IFs (and REPEATs with END REPEATs), among other things.

Additionally, now, when including literal text strings in computation scripts, it is much easier to insert return characters as well as tabs.

In previous versions of HotDocs, returns produced from computations were treated as paragraph marks. However, in HotDocs 6.1, they are now treated as line breaks. You should check existing automation in your template to make sure this change does not create problems.

Support for HotDocs Database Connection 6.1: The HotDocs Database Connection has been updated to work with HotDocs 6.

Control answer field widths: Template developers can now specify answer field widths for both regular answer fields and spreadsheet column widths. These changes are made at each variable's Variable Editor. (Make your changes at the **Advanced** tab.)

Tighter integration with document management programs: Those integrating with document management systems now have greater control over the format of their saved answer files and assembled

documents. These options are specified at the HotDocs Options dialog box, in the **File Management** folder (and the subsequent **Advanced ODMA Settings** folder).

Customize the library Properties tab: HTML-savvy developers can customize the Properties tab of the library window using custom HTML pages. Properties pages can be customized for individual files in the library, for a single library as a whole, for the main folder in a library, or for all libraries.

Duplicate button improved: The **Duplicate** button functionality (in Component Manager) has been improved. Template developers can duplicate a single variable, or they can 'batch' duplicate several variables at once.

Better control over inserted templates: HotDocs no longer tries to enforce the requirement that inserted files (such as templates and clause libraries) must be in the same folder as the host template. If the inserted file exists in a different folder, however, the path to that folder must be specified in the instruction.

Changes to CLEAR and ADD: When using the CLEAR and ADD instructions to populate a Multiple Choice variable, you can also ADD prompts for the options.

@COMPUTE and @EXECUTE supported: The instructions, @COMPUTE and @EXECUTE are now officially supported for use in dialog components.

Publish templates with ASSEMBLE instructions: When publishing templates, you can now select to automatically publish any templates used in INSERT and ASSEMBLE instructions. (Previously, HotDocs would include templates specified only in INSERT instructions if this option were selected.)

Add templates to a library using drag-and-drop: In addition to using the **Add Template** button or command to add templates to the HotDocs library, you can now drag templates (text, form, and interview) from Windows Explorer to the HotDocs template library list.

Copy components both ways using Component Manager: When you are displaying a second component list at Component Manager, you can now copy components from the current component list into it. You can also delete and rename components in this second list.

Document Assembly

Selecting child dialogs: Selecting grouped child dialogs in an interview is now more intuitive—you can either select the preceding check box (or option button) and then click the child dialog icon to display the dialog, or you can click the child dialog icon immediately. Doing this selects the check box (or option button) for you. (Selecting the check box or option button is imperative because it tells HotDocs to add the child dialog to the interview outline. It also lets you test whether a child dialog has been answered, as well as set a child dialog's answered status to true or false.)

Viewing resources: Resources assigned to both variables and dialogs now appear simultaneously in the Resource pane. (Before, either the variable or the dialog had to have 'focus' for its corresponding resource to be visible.)

Improvements to overall assembly process: Several changes have been made to improve interview speed as well as dialog navigation.

Customize dialog pane colors: You can now change the background color of the dialog pane in the assembly window. These changes are specified at the **HotDocs Options** dialog box.

Better document preview for Word templates: When previewing a Word template or assembled document, you can now get a much truer view of the document.

HotDocs Automator and Filler

Start HotDocs Automator from library window: You can now start HotDocs Automator from the **Tools** menu of the HotDocs template library.

New tools: HotDocs includes a new **Fill Fields** tool, which allows you to type text directly in the field. Improvements were also made to existing tools.

Creating form templates: If you create a new form template from Automator, HotDocs will first ask which type of form you want to create—a HotDocs form template (HFT) or a HotDocs PDF template (HPT).

Handling form overflow: When you print an assembled PDF-based form document, and the document contains unresolved answer overflow, HotDocs gives you three options for resolving the overflow: 1) review and change the answers or field properties, 2) send all overflowing answers to the addendum, or 3) ignore the overflow and leave the answer truncated.

Viewing thumbnails: If you are viewing thumbnails in HotDocs Automator or HotDocs Filler, HotDocs highlights the page number of the form page being viewed. This makes it easier to identify the thumbnail for the page you are currently viewing, especially in a form that contains many pages.

Send addendum to word processor: If an addendum is created during assembly, you can now send the addendum directly to the default word processor, rather than the Clipboard. (This option is only available from the HotDocs assembly window. It is not available in HotDocs Filler.)

New and Enhanced Features of HotDocs 6

New and Enhanced Features of HotDocs 6.0

Template Development

Use the new template library: The new template library contains several additional buttons and new menu items that allow you to perform multiple template development and document assembly tasks, including managing the contents of the library more efficiently. The toolbar and menus also make it easier

to access other HotDocs tools, including Component Manager, HotDocs Options, Answer File Manager, Component Explorer, and the Assembly Queue.

The library window is divided into two panes: the left pane shows the list of templates, clause libraries, and other library items, while the right pane shows tabbed views of either the item's properties (such as the file type and title) or a preview of the library item.

Create a new template: HotDocs gives you more control over the type of template you want to create by including a **Type** drop-down list at the **New Template** dialog box. This lists all the different template formats available, based on the supported word processors you are currently using. Template titles are automatically generated based on template file names; however, you can change the title to something more descriptive.

Use the new HotDocs editing toolbar: The look of the existing HotDocs editing toolbar has changed, and three new buttons have been added:

- The **Clause Library** button displays the clause library associated with the template. From here, you can create clauses, edit existing clauses, remove clauses, and perform several other tasks relating to clause management.
- The **dit Component** button bypasses the **Variable Field** dialog box and lets you edit the component properties of a variable while at the template.
- The **HotDocs Help** button makes it easier for you to access information in the Help files directly from the word processor window.

Create variable fields and variable components: Now when you create variables, there is greater distinction between the variable field and the variable component. For example, when you first create a variable at the template, the **Variable Field** dialog box appears, where you can assign a variable type, name, and any field-specific properties. Then, from the **Variable Field** dialog box, you can click the **Edit Component** button to display the **Variable Editor**. At this dialog box, you can assign component-specific properties, such as prompts and resources.

Assign either default or field-specific formats: You can assign answer formats, field formats (such as field widths, alignment, and fill characters), and merge text (for Multiple Choice variables) either at the **Variable Field** dialog box, or the **Variable Editor**. If you assign them at the **Variable Field** dialog box, they are available for that specific instance of the variable only. If you assign these properties at the **Variable Editor**, they become a default property of the component and will be available each time you use the variable in the template.

Use Multiple Choice variables: At the **Multiple Choice Variable Editor**, you can now provide prompts for each multiple choice option. Also, if you want to further customize each row of options, you can do so by clicking the **Options** tab and making your changes. These include adding longer prompts and merge text, and pre-selecting options for the user.

Assign resources (formerly known as variable and dialog help): The name for this type of help has been changed to *Resources* to better distinguish between HotDocs Help (the documentation provided

with the software) and template developer help (the help assigned by you). You also can now use an HTML Help (.CHM) file as a source file for variable and dialog resources.

Use the new Component Manager: With the enhanced distinction between fields and components, Component Manager becomes the preferred method for editing components in a template, including variables and dialogs. Component Manager is now a sizeable dialog box that can be arranged next to the template window so that you can work in both windows simultaneously. Using Component Manager, you can edit as many components at a time as you want. You can also drag variables from Component Manager and drop them directly in the template.

Share components between component files: The process for sharing component files has been improved. For example, the **Pointed Component File** button in the Component Manager toolbar makes it more apparent a component file is pointed (for example, ***** versus *****), and the actual process of finding the component file and pointing it is much more intuitive.

Change component file properties: At the **Component File Properties** dialog box, you can now specify a product title for the template you are creating. You can also have HotDocs hide the interview outline for the template during assembly as well as have HotDocs generate default dialog titles. The interface for several existing properties has changed, as well.

Copy components using Component Manager: You can expand the Component Manager window to show another component file's list of components, from which you can copy into the current component file. (You can no longer copy out of the current component file, however.)

Use IF, INSERT, ASK, and REPEAT instructions: All of the dialog boxes for inserting IF, INSERT, ASK, and REPEAT instruction fields have changed. At each of these dialog boxes, you can choose a specific type of field you want to merge, and then edit the underlying component.

Use clauses and clause libraries: The process of creating clauses and clause libraries is now much easier. You can open a clause library from the template-editing toolbar, and it can stay displayed as long as you need, allowing you to work in both the template and the clause library simultaneously.

Use Dialog Editor to edit dialogs: The Dialog Editor includes tabs for customizing a dialog. A **Find** option has also been added to make searching for specific variables, clauses, or dialogs easier. When you are at the **Script** tab of the Dialog Editor, you can click the **Variables** drop-down button and select **Variables in Dialog** to see only the variables used in that specific dialog. (This makes it easier to create dialog scripts since you can limit the list of variables to show only those used in the current dialog.)

Assign titles to dialogs: You can now assign titles to dialogs. The dialog title replaces the dialog name in the interview outline.

Change the layout of a dialog: In earlier versions of HotDocs, to change the placement of variables in a dialog, you had to test the dialog and make your changes directly in the test dialog. Now, you can click the **Layout** tab of the Dialog Editor and drag-and-drop variables there.

Simultaneously test and edit variables, dialogs, and scripts: You can simultaneously test and edit components within a template (such as variables, dialogs, and scripts). When you do, HotDocs displays

the variables and any corresponding answer fields in a test assembly window. You can leave that dialog open and make changes to the component and then update the dialog to see the changes implemented. Once you click OK at the Variable Editor, the changes are saved to the component file.

Simultaneously test assemble and edit a template: You can test assemble all or a portion of a template while simultaneously making changes in the underlying template text. HotDocs will update the assembly window with changes you make.

Save answers used for testing once: When you test a variable or test assemble a document, you are prompted to save your answers. Once you save them (using *Test Answer File* as the file name), HotDocs uses that test answer file for each subsequent test assembly. If needed, you can specify a new answer file or use a different answer file at the assembly window.

Use Component Explorer: The Component Explorer tool lets you more closely manage components across a large number of component files, including removing unused components, renaming components in both the component and template file, and copying and pasting components between files.

Document Assembly

Choose an answer file for assembly: When you first select a template for assembly, HotDocs displays the **Answer File** dialog box, which allows you to choose an answer file to use with the assembly. (You no longer choose any other assembly options, such as assembling Question/Answer summaries, or viewing only unanswered variable questions. All of those options are now controlled directly in the assembly window.)

View the new assembly interface: Once an assembly has started, the assembly window appears. By default, it is divided into three panes: the interview outline (or left pane) lists all the dialogs and variables asked in the template. When users click one of these items, the corresponding dialog appears in the dialog pane (or right pane). Users enter the required information in the answer fields and proceed through the interview until all the answers have been given. The interview outline is dynamic so items can be added and removed, depending on answers the user provides. Finally, if the template developer has provided helpful information about a dialog or a specific variable, that information appears in the third pane, the resource pane (below the dialog pane).

Navigate an interview: You can click items in the interview outline in any order to complete the interview. You can also click buttons in the navigation bar at the bottom of the dialog pane.

Use the new assembly toolbar buttons: The assembly toolbar buttons let you perform several different tasks, such as work with answers and answer files, send the assembled document to a word processor (so you can edit it), view the Assembly Queue, access HotDocs Options, view the resource pane, and perform other general tasks.

Use the tabbed views of the assembly window: The assembly window contains several tabs that let you change your view of the assembly process. You can view any of these tabs at various times during assembly and the information contained therein will be current. These tabs include:

- **Interview** displays a three-paned window that includes the interview outline, the dialog pane, and the resource pane.
- **Preview** shows a rough preview of the document so far as it has been assembled.
- Question Summary displays an HTML-based summary of the questions in the template.
- **Answer Summary** displays an HTML-based summary of the questions and answers provided during the interview.
- **Variable Sheet** displays a spreadsheet-like summary of variables, prompts, answers, and variable types.

For the most part, the contents of these different tabs can be saved as word processor or HTML documents, and can be attached to e-mail messages.

View the End of Interview dialog: When you complete an interview, the *End of Interview* dialog appears. It contains information about unanswered variables as well as gives instructions and options for working with the assembled document. (The End of Interview icon always appears in the interview outline.)

Use the resource pane: HotDocs now displays template developer's resource information (formerly known as variable or dialog help) in the resource pane of the assembly window. By default, this pane is continually displayed; however, you can hide and show it as needed. You can also specify an option (at HotDocs Options) that displays a resource button next to answer fields that contain resources.

Preview the assembled document: As you complete an interview, you can click on the Preview tab to view your document during assembly. Any answers you have given are merged into the text, while any answers that are still needed are marked by unanswered variable placeholders. You cannot edit the text in the Preview tab, but once you finish an assembly, you can send a copy of the document to the word processor for post-assembly editing.

Because of the way HotDocs renders assembled text in the **Preview** tab, the document sometimes will not show the formatting you see when you send a copy of the assembled document to the word processor.

Use Answer File Manager: The Answer File Manager has a newly designed interface (including a new toolbar) that makes it easier to view the properties as well as the contents of an answer file.

Change answer files during assembly: As you assemble a document, you can change the answer file you are using. When you do, HotDocs lets you save the current answer file and assembled document before you load a new answer file.

Attach answer files, assembled documents to e-mail messages: You can attach answer files, assembled documents, and question and answer summaries to e-mail messages.

Manage multiple assemblies using the assembly queue: When you select multiple templates for assembly, HotDocs automatically displays the **Assembly Queue** dialog box, which lists each template and its assembly status. Using this dialog box, you can change the order in which documents will be assembled, as well as add assemblies to and remove them from the queue.

Change your HotDocs Options (formerly known as Settings or Preferences): HotDocs now provides one central location- the HotDocs Options dialog box -to control how HotDocs operates. General options include template development, interview, and document assembly preferences, as well as file location management.

HotDocs Automator/Filler

Changes to HotDocs® Automator: Some changes have been made in HotDocs Automator to more tightly integrate it with the new HotDocs 6 interface. These include new toolbar buttons and menu items, as well as minor changes in various dialog boxes. Additionally, the default field colors have changed, and several pre-designed color schemes have been provided.

Changes to HotDocs® Filler: Some changes have been made in HotDocs Filler to more tightly integrate it with the new HotDocs 6 interface. These include some new toolbar buttons and menu items, as well as minor changes in various dialog boxes. Additionally, in the **Document** tab of the assembly window, you can type your answers directly in the form fields. If the template developer has created a custom interview (or allowed HotDocs to generate a default interview), you can move between the **Interview** and **Document** tabs to complete the interview.

Installing HotDocs

Installing HotDocs: Quick Start

Please see the installation information in the HotDocs Developer Quick Start Guide.

Becoming Familiar with HotDocs

HotDocs Overview

HotDocs is a suite of award-winning software applications that significantly reduce the time you need to spend generating client or customer specific documents (such as contracts, sales proposals, government and court forms, loan applications, and medical forms).

HotDocs' document generation software solutions form an array of products, environments, and platforms that enables users across a wide variety of work situations to take advantage of HotDocs' automated customization capabilities.

Using HotDocs, you can transform any word processor file or form into an interactive template by replacing the changeable text with HotDocs variables. Then, the next time you want to generate a completed document, just assemble the interactive template you've created. As you do this, you will be prompted for the information needed in the document and that information will be merged into the document.

When using HotDocs, you can perfect a template, minimizing the "human error" factor that repetitive typing introduces. Additionally, you can automate your templates so that verb tenses, gender references, dates, numbers, calculations, and more are updated automatically as users enter information. Custom interview questions and help resources guide you or your users through the interview.

Virtually any document in your workflow can be converted to a HotDocs template. HotDocs templates then become your gold standard—experienced colleagues can share their knowledge, and new colleagues come up to speed faster. Using HotDocs, you can compile an invaluable repository of special language, unique clauses or terms, correspondence, and more.

This document provides an overview of:

HotDocs Document Generation Workflow

HotDocs document generation follows the following workflow:

- 1. You (or another template developer) use HotDocs software products and systems to produce a generic exemplar document known as a template. (You can do this most easily by replacing an existing document's changeable text—text that varies from one client to another, such as names, dates, and institutions—with HotDocs variables).
- You tell HotDocs to use the template you create to generate an interactive interview that asks your end user (a client or customer) for the data HotDocs needs to create a customized document.
- 3. HotDocs presents the interview to your end user either in a desktop product, or in a browser.
- 4. Your end user steps through the interview, providing the information you request at the prompts you set up in the template's dialogs.
- 5. HotDocs stores this information in an answer file.
- 6. When your user finishes the interview, HotDocs assembles a final document by merging the information in the answer file with the non-changing text in the template.

HotDocs can also use existing answer files to supply an interview with commonly reused information (for example, a company's address and contact information).

The workflow diagram below shows HotDocs' capability to use both answer files (saved answers) and end user input to provide information in an interview. The diagram also shows that HotDocs can generate an answer file or an assembled document by merging the template text with the answers from the completed interview.

Getting Started



HotDocs Document Generation Workflow

Using HotDocs, you can perfect a template, minimizing the "human error" factor that repetitive typing introduces. Additionally, you can automate templates so that verb tenses, gender references, dates, numbers, calculations, and more, update automatically as users enter information. Custom interview questions and help resources guide you or your users through the interview.

You can convert virtually any transactional document in your business to a HotDocs template. HotDocs templates then become your standard for document generation—experienced colleagues can share their knowledge, and new colleagues come up to speed faster. Using HotDocs, you can compile an invaluable repository of special language, unique clauses or terms, correspondence, and more.

HotDocs Product Matrix

HotDocs provides the products described in the product matrix below to enable you to accomplish the steps of the HotDocs document generation workflow described above. Which product or products you use to accomplish the steps depends on your particular computing circumstances and needs. You can gain a better understanding of a particular product or feature by locating it within the conceptual areas which define the matrix:

- Tools layer (template development)
- Content layer (completed templates containing non-changeable text and HotDocs variables)
- Environment (desktop, client/server, or cloud computing)

- Platform layer (desktop HotDocs products, HotDocs Server, or HotDocs Cloud Services)
- Communication layer (communications protocols)
- User layer (information gathering and final document generation)



HotDocs Product Matrix

Tools Layer

The tools layer of the HotDocs product suite is the area you use to create a HotDocs template. The tools layer consists of the following:

 HotDocs Developer—a desktop application that enables a template developer to transform a word processor file into a HotDocs template by replacing the document's changeable text with HotDocs variables

HotDocs Developer LE is also available for those creating word processor-based templates without using scripting, and for those not interested in publishing their templates for others to use.

- HotDocs Automator—an aspect of HotDocs Developer that enables a template developer to transform a graphical form into an interactive template by replacing the form's changeable text with HotDocs variables
- Models markup—a simple HotDocs document markup notation system that provides non-HotDocs domain experts a means to mark up or edit a word processor document so that HotDocs Developer can transform the marked up document into a HotDocs template

Content Layer

HotDocs templates form the content layer of HotDocs. A completed template acts as a generic or exemplar document you can use as the basis for drafting documents of the same type for various specific clients. To change the template to a client specific document, HotDocs needs to gather user information and then merge that information with the non-changeable text in your template.

Environment

To accomplish the task of gathering the user information needed to assemble a final document, HotDocs' product suite includes data-gathering solutions for three separate computing environments:

- Desktop
- Client/Server
- Cloud

These environments serve as metaphors for specific software platforms that enable HotDocs to deliver data-gathering and document assembly functionality to end-users, regardless of their local computing configurations.

Platform Layer

HotDocs provides software solutions in the desktop, client/server, and cloud environments to collect client data and assemble client specific documents. These solutions form three independent but interrelated computing platforms for gathering and processing your clients' data to produce custom documents.

HotDocs provides three platform choices in the platform layer:

- Desktop HotDocs Platform
- HotDocs Server Platform
- Cloud Services Platform

You can use products from each of these platforms in any combination that enables you to meet both your and your clients' needs.

Desktop HotDocs Platform

The Desktop HotDocs Platform is not a product; rather, as the name implies, the desktop platform consists of products and communication protocols that you and your clients install on regular desktop computers or laptops.

The base product for the desktop platform is HotDocs Developer or HotDocs Developer LE. HotDocs Developer enables you to create text-based templates using either Microsoft® Word or Corel® WordPerfect®. You can also use HotDocs Developer to create a graphic-based form template. HotDocs Developer LE provides the same basic template creation ability as HotDocs Developer, but does not contain certain advanced features available in HotDocs Developer, such as dialog scripting and answer file linking.

In addition to creating templates, you can also generate interviews using HotDocs Developer (and answer interview questions yourself). However, your end-users in the desktop platform area usually use one of the products in the User Layer of the product matrix to answer interview questions.

HotDocs Server Platform

HotDocs Server is the name, both of the platform, and of the base software product upon which the platform depends. In contrast to the desktop HotDocs platform, the HotDocs Server platform requires a client/server configuration with a dedicated server to run HotDocs Server and a separate client interface (in the User Layer of the platform) for displaying interviews to your end users. HotDocs Server is the server-based version of HotDocs that enables you to use a standard web browser to generate and display interviews and, using the answers from the interview, assemble your end-user's final customized documents.

Cloud Services Platform

HotDocs Cloud Services is the name, both of the platform, and of the base software product upon which the platform depends. In contrast to the HotDocs Server platform, the Cloud Services platform does not require you to install any HotDocs software. The advantage of the cloud environment is that it frees firms from needing to install and maintain server software (or for that matter, any client software in the User Layer). Instead, Cloud Services exists as a HotDocs-provided service to which any third-party can make web service calls to provide their end-users access to HotDocs generated interviews, and to deliver to those same end-users their final documents.

Communication Layer

The communication layer of the product matrix consists of the communications protocols which enable interactions between the User layer and the parts of the HotDocs system that make up the platform layer of the HotDocs product matrix. These communications are necessary to present interviews to end users to gather the information needed to assemble final client-specific documents. The communication protocols used vary with the platform.

Desktop Environment

In the desktop paradigm, the Desktop API is the communication model you use if you want to integrate HotDocs functionality into your own, or third-party, applications. A typical use includes passing data from a third party database into HotDocs. You can also use the plug-in API to, for example, add new menu items to the HotDocs Developer library window.

HotDocs Server Environment

The HotDocs Server Platform enables you to choose several modes for communicating between HotDocs Server and your client application (a HotDocs Server host application):

- HotDocs Server .NET API—a set of native .NET classes that wrap COM functionality; these enable communication between HotDocs Server and HotDocs Server host applications on the ASP.NET framework
- HotDocs Server COM API—Provides a set of COM objects that enable communications between HotDocs Server and HotDocs Server host applications
- HotDocs Server JavaScript API—Enables you to directly integrate a browser interview in your HotDocs Server host application web page by providing direct and synchronous communication with the interview runtime (which must either be loaded in the same frame, or have the same domain name loaded by the interview frame)
- HotDocs Server Web Service API—Enables communication between HotDocs Server and a HotDocs Server host application (using web services)
- HotDocs Open SDK—An open source communication protocol for HotDocs Server or HotDocs Cloud Services; many Open SDK API calls receive quicker responses than those in the alternative communication protocols above

Cloud Services Environment

- HotDocs Cloud Services .Net API—a set of native .NET classes that enable communication between HotDocs Cloud Services and HotDocs Cloud Services host applications using the ASP.NET framework
- HotDocs Cloud Services Direct JavaScript API—Enables you to directly integrate a browser interview in your HotDocs Cloud Services host application web page by providing direct and synchronous communication with the interview runtime (which must either be loaded in the same frame, or have the same domain name loaded by the interview frame)
- HotDocs Cloud Services Embedded JavaScript API—Enables you to place HotDocs interview functionality within an IFRAME in your HotDocs Cloud Services host application web page and use the EasyXDM library for asynchronous communication between Cloud Services and the interview in a cross-domain-safe manner
- HotDocs Open SDK—An open source communication protocol for HotDocs Server or HotDocs Cloud Services; many Open SDK API calls receive quicker responses than those in the alternative communication protocols above

User Layer

The user layer of the HotDocs product matrix is where the work of presenting an interview to a user, then collecting user information in an answer file occurs. Once collected, HotDocs can use the answers (or access answers previously stored in third-party data sources) and the non-changing text in the template to assemble the final client-specific document. The products and solutions for accomplishing these tasks vary with the platform.

Desktop Environment

The desktop platform provides the following products in the user layer:

- HotDocs User—enables you or your end users to complete interviews and assemble text documents from existing HotDocs templates
- HotDocs Player—a more basic version of HotDocs user that enables you or your end users to complete interviews and assemble text documents from existing HotDocs templates
- HotDocs Filler—enables you or your end users to complete interviews and assemble documents from existing HotDocs form templates
- Third party integrations—enable you or your end users to complete interviews and assemble text and form documents with HotDocs functionality provided through the Desktop HotDocs API.

HotDocs Server Environment

The HotDocs Server platform provides several HotDocs-created examples of the host application you need to give your end-users access to HotDocs interviews. Many customers prefer to build their own host applications to customize their users' experience. The following solutions are available:

- HotDocs Workspace—a host application available from HotDocs that enables you or your end users to complete interviews and assemble documents within a web browser; available for use with HotDocs Server and HotDocs Cloud Services; uses a SQL backend to store metadata about uploaded templates
- HotDocs Template Portal—a host application available from HotDocs that enables you or your end users to complete interviews and assemble documents within a web browser; available for use with HotDocs Server and HotDocs Cloud Services; uses a SQL backend to store metadata about uploaded templates
- HotDocs Sample Portal—a sample host application designed to show how to build a custom host application for working with HotDocs Server or HotDocs Cloud Services
- Third party host applications—enable you or your end users to complete interviews and assemble text and form documents with HotDocs functionality provided through the various APIs delineated in the Communication layer

Cloud Services Environment

The HotDocs Cloud Services platform provides several HotDocs-created examples of the host application you need to give your end-users access to HotDocs interviews. Many customers prefer to build their own host applications to customize their users' experience. The following solutions are available:

- Document Services—a HotDocs Software-as-a-Service (SaaS) host application
- HotDocs Market—an ecommerce platform offering expertly crafted HotDocs templates published by bar associations and other legal industry experts.
- HotDocs Workspace—a host application available from HotDocs that enables you or your end users to complete interviews and assemble documents within a web browser; available for use with HotDocs Server and HotDocs Cloud Services
- HotDocs Sample Portal—a sample host application designed to show how to build a custom host application for working with HotDocs Server or HotDocs Cloud Services
- Third party host applications—enable you or your end users to complete interviews and assemble text and form documents with HotDocs functionality provided through the various APIs delineated in the Communication layer

HotDocs Developer Overview

This document provides an overview of:

HotDocs Developer Defined

HotDocs Developer is the desktop-based software application you use to automate any of your textual or graphical (PDF) documents, transforming them into an interactive HotDocs template. You transform an existing document into a template by replacing changeable text in the document (text that varies from client to client, such as names, dates, and institutions) with HotDocs variables.

A template enables you to generate transactional documents and forms, such as contracts, wills, trusts, insurance forms, etc. After you create a HotDocs template, you can use that template to create any number of custom documents. To do so, you use Developer (or, alternatively, HotDocs User, HotDocs Player, HotDocs Server or HotDocs Cloud Services) to generate a HotDocs interview that asks your end users for the information required to replace the HotDocs variables acting as placeholders in your template. HotDocs stores the answers your end user supplies at the interview prompts within an answer file. Once the user completes the interview, HotDocs merges that information with the non-changeable text in your template to assemble a final, customized document for your end user or client.

Using HotDocs Developer, you can accomplish every part of the workflow described above. There are products besides HotDocs Developer that can accomplish the interview generation and final document assembly portions of the process, but only HotDocs Developer (or HotDocs Developer LE) enables you to create a HotDocs Template.

In addition to the basic template automation, HotDocs Developer's template features enable you to use variables to build business logic into your template. HotDocs Developer also enables you to design the look and flow of the interview you use to collect the information you need to generate a customized document for a client.

There are two main parts to the HotDocs Developer application:

- HotDocs Developer Library Window
- HotDocs tab

You use the HotDocs Automator aspect of Developer to create a graphical template.

HotDocs Developer Library Window

When you install HotDocs Developer, the HotDocs Developer library window installs as a desktop application. The library interface enables you to group and organize within your library those files required for assembling your final documents:

- Text templates—DOCX, DOC, RTF, and WPD-based templates
- Form templates—HPT files (HotDocs PDF-based Templates)
- Component files—Only visible in the library list if they are not associated with a template, otherwise you access these files from the Component Manager
- Clause library—Only visible in the library list if they are not associated with a template

The library window does not store these elements; rather it provides a list of shortcuts to the location where you store these elements on your computer. You can learn more about organizing these elements here.

Template Creation and Generation

In addition to acting as an organizer for your template development, you need the HotDocs Library dashboard features to perform the template generation functionality such as:

- Creating new templates
- Editing templates
- Editing component files
- Setting options
- Assembling documents

Answer File Manager

The Answer File Manager window enables you to organize your answer files in much the same way that the library organizes templates. (Click the Answer File Manager button on the HotDocs Developer library window to access the Answer File Manager.)

An answer file contains the answers an end user provides when completing an interview. Saving an answer file enables you to reuse an interview's answers to assemble other documents. For example, perhaps you have an answer file that contains only client-specific answers (such as names and addresses). You can use the answers in that file to assemble other related documents that need that same information.

HotDocs Tab

When you install HotDocs Developer, the install adds the HotDocs tab as an extension to your word processor's ribbon. When you select the tab on your word processor ribbon, you can see all the editing and navigation toolbar buttons you use to add variables to a template.

						H	lotDocs	
പ			«» Variable Field	🛃 INSERT Field	🖌 Edit Component	🖶 Clause Library	🏆 Apply Colors	≣ ₄ Go To Field
	bac over over	Test Assemble	📩 IF Field	🚍 ASK Field	晶 Component Manager	🕜 HotDocs Help	🖉 Label Fields	Previous
TOLDOCS	Close		🗟 REPEAT Field	«» Other Fields 🔹	🗓 Create Model		Match Fields	Next
HotDocs			Fields		Tools		Appearance//Navigation	

HotDocs Word Processor Tab

In addition, you have access to all the native functionality of your word processor. This integration with your word processor ensures that HotDocs generates final documents using whatever font faces, design, and pagination elements you choose.

HotDocs Automator

In addition to word processor functionality, HotDocs Developer (but not HotDocs Developer LE) includes HotDocs Automator. HotDocs Automator is a forms-based development environment that enables you to automate PDF-based graphical forms (fields, check boxes, etc.). While you can access Automator in your Start menu, you can also open it from within the library, just as you would open Word for a text template.

K HotDocs Automator	- 🗆 🗙
Eile Edit View Field Tools Window Help	
🗐 🎦 💕 🕞 🎒 🜌 A 🕪 🔚 🗠 🗠 🐝 🖌 🖧 🔯	- ▶ № ♡ ♀ 🕂 🕽 🗾 🛥 🗉 • • I 🗟 🔞

HotDocs Automator Toolbar

Since many document sets consist of both word-processor documents and graphical forms, HotDocs enables you to share component files among any number of templates, meaning you can generate all the word processor documents as well as PDF-based forms in a set from a single answer file.

Template Development Work Flow

Like most software tools, HotDocs Developer has a recommended workflow. To develop templates successfully, you should follow that workflow. Your first job as a HotDocs template developer is to understand HotDocs templates.

Because HotDocs Developer enables you to create templates of varying levels of complexity (ranging from very simple fill-in-the-blank templates to those that include sophisticated business logic and financial calculations), you should become familiar with Developer's capabilities and feature set by working your way through the HotDocs Developer tutorials that are part of the HotDocs Developer install. You should

also look through the list of advanced tutorials and decide whether to download and install those tutorials now, or wait until you gain more experience with creating basic templates on your own.

Once you have gained an understanding of the basics of HotDocs templates, you are ready to begin the template development workflow:

- 1. Plan your project; i.e., determine which document or set of documents you want to automate (transform into HotDocs templates).
- 2. Mark up (identify and mark) the changeable text in the set of document you want to use as templates.
- 3. Create a library to organize your templates.
- 4. Create a template within the appropriate library.
- 5. Prepare to automate your templates by thinking about how you want to approach the automation process.
- 6. Automate your templates by creating templates from the documents you marked up, then replacing the changeable text with variables, etc.
- 7. Organize the final template library by creating folders in the HotDocs library where you can group related templates.
- 8. Distribute the finished templates in an organized way to your template users.

Craft a Well-designed Interview

The goal of creating a HotDocs template is to generate an effective interview that gathers all the information you need your end-user or customer to provide. HotDocs can then create a final customized document by merging the data your customer supplies with your template's non-changing text. That document should contain everything you want it to, in exactly the order you need.

A well designed interview should go beyond merely asking questions; it should actually guide and assist your users in correctly inputting each answer to your questions. You should practice going through the interviews generated from your templates and have a colleague navigate your workflow. A fresh pair of eyes can prove invaluable in refining your interviews so each interview accomplishes exactly what you want and your final documents provide exactly what your clients or customers need.

List HotDocs File Name Extensions

The following is a list of all file types used by HotDocs:

File Name Type of File Extension

.DOCX	Word DOCX Template/ Word Document			
.RTF	Word RTF Template			
.DOT	Word Template			
.WPT	WordPerfect Template			
.TTX	Plain Text Template			
.HFT	HotDocs Envoy-based Form Template			
.HPT	HotDocs PDF-based Form Template			
.HFD	HotDocs Envoy-based Form Document			
.HPD	HotDocs PDF-based Form Document			
.DOC	Word Document			
.WPD	WordPerfect Document			
.CMP	HotDocs Component File			
.CMP	HotDocs Interview Template			
.HDL	HotDocs Template Library			
.HDL	HotDocs Clause Library			
.HCL	HotDocs Clause Archive			
.HDP	HotDocs Publish Settings File			
.HDA	HotDocs Auto-Assemble File			
.HDI	HotDocs Auto-Install File			
.HDK	HotDocs Registration File (required when publishing templates for commercial use with HotDocs Player)			
.HAL/.HDL	HotDocs Answer Library			
	HotDocs 6 through HotDocs 2008 supported binary answer libraries (or answer libraries with the .HAL file name extension). Starting with the release of HotDocs 2009, however, HotDocs now supports both binary answer libraries and XML-based answer libraries (which use the .HDL file name extension). By default, HotDocs will continue to use binary answer libraries as long as the file properties of items in the library use characters that are compatible with your system's default language. If you define answer file properties that contain foreign characters, HotDocs will create an XML-based answer library (so that it can properly display these characters in the Properties tab of the library window.)			
.ANS/.ANX	HotDocs Answer File			

	HotDocs 6 through HotDocs 2008 supported two types of answer files—binary answer files and XML answer files. These files were typically named using the .ANS and .ANX file name extensions, respectively. Starting with the release of HotDocs 2009, however, HotDocs can save answer files in only XML format. This means that if you open a binary answer file (such as an .ANS file) in HotDocs 2009, make a change to the file, and then save it, the contents of the answer file will be converted to and saved as XML. (The file name, however, will continue to use the ANS file name extension.) If you want the answer file to be compatible with versions of HotDocs prior to HotDocs 2009, see Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs.
.HPL	HotDocs Answer Source
.manifest.xml	HotDocs Template Manifest File (This file is generated when templates are published for use with HotDocs Server 11.)
.HDPMX	Publisher Answer Source Mapping File (This file is used to save variable mapping information created and used by other third-party applications.)
.HDUMX	User Answer Source Mapping File (This file is used to save variable mapping information created and used by other third-party applications.)

Using the Keyboard in HotDocs

Use the Keyboard to Work in HotDocs

You can use the keyboard to complete many tasks in HotDocs. The following describes some of these keyboard options.

The options described below refer to using a standard U.S. keyboard. Commands that require you to press two keys simultaneously are displayed using a plus (+) character. For example, to describe the **Print** command, the keyboard shortcut would appear as **Ctrl+P**.

There are four ways you can use the keyboard to work in HotDocs:

Tab Between the Elements in a Window, dialog box

Press the **Tab** key to move between the different elements of a HotDocs window or dialog box. Similarly, press **Shift+Tab** to move through the different elements in reverse order.

If certain elements don't appear in the tab order, use one of the other methods described in this topic, such as use an accelerator key.

For example, if you're viewing the template library and you want to assemble a document, you can press the **Tab** key to move focus to the item list of the library. Then press the **Up Arrow** or **Down Arrow** key to move between items in the library. Once the template you want to assemble is selected, press **Alt+M** (to view the **Template** menu). Finally, either press the **Down Arrow** key to select **Assemble** from the menu, or press the **A** key to activate the accelerator key in the **Assemble** command. The assembly window appears.

Use Accelerator Keys

Many commands and options in HotDocs can be accessed using an accelerator key. Accelerators appear as underlined letters in a command name or prompt. To access them, press the **Alt** key while also pressing the underlined letter. If you aren't in a text field (or a field where you can enter or select text), you can simply press the underlined letter and HotDocs will move you to that element or command.

For example, to save an answer file during an interview, at the assembly window, press **Alt+F**. The **File** menu appears. Then press **S**. The **Save Answer File** dialog box appears.

For some Windows 2000 and XP users, accelerator keys aren't identified until the Alt key is pressed. To always view accelerator keys, go to Start menu > Settings > Control Panel > Display. Click the Advanced tab, and then click Effects. Clear Hide underlined letters for keyboard navigation until I press the Alt key. To always view accelerator keys on Windows Vista, go to Control Panel > Ease of Access > Ease of Access Center > Make the keyboard easier to use and select Underline keyboard shortcuts and access keys.

You can assign your own accelerator keys in variable prompts, but you need to avoid letters already in use by HotDocs in the Interview tab of the Assembly window. Currently available keys are: B, C, D, I, J, K, M, O, Q, U, W, X, Y, Z. You can download this interview template to see how accelerator keys work in variable prompts.

Use the Shortcut Menu

When working with specific elements in a window or dialog, you can frequently access a list of commands for that element by displaying the shortcut menu. To access the shortcut menu, place focus on the element and then either press the **Windows Application** key on your Windows keyboard, or press **Shift+F10**. Use the arrow keys to move between options in the menu.

For example, to view where the answer to a specific question is used in the assembled document, while your cursor is in an answer field, press the **Windows Application** key on your keyboard. (Or, press **Shift+F10**.) A shortcut menu appears. Then, either press the **Down Arrow** key to select **Go to Answer in Document** (and then press **Enter**), or press the **G** key to issue the command.

To close a shortcut menu without choosing an option in it, press the **Esc** key.

Use Shortcut Keys

Several commands within HotDocs can be accessed using a shortcut key, which is a key or a combination of keys you can press that will quickly execute the command. (Accelerator keys can be considered shortcut keys.)

For example, to send an assembled document to the word processor, press the **F11** key. Or, to move between the different tabs of the assembly window, press **Ctrl+Tab**. (Pressing **Ctrl+Shift+Tab** moves through the tabs in reverse order.)

Click here for a complete list of shortcut keys.

Full List of Keyboard Shortcuts

You can use keyboard shortcuts to complete many tasks in HotDocs. A keyboard shortcut is a key or a combination of keys you can press that will quickly execute the command.

For example, to send an assembled document to the word processor, press the **F11** key. Or, to move between the different tabs of the assembly window, press **Ctrl+Tab**. (Pressing **Ctrl+Shift+Tab** moves through the tabs in reverse order.)

Click one of the links below to see the keyboard shortcuts for that specific part of HotDocs:

- Keyboard Shortcuts at Template Library
- Keyboard Shortcuts at Component Manager
- Keyboard Shortcuts at the Script Editor
- Keyboard Shortcuts for the Interview Tab of the Assembly Window
- Keyboard Shortcuts for the Document Preview Tab of the Assembly Window
- Keyboard Shortcuts for the Form Document Tab of the Assembly Window
- Keyboard Shortcuts for a Clause Library at the Assembly Window
- Keyboard Shortcuts for Answer File Manager
- General Keyboard Shortcuts
- Keyboard Shortcuts for JavaScript Interviews
- Keyboard Shortcuts for Silverlight Interviews

Template Development Workflow

HotDocs Templates Overview

A HotDocs template is a software component created using HotDocs Developer. The template contains all of the document's boilerplate text (the parts of the document which remain the same every time the document is generated), the variable fields (which will be replaced with new information collected during the interview), and any template logic needed (this can be repeated fields, conditional paragraphs,

inserted templates etc.). The HotDocs template and the paired HotDocs component file (generated by HotDocs alongside the template) contain all the information necessary to produce multiple documents with the information gathered in a HotDocs interview.

A HotDocs Template encapsulates three important things:

- **Natively formatted content.** For example, a HotDocs Template can be created right in Word; most of Word's formatting and document features can be built right into the HotDocs template. The documents you generate from such a template are native Word documents; no conversion or reformatting is necessary.
- **Embedded Document logic.** HotDocs embeds the business logic necessary to produce correct documents right inside your template. Computed values, conditional logic, specialized formatting, lists or tables of arbitrary length, nested lists, etc... all the know-how necessary to draft the perfect document.
- **Data-gathering Interview.** Not only does a template contain the logic HotDocs needs to produce a customized document, it also contains a definition of exactly what data is required to do so. HotDocs uses this to create a dynamic data-gathering interview based on the template. HotDocs Interviews can gather exactly the data that is necessary, leaving out the questions that are not relevant according to what's already been answered.

Template Type	Description
Text Templates	You create and automate a text template in a word processor, such as Microsoft Word or WordPerfect. You can modify the underlying text of a text template both as you automate the template and as you view the assembled document in the word processor. Text templates have a .DOCX, .RTF, .DOT, .WPT or .TTX file name extension. Microsoft Word .DOT templates cannot be used with HotDocs Server.
Form Templates	A form template is based on static graphical or text content, such as a PDF document. You place variable fields directly on top of the static content so that during the document assembly process, HotDocs can overlay answers on top of the static text. When you answers are merged with the static document content, the underlying static text and formatting does not change. Form templates have a .HFT or .HPT file name extension.
Interview Templates	An interview template gathers specific information (such as court, attorney, or client information). HotDocs saves the answers your end users provide for use in assembling documents; however, unlike text and form templates, HotDocs does not generate a document directly from an interview template. The only direct output of an interview template is an answer file. However, starting with an interview

Generally speaking there are three types or categories of templates:

template, HotDocs can trigger subsequent assembly of additional documents using the ASSEMBLE instruction. Interview templates have a .CMP file name extension.

When you create a new template file, HotDocs creates a companion file called the component file. The component file contains information about variables and other components used in the template. The component file has the same base file name as the template file, but with a .CMP file name extension. Both the template file and the component file are necessary for a template to work. Whenever you copy a HotDocs template—for example, to share a template with another user—you must be sure to copy both the template file and the component file.

The component file works in the background—as you create various components in the template, HotDocs stores the component file. Normally, each template uses its own component file, but you can make two or more templates share one component file.

Interview templates consist of only a component file; they do not have a separate template file.

Templates used with HotDocs Server also require an additional file: the template manifest (.manifest.xml). Like a component file, this file also shares the same base file name as the template except for the file name extension. It is created when you publish a template for use with HotDocs Server.

Versions of HotDocs Developer prior to 11 uploaded JavaScript (.JS) and Variable Collection (.HVC) files when publishing content for use with HotDocs Server. These files are no longer of concern when working with HotDocs Server version 11 or later.

Template Relationships

Any HotDocs template can potentially refer to (and thus rely on) other templates in two ways:

- **INSERT instructions.** When one template "inserts" another, this means content and logic from the inserted template dynamically become part of the parent template. The inserted template can influence the parent template's interview (changing which questions need to be asked, for example), and the content of the inserted template becomes part of the assembled document that is produced.
- ASSEMBLE instructions. When one template "assembles" another, this means the assembled template is "queued up" for subsequent interview and/or assembly after the current interview or assembly has been completed by the user. This is a separate operation, but it is necessary to implement the full feature set on which HotDocs template authors generally rely for proper execution of their templates.

Plan Your HotDocs Project

Gather and Review Documents

Before you can automate any templates, you must first identify which documents you want to include in a set of templates. You should gather all of these documents into one central location. Documents you should consider are those you routinely create for clients.

As you review your documents, you'll probably realize you've created the same type of document for multiple clients. While each of these documents contains generally the same content, some documents may contain unique or different paragraphs or clauses. To capitalize on the work you've already done, create a HotDocs Model where you copy all of these text variations. Then, when you automate the template, you can use HotDocs scripting to include or omit these different paragraphs, depending on the user.

Define the Audience

Another point to consider is your audience. Are you the only one who will use the automated templates, or will you share your work with others, perhaps in your firm or more broadly? If others will use the templates, how do you plan to distribute these templates to them? (Template distribution options are covered later in this document.)

As you identify your audience, you should also consider their experience in using the types of documents you are automating. If your users don't have the same understanding or experience you do with the documents, you may need to change the way the templates are automated to help them better understand the documents. For example, you may need to include additional resources to help users better understand how to answer questions in the interview.

Previous Topic Next Topic

Mark Up the Document Set

Mark Up the Document Set

The process of marking up a document for template automation includes identifying the text in the document that will change, depending on the person for whom the document is being created. Once identified, you indicate the variable that should replace that text. Variable information includes names, dates, numbers, and so forth.

How you mark up the document depends on your project and your needs. Following are some options:

- You can create a HotDocs Model using the HotDocs Markup Tools. Once the document is marked up, you can convert it to HotDocs template format. (See Introduction: Create HotDocs Models for details.)
- You can manually mark up word processor documents and store the markup information in a spreadsheet application, such as Excel. You can then take the marked up documents and create
templates from them, referring back to the spreadsheet for specific information about how the template should be automated.

• Another option is to simply print a copy of each document in your set and mark up the documents by hand. You can write directly on the printed document what variables should be used and identify which properties should be assigned to these variables.

Regardless of which markup option you choose, you should try to address these questions in the markup:

- How should variables be named? While it might not seem important at first, defining a naming scheme up front will save you time later when you need to manage a large list of components. For example, should you name related variables using a consistent format (like *Client Name*, *Client Address*, *Client City*, *Client State*, etc.)?
- 2. Which variables can be reused in other templates in the set? When templates require the same information, you should mark the variables using the same name. This way, when users save answer files after assembling a document, the answers can be used with those other documents. This saves the user from retyping common answers.
- 3. How should answers in the document be formatted? For example, should text answers be written using all capital letters? Should number answers be numeric (7) or spelled out (*seven*)?
- 4. How much information does the user need to know in order to answer certain questions during the interview? How should questions be phrased, punctuated, and capitalized? Should they contain examples of possible answers?
- 5. How much additional information should be included with variables? For example, when should variable resource (or help) information be included? How much is too much?
- 6. What other information should be known about the variables? For example, are there restrictions or limits for answers? Should an answer be formatted a certain way? If the answer must be calculated, what kind of calculation should it be? (For a complete list of properties you can assign to the different variables, see Appendix 1: Variable Properties.)
- 7. What text in the document should be included only under certain conditions? For example, if there's a paragraph about the client's spouse, but the client isn't married, you wouldn't want to include the paragraph.

You can also mark in the document where lists of answers should be merged and where boilerplate text should be inserted. See Appendix 2: Template Instructions for details.

◀ Previous Topic Next Topic ▶

Prepare to Automate the Templates

Prepare to Automate the Templates

Once the templates in your set are marked up, you should think about the following things before you begin the actual automation process:

Decide if you want to share component files. If there are common variables across most of the templates, you may consider pointing each template in your set to a shared component file. Then, when you make a change to a shared component, (for example, if you change a prompt or resource text, the change will appear in all templates that use that component.) (See Use One Component File for Multiple Templates.)

It's a good idea to point all the templates that will share a component file to the shared component file before you start creating components in the templates. That way, they are saved directly in the shared component file and are ready for use when you automate other templates.

Identify boilerplate text. If there is boilerplate text in the templates, consider saving that text in a separate template file and then dynamically inserting that template in each template that requires it. (Examples of boilerplate text include captions, signature blocks, and so forth.) Then, when you need to make a change to the text, you only need to make it in one file.

Create common document styles. If automating a Microsoft Word template, make good use of styles. Ultimately, this will help retain formatting throughout your templates. The Microsoft Word Help file contains a lot of useful information on using styles.

Set HotDocs Options. As you prepare to automate your templates, there may be some custom settings you want applied to all new templates you create. For example, if your templates will be used with HotDocs Server, you can specify a setting that enables all new templates you create for use with HotDocs Server.

To view a list of settings you can change, at the HotDocs template library, choose **Options** (**Tools** menu) and then open the **Template Development** folder. (See HotDocs Options)

Previous Topic Next Topic

Automate Your Templates

Automate Your Templates

The next step in the process is to actually start automating your templates.

You can find detailed instructions on each phase of this process in the HotDocs Help.

Create the Template

You first begin by creating templates from the documents you marked up.

When you create a new template, you can enter a title and a description for the template. Include as much information as necessary to better identify the file. For example, if the file that's created and saved to disk uses some cryptic file naming scheme (such as *RLWHCP.docx*), you could enter a different title that makes

more sense, such as *Revocation of Living Will and Health Care Proxy*. In the description you can enter as little or as much information about the template as you feel necessary.

Also, if you plan to point your templates to a shared component file, you should do so right after you create the template and before you begin any automation work. That way HotDocs will correctly save any variables and other components you create to the shared component file.

We recommend that template names are kept to less than 52 characters and don't use non-ASCII characters. This will reduce the possibility of the template name causing errors if you later enable it for use with HotDocs Server.

Replace Template Text with Variables

Once the template has been created, you replace changeable text in the template with variable fields. If you've previously marked up the template, you should be using variable names and properties you defined in the markup.

As you create variables, you should test them (by clicking the **Test** button on the component editor). This shows you how the question will appear to the user. Frequently, testing helps you identify ways to improve the variable question. For example, if you're testing a Text variable that requires the user to enter a long description, you may realize you forgot to adjust the answer field height to allow for multiple lines of answer text.

Make Text Conditional

Once you have replaced all of the variable information with variable fields, you should then review the document to identify sections of text that are optional. To mark these sections, you use IF instructions. IF instructions use a True/False variable (for example, *Is the Client Married?*) to determine whether to include text or not. More complicated instructions usually require more than a simple yes/no question to determine. An example of an expression would be IF YEARS FROM(Child's Birth Date, TODAY) <= 17.

There are two different ways you can use conditional text:

First, you can use IF and END IF instructions to include or exclude simple optional text. For example, in an employment agreement, you may need to include or exclude a paragraph describing a probationary period for the new employee. If the employee isn't required to complete a probationary period, you wouldn't include information about it.

Second, you can use IF, ELSE IF, ELSE, and END IF instructions to include alternative text. For example, you may have different versions of a paragraph, only one of which the user will include in the assembled document. Using ELSE IF and ELSE, you can create a series of conditions under which the user determines which text HotDocs will include.

Repeat Variables to Create Lists

Sometimes there are variables in a template that you need to repeat so users can enter more than one answer. For example, you may need to list the client's children. Because you don't know how many children a client may have, you can repeat the *Child Name* variable. This allows the user to enter as many children as necessary. You accomplish this by using a REPEAT instruction. You can repeat a single variable, or you can repeat entire sections of text.

Often, you must repeat information within a word processor table. For instructions on how to repeat a single cell or an entire row, see Use a Word Processor Table to Display a List.

Group Variables in Dialogs

As you know, variables in the template represent the questions users must answer during the interview. You can organize these variables into logical groupings, called dialogs.

Typically, when you create a dialog, you group similar questions together. For example, if the template asks for the name and address of a client, you could group all of these variables together in a single dialog named *Client Information*. Information about the spouse could be asked in a separate dialog, as could information about the case.

If HotDocs should ask questions only under certain conditions, you can use dialog scripting to include or exclude those questions. You can also select several different options that control the way the questions appear in the dialog. For example, you can choose to place all of the questions above the answer fields, or you can choose to place each question to the left of its answer field.

Test Assemble the Template

Test assembling a template is an imperative task in the template development process. Testing allows you to see the full interview as the user will see it. It provides an opportunity for you to correct mistakes, as well as fine-tune the interview process by improving questions and adding other information to make the process easier for the user.

It's important that as you are testing, you enter many different combinations of answers. This exercises the scripting in your template and helps you ensure that no matter what answers users enter, HotDocs will assemble the document correctly.

As you test assemble a template, you should check for the following things:

- HotDocs isn't reporting any errors or warnings that you need to fix.
- The interview outline looks right and operates properly.
- Dialogs look right and operate properly.
- HotDocs has assembled the document correctly with no unanswered questions and has correctly formatted the answers.

If you find problems as you are testing, you can leave the test assembly window open and make changes to the underlying components. You can then update the test assembly window with your changes.

◀ Previous Topic Next Topic ▶

Organize the Final Template Library

Once all of your templates are automated, you can organize them within the library. How you organize the library depends on the template set. (See Introduction: Use HotDocs Libraries.)

One way you can organize the library is to create folders for storing the templates. Frequently, you can group similar templates in these folders. For example, wills can be organized in one group or a series of subgroups, depending on the number and types of wills you have automated. (See Organize the Contents of a Library.)

If you have a large set of templates, it may make more sense to create individual libraries for the template types. Real estate templates can be stored in one library, while family law templates could be saved in a different library.

Because the items in the library work like shortcuts to the actual files, you must be cautious when working with the actual files. Moving an actual file without updating the reference in the library will make the file inaccessible from the library. If you need to move the files in the library, use either the **Move** command or the **Copy** command (both commands found in the **Template** menu). If you need to update references, choose **Properties** or **Multiple** (**Edit** menu). (See Move Items to New Locations on Disk, Change the Properties of a Single Library Item, and Change the Properties for Multiple Items in a Library.)

When you created the templates, you may have added template titles and descriptions for each template you added to the library. If you did not, this is a good time to do this. Template titles and descriptions are an easy way to help the user identify the template. You do this by editing the template properties. (See Change the Properties of a Single Library Item.)

Previous Topic Next Topic

Distribute the Templates

Distribute the Templates

The final task in the template development process is to distribute your template sets so they can be used for assembly. Before doing this, you must first identify who will use your templates.

If you've automated your templates for your own practice, there may not be a distribution plan—you can assemble documents from the same templates you are editing. Or, if you'd like to separate template development from document assembly, you can create a new library and copy your completed templates

to a different folder. As you make corrections to the templates, you can update those files saved in the assembly-only location.

If you plan to share your templates with others in the firm, you should place copies of the files on a network location for your co-workers. You should restrict access to the templates as much as possible to keep users from inadvertently changing the files. Some ways you can do this include:

- Making the network folder where the templates are saved read-only. (You should also make the template files themselves read-only.)
- Placing a copy of the files on a network and then having users import the library to their local drives for assembly-only. (See Copy Templates to New Locations and Import One Library Into Another.)
- Publishing the template set as a HotDocs auto-install file, which users can then install on their computers. (See Publish Templates as Auto-Install Files (HDI).)

If you plan to share your templates with those outside the firm, you should publish the templates and distribute them using the Internet or even a CD.

If you plan to sell your automated templates and distribute copies of HotDocs Player with the templates, you must acquire a publishing license from HotDocs Corporation, which allows you to redistribute HotDocs Player to your customers. (HotDocs Player is a free, assembly-only version of HotDocs.) (See Publish and Register Templates for Use With HotDocs Player.)

Previous Topic Next Topic

Conclusion

Conclusion

As with any major project, the more planning you do up front, the better your results will be. While this document covers the basics, it's expected that you will also discover tips and tricks that will help you improve the template automation process.

For helpful lists of Variable Properties and Template Instructions see the appendices to this section:

- Appendix 1: Variable Properties
- Appendix 2: Template Instructions

Previous Topic

Appendix 1: Variable Properties

Appendix 1: Variable Properties

Following is a list of the different variable types, followed by the properties you can assign to each variable type:

Text Variables	
Patterns	HotDocs allows you to assign a pattern to a text variable. The two most common patterns are telephone number patterns and Social Security number patterns. These patterns format users' answers correctly both in the interview and in the assembled document.
Answer field height	Each variable in a template is asked in an answer-gathering dialog during the interview. You can control how tall (up to 12 lines) the answer field is in the dialog.
Maximum	The maximum number of characters allowed in a text answer is 15,000. If an answer requires a specific number less than this, you can specify it.
Number Variables	
Maximum and minimum values	You can specify limits for how large or small a number can be. For example, if new employees are allowed between 10 and 15 paid vacation days, you can specify these limits so that users don't inadvertently specify more vacation days than they are allowed.
Decimal places	You can indicate the number of decimal places (up to 7) that should be allowed in a number.
Currency symbol	You can designate whether a currency symbol should be used for a monetary amount. Predefined options include \$, £, DM, and \in , but you can enter your own. (Patterns have a 3-character limit).
Multiple Choice Variables	
Multiple Choice options, prompts, and merge text	Multiple Choice variables allow users to choose an answer from a list of predefined options. If the options you specify aren't descriptive enough, you can define a prompt for each option. Additionally, you can specify different text for each option that will be merged in the completed document.
	For example, say you need to merge gender-specific pronouns (like he/she or his/hers) in the document. To do this, you would create a Multiple Choice variable (Gender) with two options— Male and Female—and then specify merge text that will merge the correct pronoun (he or she), based on the user's gender.

Appendix 2: Template Instructions

Selection options and styles	You can allow users to select only one option or select multiple options. You can also designate whether the options appear as buttons or as lists. By default, all new Multiple Choice variables are set to Select One Only .
Computation Variables	
Scripts	Computation variables tell HotDocs to perform a particular task during assembly, such as insert one template into another or perform some action based on an answer the user provides. Similarly, they are used to calculate monetary amounts or other figures, or find the number of years between two given dates, and so forth.
Previous Topic	

Appendix 2: Template Instructions

HotDocs Instruction	What It Does
IF instruction	Designates that a variable, word, phrase, or paragraph should only be included in the assembled document when a certain condition is met.
REPEAT instruction	Repeats variables and text so that users can enter more than just one answer.
INSERT instruction	Inserts another template into the current template. (For example, perhaps you have a separate file that includes boilerplate language. You can use a separate document for this language and then just insert it wherever you need it.) In most cases, inserted templates appear "inline" with the template into which they are inserted.

Previous Topic

Using Libraries

At a Glance: The HotDocs Library Window



Introduction: Use HotDocs Libraries

HotDocs files—such as templates, clause libraries, and answer files—reside in folders on your local disk or on a network. To make it easier to organize templates and clause libraries, you can use a HotDocs template library or, for answer files, the Answer File Manager. Using a library lets you assign titles and descriptions to files, and, in the case of answer files, track the history of when a file is used. Template libraries help you organize template files and provide quick access to many HotDocs tools, such as the Assembly Queue and HotDocs Options:



The Answer File Manager is a library of answer files. You can group answer files as well as sort them. You can also view a history of when a specific answer file was used:



Both the template library and Answer File Manager are divided into two panes. The left pane shows a list of the items referenced in the library. Depending on what kind of library it is, these items can include folders, templates, clause libraries, or answer files. The right pane displays information about the selected item. A different view shows the contents of the file.

The items you see in the left pane are not the actual HotDocs files on your local disk or network. Instead, they are references to the files, operating similar to Windows-type shortcuts. When you select an item for assembly, HotDocs follows the assigned file path and locates the actual template or answer file.

Additionally, folders in a library help you organize the list of files in the library—not the actual folders on the local disk or network.

Because the items in the library work like shortcuts to the actual files, you must be cautious when working with the actual files. Moving an actual file without updating the reference in the library will make the file inaccessible from the library. If you need to move the files in the library, use either the **Move** command or the **Copy** command (see Move Items Within a Library and Copy Templates to New Locations). If you need to update references, choose **Properties** or **Multiple** from the **Edit** menu. (See Change the Properties of a Single Library Item or Change the Properties for Multiple Items in a Library.)

The library window is the starting point for many HotDocs tasks. Such tasks include organizing templates, moving templates and component files, editing templates, assembling documents, and accessing other HotDocs tools, such as Answer File Manager, HotDocs Options, and Component Manager.

At a Glance: The New Library dialog box

New Library		? 💌
Eile name:		
	A	
<u>T</u> itle:		
Descriptions	0	
Description:		
	G	
		OK Cancel

After opening the HotDocs Library, you can open the **New Library** dialog box by clicking on the **library** button on the toolbar.

In the first text field \underline{A} you can enter a file name for the new template and you can use the $\underline{\Box}$ **Browse** button to the right specify a location for the new library file in your file system.

HotDocs will automatically copy the file name into the second text field **B** as the template title, but you can edit this if you would like to use a different title for the template. This title will appear in the title bar of the library window.

In the third text field \bigcirc you can enter an optional description of the library. Here you can specify information about the library, such as what kinds of templates are included, or when the library should be used. If specified, it appears when the user views the properties of the top folder in the library.

To learn more about creating a new library follow the link below:

• Create a Library

Creating a Library

You may want to use separate libraries for templates that deal with different matters. For example, you may want to create one library for divorce templates and another library for estate planning templates. When you create a library, you create an .HDL file on the disk. Once it's created, you can add references to templates to it as well as organize it in whatever way you need.

By default, all new libraries you create will be saved in XML format. If you need to use the library with versions of HotDocs prior to HotDocs 2009, you must specify this format by choosing **Save Library As** from the **File** menu and then choosing **HotDocs 2005-2008 Library** from the **Save as type** drop-down list. (See HotDocs and XML File Formats for details.)

To create a library

- 1. At the HotDocs library window, click the **New Library** button. The **New Library** dialog box appears.
- 2. Click the Rowse button next to the File Name field. The New Library File Name dialog box appears.
- 3. Browse to the location where you want to save the file and enter a name in the **File Name** field. (Click **OK** once you have specified the file location.)
- 4. At the **New Library** dialog box, enter a name for the library in the **Title** field. This title appears as the top folder in the library window. (Library titles can be up to 100 characters long.)
- 5. Optionally, type a description of the library in the **Description** field. The description appears in the **Properties** tab when the user views the main folder in the library.

Once you create a new library, you can add templates and other files to it. See Add Templates and Other Files to a Library.

See Change HotDocs Program File Locations for information on the default location HotDocs looks for and saves library files.

Open a Library

Often you have templates that are located in different libraries. You can access any library by using the **Open** command at the library you are currently viewing.

To open a different HotDocs library

- 1. Start HotDocs. (See Start HotDocs.)
- 2. At the library window, click the **Open Library** button. The **Open Library** dialog box appears.
- 3. Locate and select the library (.HDL) file with which you want to work, then click **Open**. The folders, templates, and clause libraries in the new library appear in the template list.

After opening the library, you can edit the templates, or use them to assemble documents.

See Change HotDocs Program File Locations for information on the default location HotDocs looks for and saves library files.

At a Glance: The Add Item dialog box

Add Item		? <mark>×</mark>
<u>T</u> ype: Text Template	۵	•
<u>File name:</u>		
	B	
Tite:	a	Ŧ
Description:		
	O	
	<u> </u>	Cancel

After opening the HotDocs Library, you can open the **Add Item** dialog box by clicking on the **add Item** button on the toolbar.

At the top of the dialog box is a drop-down list <mark>A</mark> where you can choose the type of item you wish to add to the library. You can choose from:

- Text Template
- Form Template
- Interview Template
- Clause Library
- HotDocs Model
- Auto-Assemble File

- Web Address
- Folder

In the first text field \mathbf{B} you can enter a file name for the new template or you can use the **Browse** button to the right to locate a file in your file system.

In the second text field \bigcirc you can specify the title of the item, which appears when you select the item and view its properties. HotDocs suggests a title based on the file name. If the template provider has specified a default title, you can click the T Get Title button and that title will be used instead.

In the third field **D** you can enter an optional description of the library item, such as how the item should be used, or when it was created. This information appears when you select the item and view its properties.

To learn more about adding an item to a library follow the link below:

• Add Templates and Other Files to a Library

Adding Templates and Other Files to a Library

After you create a library, you can add folders, template files, Web addresses, and miscellaneous files to the folder. In order to upload a template (or any other file) to a host application, you must first add it to a library.

When you create new templates, they are automatically added to the library. Clause libraries created at the template library are likewise added to the library automatically. Adding an item to a library does not affect the files on your local disk or network. For example, you could select a form template from a library on your network and add it to a library on your local disk. However, the actual file would remain on the network.

Adding a Single Item to a Library

The items you add using this method are those available in the Type drop-down list.

To add a single item to the library

- 1. At the HotDocs library window, select a folder.
- 2. Click the **Add** button. The **Add Item** dialog box appears.

- 3. From the **Type** drop-down list, select the appropriate kind of library item. (Your options include **Text Template, Form Template, Interview Template, HotDocs Model, Clause Library, Auto-Assemble File, Web Address**, and **Folder**.)
- 4. Click the **Browse** button next to the **File name** field and locate the library item.
- 5. Select the item you want to add and click **OK**. The **Add Item** dialog box appears again.
- 6. In the **Title** field, enter a title for the library item (100 characters or less). The title will identify the item in the library.
- 7. Optionally, enter a description in the **Description** field. The description appears in the **Properties** tab when the library item is selected.

Adding Multiple Items at Once

You can add multiple files to a library simultaneously. Doing this however, means you must wait to assign properties (such as titles and descriptions) to each individual file until after you add the files to the library.

To add multiple items at once

- 1. Follow the steps in the procedure above until you choose the item typel
- 2. Click the **Browse** button next to the **File name** field and locate the library item.
- 3. Select all the items you want to add and click **OK**. Once they are added, you can modify the properties.

Adding Miscellaneous Files

You can also add miscellaneous files such as word processor documents, PDF files, and so forth to the library. Once you add the file, you can assign properties (such as titles and descriptions) to each individual file.

To add miscellaneous files

• In Windows Explorer, browse to the file or files you want to add to your library, drag and drop the files onto the library at the location where you want them. Once they are added, you can modify the properties.

Work with Templates and Other Files in a Library

You use a HotDocs library to organize and maintain your template set, including adding and removing templates from the library and changing the properties of templates in the library. (See Organize the

Contents of a Library.) You can also use commands in the library to work directly with the templates. For example, you can select a template and assemble it.

To use the library to access the templates and clause libraries

- 1. At the HotDocs library window, open the desired HotDocs library. (See Open a Library.)
- 2. Perform tasks as explained in the following table:

То	Do This
Assemble the selected document	Select the template and click the Assemble button.
Create a new template or clause library and add its reference to the library.	Click the New Template button. (See Create a New Text Template File or Create a Form Template.)
Create a new folder in the library	Choose Add Folder (Edit menu).
Edit an existing template or clause library	Select the item and click Edit . (See Edit a Template or Customize a Clause Library.)
Add an existing item to the library	Click the Add Item button. (See Add Templates and Other Files to a Library.)
Remove the selected item from a library	Click the Remove Item button. (See Remove Items from a Library.)
Move items up or down in the library list	Click on the item and drag it up or down in the list. HotDocs displays a horizontal bar indicating where the item will be placed when you release the mouse.
Modify the selected item's file name, title, or description	Click the Properties button. (See Change the Properties of a Single Library Item.)
View or hide all of the items within a folder in the library list	Choose Expand All (View menu). (To hide the list again, choose Collapse All .)

At a Glance: The Remove Item Dialog Box

Remove	Item 🛛 🕄 🔀
<u> </u>	You have selected 1 item to be removed from the library. In addition to removing the selected item, you can Remove the 0 unselected items contained in selected folders
	Are you sure you want to remove the selected item from the library?

After opening the HotDocs Library, you can open the **Remove Item** dialog box by clicking on the **Memove Item** button on the toolbar.

You can use the two check boxes to specify any options for the removal. Check the top box to approve the removal of items stored within the folder you have selected and check the bottom box to delete the selected files from the disk as well as deleting the references from the library.

To learn more about removing an item to a library follow the link below:

• Remove Items from a Library

Remove Items from a Library

Libraries do not contain the actual files that HotDocs uses, such as template files, component files, or answer files. Libraries only contain *references* to those files, similar to shortcut icons on a Windows desktop. When you remove an item from a library, you can choose to remove just the reference, or you can choose to delete the actual file from disk.

The available options for removing items in a library depend on the type of item:

- When you remove the reference for a template, you can choose to also delete the associated template file, component file, and clause library file (if any) from disk. If you do this, the files you delete are either sent to the Windows Recycle Bin (if the template is stored on your local disk) or permanently deleted (if it's stored on a network drive). Or, you can delete only the reference, leaving the template, component file, and clause library file in place on the disk—perhaps for use in another library.
- When you remove the reference for a folder, you can choose to delete the items within the folder. If you choose to remove the items in the folder, all of the items in that folder and in all subfolders will be removed. If you choose to leave the files, they will be moved to the next higher folder.

To remove items from a library

- 1. At the HotDocs library window, select the item (or items).
- 2. Click the **Remove Item** button. The **Remove Item** dialog box appears, asking if you want to remove the reference to the file from the library.
- 3. Optionally, if you have selected a folder and you want to remove items within the folder, select **Remove the unselected items contained in selected folders**.
- 4. Optionally, if you want to delete the actual files from disk, select **Permanently delete the files** for selected items from the disk.
- 5. Click Yes.

If the template you are deleting is in the same folder as another template with the same file name but a different file name extension (for example, you are deleting *Invoice.rtf* and the same folder also contains *Invoice.wpt*), HotDocs doesn't delete the associated files, like the component file. You need these files for the other template.

At a Glance: The Item Properties dialog box

Item Properties		? <mark>×</mark>
Eile name:		
<u>T</u> ite:	0	Ŧ
Description:		
	G	
		Cancel

After opening the HotDocs Library, you can open the **Item Properties** dialog box by clicking on the **Properties** button on the toolbar.

In the **File name** field vou can enter a file name for the new template. You can use the **Browse** button to view or change the location on the disk where your template is saved. This field can also be used to enter command-line options for the template. For example, by adding different command-line options to the end of the file name, you can have HotDocs automatically save answer files, send the finished document directly to the word processor and many more.

HotDocs will automatically copy the file name into the **Title** field **B** when you click into it, but you can edit this if you would like to use a different title for the template.

In the **Description** field **G** you can enter an optional description of the library item, such as when it was created, or how it should be used. This description will appear when the user views the item properties tab in the library.

Clicking on the $\mathbf{\mathcal{F}}$ Get title button pulls the title and description information from the Component File Properties and will overwrite anything currently in both fields.

To learn more about item properties follow the link below:

- Change the Properties of a Single Library Item
- Change the Properties for Multiple Items in a Library
- Introduction: Command-Line Options

Change the Properties of a Single Library Item

You can change the title and description of a library item. If the item is a template or clause library, you can also change the file name and add command-line options.

Changes made at the **Item Properties** dialog box affect only the reference to the item—not the template or clause library files themselves. For example, if you change a template's file name at the **Item Properties** dialog box, it does not change the template's name on your local disk or network. Unless you also update the underlying file, the template may no longer work because the reference is incorrect. (See Rename Templates Using Template Manager for information about renaming templates and having all references to the file updated—including in the library.)

To modify a library item

- 1. At the HotDocs library window, select the item you want to modify.
- 2. Click the **Properties** button. The **Item Properties** dialog box appears.
- 3. Make changes based on the following options:

То	Do This
Change the file path	In the File Name field, type the new file path, or click the Revues button and specify a new path.

Change the item title	In the Title field, either type a new title in the field, or click the $\mathbf{\mathcal{T}Get Title}$ button to use the default title. (Titles can be 100 characters long.)
Change the command-line options	In the File Name field, type the new options. (For example, <i>Refund.wpt /af=bwhittington.anx</i>). See Introduction: Command-Line Options.
Change the item description	In the Description field, enter a new description. (The description appears in the Properties tab when the library item is selected.)

You can change the properties of several other library items at once. See Change the Properties for Multiple Items in a Library.

At a Glance: The Multiple Item Properties dialog box

Multiple Item Properties	? 💌
Elle path:	- 🞑 🗖
File name <u>e</u> xtension:	
G	-
	OK Cancel

After opening the HotDocs Library, you can open the **Item Properties** dialog box by clicking on the **Multiple** button in the **Edit** Menu with the items selected.

In the first text field \triangle you can choose to leave the <default> file path, enter a new file path for the selected items, choose from recently used paths in the drop-down list, click the \triangle **Browse** button to navigate to and select a new folder, or click the \triangle **Reference Path** button to select the path you predefined in HotDocs Options (See Change HotDocs Program File Locations).

Using the second text field , you can keep the <various> file names,or choose to give all the selected items the same file name extension (by typing it into the field or alternatively, by selecting one of the items existing extensions using the drop-down list). Note, however, that changing the file name extension here does not change the extension for the actual file.

In the third text field vou can type in any command-line options you would like to add to the items or choose from a list of currently used options in the drop-down list. Any options you specify here will be applied to all the items you have selected. (To learn more about command-line options see Introduction: Command-Line Options and to view a list of command-line options see Full List of Command-Line Options)

To learn more about item properties follow the link below:

- Change the Properties of a Single Library Item
- Change the Properties for Multiple Items in a Library

Change the Properties for Multiple Items in a Library

You can change the folder paths, file name extensions, and command-line options for two or more library items in a folder. For example, if you have used Windows Explorer to move templates from one folder to another, you can update all the file paths in the library at once. (Remember, moving the files on disk does not automatically update the references to those files in the library.)

If you select a folder, the changes affect all items in the folder and in any subfolders. For example, if you select the top folder of a library, you will change the properties for every item in the library.

Changing properties of library items does not affect the files on your local disk or network. For example, changing the file path in the **Multiple Item Properties** dialog box does not actually move the file to a different folder—you must manually move the files. (See Move Items to New Locations on Disk.)

To change the properties of two or more items in a library folder

- 1. Select the items (or the folder that contains the items) and select **Multiple** (**Edit** menu). The **Multiple Item Properties** dialog box appears.
- 2. Make changes as specified in the following table:

То	Do This
Change the file path for the selected items	Click the File path Browse button and browse to the new folder. Then click OK . The new folder path is inserted. (Click the File path drop-down button to view a list of all the various file paths used for the templates in your list.)
	To specify the default folder the different files, choose <default></default> from the File path drop-down list and click OK . (The default folder is the location

	HotDocs looks for the file when no file path is given. You can specify what this default location is at HotDocs Options. See Change HotDocs Program File Locations.)
Change the reference path for the selected items	Click the Reference path button and choose a path from the Select Reference Path dialog box. When you click OK , HotDocs inserts the reference path keyword in the File path field.
	To remove all reference paths and specify the default folder for the different files, choose <default></default> from the File path drop-down list and click OK . (The default folder is the location HotDocs looks for the file when no file path is given. You can specify what this default location is at HotDocs Options. See Change HotDocs Program File Locations.)
	The Reference path button isn't available unless you have at least one reference path defined in the File Locations folder of HotDocs Options. (See Assign Reference Paths to HotDocs Files.)
Change the file name extension for the selected	Click the File name extension drop-down button and make a selection. If the desired extension is not listed, type it.
items	HotDocs displays a list of the file name extensions for all the items you have selected in the File path drop-down list. If you have multiple file types selected, <various></various> appears in the File name extension field.
Change the command-line option or options for the selected items	Type the command-line option (for example, <i>/af=path and file name</i>) in the Command-line options field and click OK . If assigning multiple options, include spaces between each option. (See Introduction: Command-Line Options.)
	To remove existing command-line options, select <none></none> from the Command-line options drop-down list and click OK.

- 3. Click **OK** at the **Multiple Item Properties** dialog box.
- 4. When HotDocs confirms the change, click **Yes**.

At a Glance: The Folder Sort Options Dialog Box

Folder Sort Options		? <mark>×</mark>
Sort Order		
Sort Subfolders		
_	ОК	Cancel

After opening the HotDocs Library and clicking on your chosen folder (or the entire library), you can open the **Folder Sort Options** dialog box by clicking on the 2+**Sort** button on the toolbar.

Using the multiple choice menu \boxed{A} you can choose between **Ascending** or **Descending**. **Ascending** sorts the contents of a folder from 1 to 9, and from A to Z and **Descending** sorts the contents of a folder from 9 to 1, and from Z to A.

Clicking on the check box B below will sort items in any subfolders based on the Sort Order option you selected.

To learn more about sorting library folders follow the link below:

• Sort Items in a Library

Sort Items in a Library

To help you locate or organize files in the library, you can sort the items in a folder in alphanumeric order.

To sort library items

- 1. At the HotDocs library window, select the folder whose items you want to sort.
- 2. Click the $2 \downarrow$ Sort button. The Folder Sort Options dialog box appears.
- 3. Select the order for sorting the library items:
 - To organize them A to Z and 1 to 9, select Ascending, then click OK.
 - To organize them *Z* to *A* and *9* to *1*, select **Descending**, then click **OK**.
- 4. Optionally, select Sort Subfolders to sort the folders inside the selected folder.

The template list changes to show the rearranged library items.

You can save templates you frequently use to a *Favorites* folder in the library for quicker access, select the template and click **Add to Favorites** (**View** menu). See **Create a Favorites Folder in the Template Library**.

Organize the Contents of a Library

Once you create a library and add templates to it, you often want to organize the contents of the library into logical groups. This may include grouping items by subject matter in folders, or grouping items alphabetically. Additionally, you may need to print a list of templates in the library or remove templates you no longer need.

To organize the contents of a HotDocs library

- 1. At the HotDocs library window, open the desired HotDocs library. (See Open a Library.)
- 2. Use any of the following commands to work with files in the library:

То	Do This
Add a folder to the library so you can group library items	Choose Add Folder (Edit menu). The Add Item dialog box appears where you can specify the folder name and a description. (Once you create a folder, you can move items to it.)
Print a list of the titles, file names, and descriptions of templates and folders in a library	Click the Print Library button. (See Print a List of Items in a Library.)
Remove an item from the library	Select the item and click Remove Item button. (See Remove Items from a Library.)
Remove an item from the library and places a copy of it on the Clipboard so you can paste it elsewhere in the library	Select the item and click the $ auCut button.$
Copy an item in the library to the Clipboard so you can paste it elsewhere in the library	Select the item and click the aCopy button.
Paste an item to a new location in the library	Select the folder in the library where you want the item pasted and click the $\square Paste$ button.
Sort items in a library in ascending or descending order based on their titles	Select the folder on which you want to sort, and click the 2 Sort button. (See Sort Items in a Library.)

Using Libraries

Change an item's file name, title, or description	Select the item and click the Properties button.
View the file names for templates in the library (rather than the template titles)	Choose File Names (View menu). (To view template titles again, choose Template Titles .)
Search for a specific template in the library list	Type the text for which you are searching in the Find field. HotDocs will show only those files that contain the search text in the template title, file name, or description. (See Search for a Specific Template in a Library.)
Group templates you use most frequently in a <i>Favorites</i> folder	Select the template and choose Add to Favorites (Edit menu). (See Create a Favorites Folder in the Template Library.)

Create a Favorites folder in the Template Library

You can create a *Favorites* folder in your template library and add templates to it. This may be useful if you find you assemble some documents more than others and you want to access them quickly each time you view the library.

To add templates to a Favorites folder in the library

- 1. At the template library, select the template you want to add to the Favorites folder.
- 2. Click Add to Favorites (Edit menu).

If no Favorites folder exists, HotDocs creates the folder and adds the template to it.

Create a New Library by Exporting Part of an Existing Library

You can create smaller libraries by exporting some of the folders and files of a larger, existing library. Creating a library this way eliminates the need to manually create a new library and then add each folder, template, or clause library to it.

When you export part of a library, HotDocs asks you for the file name and title of the new library. Then it creates the library, and finally creates the library references in it. During the process, you can choose to

have HotDocs copy the files in the library you are exporting to the new location, or you can choose to have the library reference the original files.

Exporting will only export the files you select at the library. This means that if a template you are exporting contains references to other templates or files not listed in the library, those items will not be exported and you may receive errors when you attempt to use the template for assembly.

To export part of an existing library

- 1. At the HotDocs library window, select the library items you want to export. (If you select a folder, all the items in the folder will be exported.)
- Choose File > Export Library To > HotDocs Library File. The Export HotDocs Library File dialog box appears.
- 3. At the **File name** field, click **Growse**, select a location for the new library, and enter a library name in the **File name** field. (Click **OK** when you are finished.)
- 4. At the **Export HotDocs Library File** dialog box, enter a name for the new library in the **Title** field, or accept the suggestion HotDocs makes. This title will appear in the **Properties** tab of the HotDocs library window.
- 5. Optionally, in the **Description** field, type a description of the library. (The description appears in the **Properties** tab when a library (or folder) is selected.)
- 6. Click **OK**. HotDocs asks if you want to export the selected files:
 - Click **Yes** to copy the selected files to the same folder as the new library.
 - Click **No** to have the new library refer to the files in their current location.

After exporting the items, the original library items are still available. (HotDocs does not delete the references from the original library—templates can be referenced in multiple libraries.) If you want the old references deleted, select the library items and click **Remove Item**.

HotDocs does not automatically display the new library. To open it, click **Open** and select the file name for the new library. (See Open a Library.)

At a Glance: The Import Library Dialog Box



After opening the HotDocs Library, you can open the **Import Library** dialog box by choosing **Import Library** from the **File** menu and selecting a HotDocs library at the browse window.

You have three options at the Import Library dialog box:

- **Assembly Only:** Imports the library in such a way that you will be able to assemble documents, but not modify the library structure or edit items in the library.
- **Editing and Assembly:** Imports the library in such a way that you will be able to edit the items in the library, assemble them, and modify the structure of the library.
- **Cancel:** Cancels the import process without making any changes to the library.

To learn more about importing a library follow the link below:

• Import One Library Into Another

Import One Library Into Another

You can import a copy of another library into the current library. The imported library will become a folder in the current library.

There are three types of imported libraries. Which type you choose depends on how you need to use the library and its contents:

- **Assembly Only:** You can use the library to assemble documents, but you can't change the library's (or folder's) structure or edit a template. This type of import is useful, for example, when you want to access company templates stored on a network drive, but you don't have authorization to edit the templates or the imported library's structure.
- Editing and Assembly Using Original Files: You can use the library to assemble documents and edit templates. You can also modify the structure of the library. The imported library references the same files as the original library. (Templates and clause libraries can be referenced in more than one library.) If you edit a template from either the imported library or the original library, changes you make will appear in documents produced from either library. This type of import is useful when you are maintaining company templates stored on a network.
- Editing and Assembly Using Copies: You can use the library to assemble a document and edit templates. You can also modify the structure of the library. However, this method copies the files to a new location, rather than refers to the originals. Any changes you make to the templates are applied only to the copies and not the original files. This is useful if you are copying a library from a removable disk or network on your local disk.

When importing a library, HotDocs looks for the items referenced in the library in the same folder as the library itself. If they are not saved together, importing will not work.

To import one library into another

- 1. At the HotDocs library window, select the folder where you want the imported library to be located.
- 2. Click Import Library (File menu). The Import Library dialog box appears.
- 3. Select the library you want to import and click **Import**. The **Import Library** dialog box appears.
- 4. Choose how you want to import the templates:
 - Click Assembly Only to assemble documents from the templates—and nothing else.
 - Click Editing and Assembly to edit the templates and assemble documents from them.
- 5. If you select **Editing and Assembly**, HotDocs displays another dialog box, prompting you to choose whether to copy the associated files or use the originals:
 - Click **Yes** to import copies of the files to a new location you specify. Changes you make to the templates will affect files in the new location only. The **Browse for Folder** dialog box appears, prompting you to specify where the copied files should be placed.
 - Click **No** to have the library reference the original files. Changes you make to the templates will affect everyone using those files to assemble documents.

Print a List of Items in a Library

You can print a list of the files and folders in the library currently displayed in the HotDocs library window, allowing you to see the library structure, as well as the file path and the description of each item.

To print a HotDocs library

- 1. At the HotDocs library window, open the library whose contents you want to print. (See Open a Library.)
- 2. Click the d Print Library button. The Print dialog box appears.
- 3. Click **OK**. The list of titles, file names, and descriptions are printed at the selected printer.

To preview what the printout will look like before you print it, select **APrint Preview** (File menu). To return to the library window, click **Close**.

Save a Library

HotDocs saves all changes you make to a library immediately after you make the change. Clicking **Save** (**File** menu) simply displays a message confirming this

Save the Contents of a Library as a Text File

You can save the contents of a HotDocs library—including folder names, template file names, titles, and descriptions—as a text file. The hierarchy of the library is preserved within the text file. This is useful if you want to spell check the text used in your library. (To do this, you must open the text file in a word processor with spell-checking capabilities).

To save the library as a text file

- 1. At the HotDocs library, select **Export Library To > Plain Text File** (**File** menu). The **Save As** dialog box appears.
- 2. Select a location to save the text file to, enter a name for the file, then click **Save**. A .TXT file containing a tabbed list of the folder names and template titles is saved to the location you choose.

Move Items Within a Library

To organize your library, you can move an item to a different location in the same library by dragging, by cutting and pasting, or by copying and pasting.

These methods do not move the actual file to a different location on your local disk or network. See Move Items to New Locations on Disk for information on doing this.

To drag library items to a new location

• At the HotDocs library window, select the library item and drag it to the new location using the mouse. (As you drag, HotDocs uses a horizontal bar to indicate where the item will be placed when you release the mouse button.)

To cut and paste several library items

- 1. At the HotDocs library window, select the items you want to move.
- 2. Click the *d* **Cut** button (or press **Ctrl+X**.)
- 3. Select the new location and click the **Paste** button (or press **Ctrl+V**).

To copy and paste several library items within the library

- 1. At the HotDocs library window, select the items you want to copy.
- 2. Click the **Copy** button (or press **Ctrl+C**).
- 3. Select the new location and click the **Paste** button (or press **Ctrl+V**).

Move Items to New Locations on Disk

Using the **Move** command (**Template** menu) at the HotDocs library window, you can move template and clause library files to new locations on your local disk or network. Using this command also updates the library reference to the file. (Moving files in other ways, for example, by using Windows Explorer, requires that you manually update the library reference.)

To move template and clause library files

- 1. At the HotDocs library window, select the templates and clause libraries you want to move.
- 2. Click Template menu>Move ≌.
- 3. Browse to the folder where you want to copy the template or templates.
- 4. Click **OK.** HotDocs removes all associated files except for inserted templates (component files, default answer files, and so forth) from the old location, pastes them at the new location, and then updates the references to the files in the HotDocs library.

HotDocs handles the **Move** command a little differently if you are moving the template from a folder that contains another template with the same file name but different file name extension (for example, if you are moving *Invoice.docx* from a folder that also has *Invoice.wpt*). In such a case, HotDocs copies the associated component file to the new folder, but does not delete it from the original folder because it is still needed by the other template.

You can also copy templates to new locations on disk and have the library reference updated. See Copy Templates to New Locations for details.

When a template has insert fields, HotDocs does not move those inserted templates along with the parent template. If you want to move a template that has insert fields you can use the Publishing Wizard to publish the initial template to the new location as a standard template file and HotDocs includes a copy of all the files necessary for assembly.

Copy Templates to New Locations

Often you need to make a copy of a template and save it to a new location on disk. You can use the **Copy** command at the template library to do this. As you are copying, you can choose to make the template and component file read-only. You can also choose to update the reference to the file in the library so the library refers to the copied template rather than the original.

Be aware that copying a template creates two templates with identical template IDs. This can lead to template ID conflicts should you upload both the original and copied templates to a web server.

To copy the template to a different location

- At the HotDocs library window, select the templates and clause libraries you want to copy. (Press Ctrl or Shift to select multiple items.)
- 2. Click **Template** menu>**Copy**.
- 3. Browse to the folder where you want to copy the template or templates and click **OK**.
- 4. You can then set some properties for the copied templates:
 - Select **Mark newly created templates as read-only** to keep users from editing the template. (This assigns the read-only property to both the template and the component file.)
 - Select **Update library entries** to refer to the newly created templates to have HotDocs change the file path for the library item so that it uses the file path for the copied template.

When a template has insert fields, HotDocs does not copy those inserted templates along with the parent template. If you want to copy a template that has insert fields you can use the Publishing Wizard to publish the initial template to the new location as a standard template file and HotDocs includes a copy of all the files necessary for assembly.

Assembling Documents

Completing an Interview

Overview: Assemble a Text or Form Document

There are two types of documents you can assemble—text documents and form documents:

- Text documents are documents you view in your word processor. Once the document is assembled, you can edit any of the text.
- Form documents are those in which the underlying text is *static*, meaning it cannot be modified. Only the answers that are merged into the document can be modified. (An IRS form is a good example of a form document.)

The process of assembling both of these types of documents is very similar. However, there are specific things you should know about assembling form documents. Please see Introduction: Assemble a Form Document for a description of these differences.

When you assemble a document, HotDocs displays the assembly window, which is divided into three panes—the *interview outline*, the *dialog pane*, and the *resource* pane. The interview outline shows a list of dialogs (or groups of questions). You can click on any dialog in the outline and view (and answer) the questions in the dialog pane. Finally, where provided, the resource pane shows information to help you answer questions in the dialog.

As you answer questions, HotDocs merges your answers into the assembled document, which you can view by clicking the **Document** tab at the assembly window. As you are reviewing the assembled document, if you see an answer that needs to be edited or changed, you can make the change right in the **Document Preview** tab. (Some template providers may prohibit you from changing answers at the **Document Preview** tab. If this is the case, the buttons and commands that control this feature will be disabled.) Additionally, if the template provider has allowed it, you can edit the actual text of the document while viewing the **Document Preview** tab. (These features are available to Microsoft Word users only.) You can also use the document navigation bar to move between answers in the document.

If you need to generate quick views of the document—for example a list of just the questions or a list of questions and answers (without all of the other document text)—you can view the **Question Summary** tab or the **Answer Summary** tab. Both provide you with a short list of information required in the document. You can also view a spreadsheet of the template's variables and any answers that have been given.

Once all of the answers are provided and the document is correct, you can send a copy of the document to a word processor (if you are assembling a text document) or you can view the assembled document at the **Form Document** tab (if you're assembling a form document). If no further editing is needed, you can print the document or save it to disk. You also can save the answers you've provided in an answer file, which can then be used to assemble other documents.



The following diagram shows the assembly process:

At a Glance: The Assembly Window

🖞 Employment Agreement - HotDocs	Developer	- • •
File Edit View Navigate Tools	Help Menu Bar	
New Answer File	- 19 🕾 🖓 🕞 🚽 🖾 💟 🍄 🗟 🗟	
2 ~ ~ & 🖬 🖺 3+ 3+ 3	I+ ∃+ ≵↓ 🚄 📲 🕤 <mark>(Tool Bar</mark>)	
Interview Document Preview	Question Summary Answer Summary Tabs	
Employee Information	Employee Information	
Agreement Information	Employee Name	
End of Interview	Employee Gender Male	
	Female	
Interview Outline		
	Interview Pane	
	Nationalian Bar	
	M First Al A Dreviour Next No. Las	+ N Einich El
	N TIER N LIENDUS MERL E	
		^
	Resource Pane	-
5		

When you choose to assemble a template HotDocs will open an **Interview** window. Using this window you can enter answers, view resources, view summaries, and preview the finished document.

The main portion of the **Interview** window is split into 3 panes. The Interview Outline, the Interview Pane, and the Resource Pane.

The Interview Outline shows a list of dialogs within the interview. Next to each dialog is an icon indicating the dialogs status and type:

□None of the questions in the dialog are answered.

At least one of the questions in the dialog is answered.

■All of the questions in the dialog are answered.

ERepeats the dialog, allowing you to enter more than one set of answers.

➡ Allows you to enter a new set of answers in a list.

In the Interview Pane you can see questions on the selected dialog and spaces to type or select your answers. These can be in the form of text or number fields, check boxes, spreadsheets or drop down menus.

Some dialogs or questions will have extra resource information to help the user answer the question (see Add Resource Information to a Variable or Dialog). This information can be displayed in the Resource Pane.

Between the Resource Pane and the Interview Pane you can see the Navigation Bar. This contains the following buttons to help you move though the interview:

- **H**First Dialog
- •
- Previous Dialog
- Next Dialog
- Next Unanswered Dialog
- H Last Dialog
- 🔊 Finish Interview

Interview is just one of the five available pages you can look at from this window. You can move to the other pages by clicking on one of the Tabs below the Toolbar:

- Document Preview: A preview of the finished document showing the answers in place.
- **Question Summary**: Displays a list of questions that are asked in the interview with space for answers. This can be printed and used as a paper questionnaire version of the interview.
- **Answer Summary**: Displays a summary of questions in an interview as well as any answers you entered.
- **Variable Sheet**: Displays a spreadsheet of variables, prompts, current answers, and variable types. You can sort the spreadsheet by clicking the spreadsheet column headings.

You can click on the Interview tab to return to the interview you are working on.

Above all of the tabs are the Menu bar and Toolbar. From the Toolbar you can access the following buttons:

- **d**HotDocs Library: Displays the HotDocs library window.
- **Recent Answer Files Drop-down List:** Displays a list of recently used answer files. Click this button to select a different answer file.
- Mew Answer File: Replaces the current answer file with a new, empty answer file.
- **Open Answer File:** Opens the Open Answer File dialog box so you can select another answer file for the interview.
- **Save Answer File:** Saves changes to the answer file.
- Send Document to: Sends a copy of the assembled document to a word processor or HotDocs Filler, depending on the type of template you are using. Once in the word processor or HotDocs Filler, you can make post-assembly changes to the document. Be aware, however, that any changes you make will only affect that document -- your changes will not be saved in the answer file.
- Save Document: Saves the document to disk. The Save As dialog box will appear, allowing you to specify a name and location for the saved file.
- **d** Print Document: Prints a copy of the current document.
- Instant Update: Determines when HotDocs updates the interview. If the button is pressed down, this happens when you move between answer fields. If it is not pressed down, this happens when you navigate to another dialog. This can help speed up how quickly you are able to move through answer fields in the dialog.
- Show Resource Pane: Shows and hides the resource pane where helpful information about a question or group of questions can be viewed.
- **Check Spelling:** Checks the spelling of text answers entered during the interview.
- Show Assembly Queue: Displays the assembly queue, which lists all of the templates you have chosen for assembly. The queue gives you greater control over the assembly process when you are assembling multiple documents.
- **Interpretent Section 2** For the HotDocs Options dialog box where you can specify several preferences that control how HotDocs works.
- **Whelp:** Opens the relevant page of the HotDocs Help File.
- **Erase Answer:** Clears the answer from the answer field.
- **Undo:** Undoes the most recent change made to an answer.
- **Redo:** Restores changes that were made to an answer.
- **Cut:** Cuts the selected text and saves it to the Windows Clipboard. You can then paste the text into another answer field.
- Copy: Copies the selected text and saves it to the Windows Clipboard. You can then paste it elsewhere.
- **Paste:** Pastes any text that has been saved to the Windows Clipboard into the current answer field.
- **Insert Repetition in a Spreadsheet:** Inserts a new, empty repetition above the current repetition.
- **Delete Repetition in a Spreadsheet:** Deletes the selected repetition from the interview. Any answers you have entered will be lost.
- **I** Move Repetition Up in a Spreadsheet: Moves the selected repetition up in the list of answers. You should be aware, however, that if the template provider has assigned sorting options to the list, answers in the assembled document will be listed in that order instead of the order you specify here.
- **=**+**Move Repetition Down in a Spreadsheet:** Moves the selected repetition down in the list of answers. You should be aware, however, that if the template provider has assigned sorting

options to the list, answers in the assembled document will be listed in that order instead of the order you specify here.

- **Sort Spreadsheet:** Sorts columns in a spreadsheet in either ascending or descending order. To sort on a second level, include those search options on the second line of the Sort Spreadsheet dialog.
- **Highlight Fields:** Highlights all answers or unanswered variables at the document preview tab.
- **Edit Document Text:** Opens the Document Text Editor where you can edit modifiable text within the document.
- **Go to Answer in Document (at Interview tab):** Positioning your curser within an answer field and clicking this button will take you to the answer at the Document Preview tab, letting you view the answer in place.
- **Context** Go to Answer in Interview (at Document Preview tab): Positioning your curser at an answer on the Document Preview tab and clicking this button will take you to the corresponding answer field in the Interview tab.

When assembling a form document additional tool bar options become available:

- **Field Properties:** Changes the properties for the selected field or fields.
- **AFont:** Changes the font for the selected field or fields.
- **Override Answer:** Allows you to type any answer in the field, even if the field requires a certain type. For example, you can override a number-only field and type a text answer.
- **Fill Tool:** Allows you to enter information in the field, such as an answer.
- **Select Tool:** Activates the Select Tool. This button allows you to select an existing field so that you can move it, resize it, or edit its properties (including the variable properties).
- **Scroll Tool:** Makes the Scroll Tool the current cursor. Use the cursor to grab the current form page and drag it around the window.
- **Zoom Tool:** Activates the Zoom Tool, allowing you to increase or decrease magnification of the template.
- **View Page Width:** Displays the entire width of the current form page in the window.
- **View Full Page:** Shows the whole form document in the window.
- **Show Fields:** Shows and hides the colored field backgrounds during direct-fill assembly of a form document.
- **Show Thumbnails:** Shows and hides thumbnail images of each page in the form. You can use these pages to move through the document.
- **Previous Page:** Moves to the page just before the current page in the form.
- **Next Page:** Moves to the page just after the current page in the form.

In the Menu Bar you can access further options not available on the Toolbar.

To learn more about assembling templates follow the links below:

- Introduction: Assemble a Text or Form Document
- Adjust HotDocs Windows and Panes
- Update Your Interview Outline and Document
- Preview a Template Before Assembling It
- Assemble a Text or Form Document
- Assemble an Interview Template
- Assemble a Document Without Showing an Interview
- Assemble Multiple Documents
- Assemble a Clause Library from the Template Library
- Navigate Through Answer-Gathering Dialogs
- View Resources During Assembly
- Use Repeated Dialogs During Assembly
- Use Inserted Dialogs During Assembly
- Select Clauses During an Interview
- Use the End of Interview Dialog
- Spell Check Answers

Adjust HotDocs Windows and Panes

You can arrange the HotDocs windows and panes to match your work preference.

To work with HotDocs windows

At the HotDocs window, complete any of the following steps:

- To move a window, click on the title bar and drag the window to another location.
- To maximize or minimize the window, click either the **Maximize** or **Minimize** buttons in the window title bar. The **Maximize** button causes the window to fill the entire screen, while **Minimize** closes the window and places it on the Windows taskbar.
- To resize the window to a specific size, move the mouse pointer to a border or corner of the window. When the icon becomes a resize arrow, click and drag the border to the desired position.
- To resize a pane within a window, move the mouse pointer to a border between two panes. The icon becomes a resize arrow. Click and drag the border to the desired proportions.
- To rearrange toolbars, move the mouse pointer to the anchor bar of one of the toolbars. Drag the toolbar to the desired location in the window.

When moving toolbars, you can separate toolbars from the assembly window. To reattach the toolbar, move it to a position over the menu bar. You can also remove toolbars by clearing the toolbar at the **View** menu.

Update Your Interview Outline and Document

Some answers you enter during an interview may cause HotDocs to update the appearance of the interview outline. Additionally, each time you change an answer while viewing the **Document** tab, HotDocs updates the entire document. Depending on the complexity of the interview or the document, this updating may take longer than desired. To minimize the time HotDocs takes to update the interview or document with your changes, you can turn off **CINSTANT Update** to make HotDocs update only when necessary.

When **Instant Update** is turned off, information in the interview and document may become outdated. When this happens, the **Instant Update** button changes appearance. You can click the button and HotDocs updates the interview or document so that all information will be current. HotDocs also updates the document and interview when you move between tabs in the assembly window, as well as when you print, save, or send the assembled document to the word processor.

To control when HotDocs updates the interview outline or document

While viewing either the **Interview** tab or the **Document** tab at the assembly window, click the **Z Instant Update** button:

- When the button is selected (or pressed in), HotDocs updates the interview outline as you move between answer fields. If you're viewing the **Document** tab, it updates the entire document.
- When the button is not selected (or not pressed in), HotDocs updates the outline only when you move between dialogs. If you're viewing the **Document** tab, it updates just the answer field you are currently viewing (and any other answer fields that merge the same answer).

Preview a Template Before Assembling It

To help you identify the correct template, you can preview the template before you select it for assembly. The **Preview** tab shows the text of the document, as well as where HotDocs variable fields are merged.

To preview a template

1. At the HotDocs library window, select a template.

2. Click the **Preview** tab. The library window changes to show the template text.

To remove the **Preview** tab from the library window, choose **Preview Tab** (View menu).

To move the **Properties** and **Preview** tabs to the top of the pane, select **Tabs at Top** (View menu.)

The **Preview** tab shows the information in the template correctly; however, it may not be formatted correctly—particularly when previewing WordPerfect templates. Additionally, previewing form templates will not show you variable fields.

You can preview a simpler version of the template by choosing **Markup View** at the template library window. This changes the preview so it shows simple mark-up fields, rather than complicated automation. (**Markup View** is supported for Word templates only.)

Search for a Specific Template in a Library

Some libraries may contain a long list of templates. Instead of browsing through the template list, you can search for a specific template or clause library based on the template's title, file name, or description.

To search for a template or clause library

- 1. At the HotDocs library window, select the **Find** check box, located below the template list. The template list changes, showing each template in the library in alphabetical order.
- In the **Find** field, type the text that is in the title, description, or file name of the desired template. The template list changes to show only the items that contain the text you type. (The **Find** text field is not case-sensitive.)
- 3. To view the entire library again, clear the **Find** check box.

To show file names instead of template titles in the file list, choose **File Names** from the **View** menu. To view template titles again, choose **Template Titles**.

Assemble a Text or Form Document

You assemble a document by selecting a template and then answering the questions in the interview. Answers to your questions are then merged in the correct places in the document. Once all of the questions are answered, you can view, print, or save the document.

To assemble a document

- 1. At the HotDocs library window, select a template. (If you're assembling a form document, see Introduction: Assemble a Form Document for additional instructions.)
- 2. Click Resemble. The Answer File dialog box appears. (You can also start an interview by double-clicking the template.)
- 3. Select an answer file (see Select an Answer File for Assembly), then click **OK**. The assembly window appears, showing the interview outline and the first information-gathering dialog.
- Answer the questions in each dialog, clicking Next to advance to each new dialog. (See Navigate Through Answer-Gathering Dialogs.) (You can also advance to the next dialog by pressing Alt+N, Enter, or Page Down.)

After the final dialog, the **End of Interview** dialog identifies how many questions are unanswered. This dialog also provides options for working with the assembled document. See Use the End of Interview Dialog for specific details.

HotDocs updates the interview as you enter answers in a dialog. If you experience a significant delay when moving from one answer to the next, you can have HotDocs update the interview less frequently. (See Update Your Interview Outline and Document.)

At a Glance: The Answer File dialog box

e to use with this assembly: wer File	Answer File	? ×
wer File atically select this answer file for use with the next assembly the assembled document to Microsoft Word without displaying an interview	Answer file to use with this assembly:	BG
natically select this answer file for use with the next assembly the assembled document to Microsoft Word without displaying an interview	New Answer File	- 🙆 🖲
the assembled document to Microsoft Word without displaying an interview		
	Automatically select this answer file for use with the	ne next assembly
	🗌 Automatically calent this assure file for use with th	a next secondly

When you click to assemble a template you will usually see the **Answer File** dialog box appear. Unless you have set a default answer file (see Create a Default Answer File) then the **Answer File** field \triangle will show the option **New Answer File** and when you click **OK** it will start a blank answer file which you can choose to save during assembly or after assembling the template.

Clicking on the **Answer File** field A will allow you to choose from a drop-down list of recently used answer files.

If you would like to use a previously saved answer file click on the **Open Answer File** button **b** where you will be taken to **Answer File Manager** to choose the file you would like to use for this assembly.

Once you have chosen an answer file the two check boxes D ungray. The top check box allows you to have HotDocs automatically select this answer file the next time you assemble this template. The bottom check box allows you to send the assembled document to Microsoft Word without displaying the interview. This is useful if you have a completed answer file that you don't wish to edit as HotDocs will immediately create the document.

If after selecting an existing answer file you decide you would instead like to use a new answer file then clicking the **Answer File** button **G** will return the **Answer File** field to displaying **New Answer File**.

To find out more about selecting an answer file follow the link below:

• Select an Answer File for Assembly

Select an Answer File for Assembly

When you assemble a document, you must choose whether to use a new answer file or an existing file. After you complete each interview, HotDocs can save your answers for use with other assemblies.

To select an answer file

- 1. At the HotDocs library window, select a template, then click **Assemble**. The **Answer File** dialog box appears.
- 2. Accept the suggested answer file or change the answer file by choosing one of the following options:
 - Click the **Open Answer File** button to open the **Open Answer File** dialog box where you can choose an existing answer file. After making your selection, click **Open**.
 - Click the **New Answer File** button to create an empty, untitled answer file.
 - Select **Automatically select this answer file for use with the next assembly** to have HotDocs suggest this answer file for all subsequent documents you assemble.
- 3. At the Answer File dialog box, click OK. The assembly window appears

You can also use Windows Explorer to access your answer files. See Introduction: Use Answer Management.

Some templates may have an answer file assigned by the template provider. For templates like this, the **Answer File** dialog box does not appear.

Assemble an Interview Template

An interview template does not produce an assembled document. Instead, it gathers and (and sometimes automatically saves) information (such as court, attorney, or client information) that can be used in assembling other documents. An interview template is distinguished from other templates in the library by its icon:

For example, a template library includes an interview template named *Client Information*. This template gathers personal information about each client—such as name, address, gender, and birth date. This information may be needed in other templates in the library. Before you ever assemble any of these other documents, you could assemble the interview template and create a set of answers. Then, whenever a dialog in another template requires client information, you can access the answers you've already entered.

To assemble an interview template

- 1. At the HotDocs library window, select the **interview template** and click **Assemble**. The **Answer File** dialog box appears. (If the template is already associated with an external answer source, you will not be prompted to select an answer file. Skip to step 3.)
- 2. Select an answer file, then click **OK**.
- 3. Answer the questions in each dialog, clicking **Next** or **Previous** to move between dialogs.
- 4. At the end of the assembly, HotDocs may prompt you to save your answers in an answer file.
 - If this prompt appears, click Save or Save As to view the Save Answer File dialog box, where you can save the answers you've provided. Or, click Don't Save to discard the answers.
 - If this prompt doesn't appear, the template is probably associated with an external answer source, so HotDocs automatically saves the answers to that answer source.
- 5. Close the assembly window.

Assemble a Document Without Showing an Interview

You can assemble a document without answering any questions in the interview. When you do this, HotDocs uses the answer file you specify and assembles the document. Depending on which *Finish Interview Action* options you have selected, you can choose where you will view the assembled document—either in the word processor (text templates), HotDocs Filler (form documents), or at the **Document** tab of the assembly window. (See Control What Happens When You Finish an Interview.)

To assemble a document without viewing the interview

- 1. At the HotDocs library window, select one or more templates.
- 2. Click **Assemble**. The **Answer File** dialog box appears.
- 3. Select an answer file. (See Select an Answer File for Assembly.)
- 4. Select **Send / Go directly to ... without displaying an interview** and click **OK**. HotDocs assembles the document using the answers in the answer file you selected. It then displays the assembled document in the application specified at HotDocs Options.

Overlay Answers in the Current Answer File

When you assemble a document, you can use answers from an existing answer file and overlay them in the current answer file, overwriting any existing answers in the current answer file.

For example, perhaps you have an answer file that contains only client-specific answers, such as names and addresses. You can use these answers to assembe several related documents. However, to use this answer file in the traditional way (selecting it at the **Answer File** dialog box), you would most likely add answers to the file, making it less useful in assembling other documents. To make the file useful for multiple templates, you can select a new or existing answer file at the **Answer File** dialog box, and then use the answer file that contains your client-specific answers as an overlay answer file. Answers you enter or change during the interview are saved to the current answer file—not the overlay answer file.

To use an overlay answer file during assembly

- 1. At the HotDocs library window, select a template and click ***** Assemble. The Answer File dialog box appears.
- 2. Select an answer file, and click **OK**. The assembly window appears.
- 3. Click File, then click Overlay Answers. The Open Answer File dialog box appears.
- 4. Select an answer file to overlay, and click the **Open** button. The assembly window appears again, showing information from the overlay answer file; however, the answer file you selected in step two is still assigned (as seen in the **Current Answer File** drop-down list).

HotDocs 5 users: You can no longer save or load pattern answer files. Instead, to merge specific answers from one file into the current answer file, use the **Overlay Answers** command at the **File** menu of the assembly window. To distinguish overlay answer files from regular answer files (as you could with pattern answer files), create a separate folder in Answer File Manager and store your overlay answer files there.

You can also specify an overlay answer file at the command line.

Assemble Multiple Documents

You can select multiple templates or clause libraries, and then start one assembly process that will address each template. Each template or clause library will create a separate document, using the same answer file, unless you select a different answer file during the interview. (See Switch Answer Files During Assembly.)

To assemble multiple documents

- 1. At the HotDocs library window, press **Ctrl** or **Shift**, then select all the templates and clause libraries you want to assemble.
- 2. Click **Assemble**. The assembly queue status box (located in the status bar of the assembly window) displays how many assemblies are waiting, and the **Answer File** dialog box appears.
- 3. Select an answer file and click **OK**. (Select an Answer File for Assembly.) The assembly window appears.
 - If the item is a template, the first answer-gathering dialog is shown. (See Assemble a Text or Form Document.)
 - If the item is a clause library, the clause library is shown. (See Select Clauses During an Interview.)
- 4. Provide answers or select clauses for the interview.
- 5. After completing the interview, the *End of Interview* dialog notifies you that other assemblies are ready to begin.
- 6. Close the assembly window, and begin assembling the next document.

The assembly queue and assembly queue status box (located in both the library and the assembly window status bars) can help you track the assembly of each document. See Use the Assembly Queue for details.

At a Glance: The Assembly Queue Dialog Box

Assemble	Auto Start 🔺 🤜	×	🎸 Clean Up	0
Template		- 14	Status	
	0			

When you have the assembly window open you can click on the *E* **Assembly Queue** button to view the **Assembly Queue** dialog box.

At the top of the dialog is the **Assembly Queue** toolbar A where you can access the following options:

- Resemble: Manually starts the next assembly. (This option is available if Auto Start is not selected.)
- **Auto Start:** Switches the assembly mode between automatic (where HotDocs automatically assembles documents listed in the queue) and manual (where you must manually click the Assemble button for each document listed in the queue).
- **Move up:** Moves the selected entry closer to the top of the queue.
- **Move down:** Moves the selected entry closer to the bottom of the queue.
- **XDelete:** Removes the selected template from the assembly queue. You can only delete templates that are not currently being assembled.
- **V** Clean up: Removes entries for assemblies that have been completed.
- **Whelp:** Opens the relevant page of the HotDocs Help File.

Below that is the **Assembly Queue List** where you can see the templates selected for assembly and their current assembly status. Templates will be assembled in this order, so change the order, select a template and click the **Move Up** or **Move Down** button.

To learn more about the Assembly Queue follow the link below:

• Use the Assembly Queue

Use the Assembly Queue

The assembly queue shows a list of templates—some that may be waiting to be assembled and some that may have recently been assembled. Using the assembly queue, you can start and stop assemblies, change the assembly order, and change how assemblies are started. In addition, the assembly queue status box (which is located in the status bar of the template library window and assembly window) also shows when templates are queued for assembly.

The assembly queue provides two methods for starting assemblies: auto mode and manual mode. These methods are controlled by the **PAuto Start** button in the assembly queue toolbar. When assembling multiple templates and **PAuto Start** is selected, HotDocs automatically starts each assembly, following the order in which the templates are added to the queue. When assembling multiple templates and **PAuto Start** is not selected, HotDocs starts the first assembly, but then waits for you to start each subsequent assembly by clicking the **PAuto Start** button on the assembly queue toolbar.

As templates are added to the assembly queue, they are assigned a status, which helps you identify each template's stage in the assembly process. (For a description of template statuses, click here.)

To add Items to the assembly queue

HotDocs will allow you to assemble only one document at a time. If you are currently assembling a document and you attempt to assemble another one, HotDocs will add it to the assembly queue and it won't be assembled until the current assembly is complete. If there are multiple templates in the queue, HotDocs will assemble them in the order they are added.

- 1. At the **End of Interview** dialog select the option to assemble the template.
- 2. If another template is currently being assembled, the **Multiple Assemblies** dialog box will appear.
- 3. Click Yes to add the template to the Assembly Queue.
- 4. You can also tick the check box at the bottom of this dialog, **Don't warn me about multiple assemblies again**, to keep this dialog box from appearing. The next time you attempt to assemble a document when HotDocs is already busy assembling, the template will be added to the queue.

To work with templates in the assembly queue

- 1. At the HotDocs library window, click the **Seembly Queue** button. The **Assembly Queue** dialog box appears. (You can also select **Assembly Queue** (**Tools** menu) or double-click the assembly queue status box in the status bar of the template library window.)
- 2. At the library window, select multiple templates and click **Assemble**. Entries for each template appear in the assembly queue. Then the **Answer File** dialog box appears. (See Assemble Multiple Documents.)
- 3. At the **Answer File** dialog box, select an answer file and click **OK**. (See Select an Answer File for Assembly.) The first dialog appears in the assembly window.
- 4. Throughout the interview process, perform any of the following tasks using the following commands in the Assembly Queue:

То	Do This
Change the order in which pending or confirmed templates are assembled	Select the item in the assembly list and click either the Up button or the Down button. (Templates that are currently assembling cannot be moved in the queue.)
Remove a confirmed assembly	Select the item in the assembly list and click the Delete button. (Templates that are currently assembling cannot be removed from the queue.)
Change whether assemblies start automatically or manually	Click the Auto Start button. When Auto Start is turned off, clicking the Assemble button at the HotDocs library only starts the first assembly. To begin subsequent assemblies, you must also select an entry in the assembly queue and click the Assemble button there.
Manually start an assembly	Click the 🏁 Assemble button.
Remove the entries for all completed assemblies from the Assembly Queue	Click the V Clean Up button.

When HotDocs closes, it remembers the assembly queue display setting. If you close HotDocs while the queue is showing, the queue will automatically be displayed the next time you open HotDocs.

You can have completed assemblies automatically removed from the assembly queue. See Include List of Completed Assemblies in Assembly Queue for details.

Assemble a Clause Library from the Template Library

A clause is a section of text, usually a commonly used paragraph, and a clause library is a collection of clauses. You can select clauses from a clause library and use them to assemble text documents. (See Introduction: Use Clauses and Clause Libraries.)

In HotDocs, you can view a clause library and select clauses for inclusion in a document both at the template library and during the interview for a text template.

• When a clause library is referenced in a template library, you assemble the clause library and then choose where the assembled clause text is merged. You can choose to send the assembled clauses to a new, empty document; copy the assembled text to the Windows Clipboard (so you can paste it into an existing document); or paste the assembled text directly into an open word processor document. (Instructions for this method are included below.)

• When a clause library is asked during the interview of a text template, the clause text is automatically inserted into that assembled document.

To assemble a clause library from the template library

- 1. At the HotDocs library window, select a clause library and click **Assemble**. The **Answer File** dialog box appears.
- 2. Select an answer file and click **OK**. The clause library appears in the dialog pane.
- 3. (Optional) Perform either of the following tasks:
 - Select the **Find** check box to find all of the clauses with specific text in their title or description. In the text field, type the text for which you are searching.
 - Click **Preview** to view the clause text in a pop-up dialog box. To return to the clause library, click the X in the upper-right corner of the pop-up dialog box.
- From the Available list, select each clause you want to use and click ♥ Select to add it to the Selected list. To remove a clause, select it, then click ♥Remove. (Press Ctrl or Shift to select multiple items.)
- 5. **(Optional)** Reorder the clauses in the **Selected** list by selecting and dragging a clause to the desired position. The clauses will be inserted into the document in this order.
- 6. When all the necessary clauses are in the **Selected** list, click **Next** on the navigation bar.
- 7. Answer any questions in the interview. Once all questions are answered, complete any of the following steps:
 - Click the Send the assembled document to the word processor button (or press F11) to open the assembled document in your word processor.
 - Click Paste the assembled document into the open word processor document (or press Shift+F11) at the *End of Interview* dialog. This will insert the assembled document into an open word processor document.
 - Microsoft Word users: Choose File > Send Document To > Clipboard (or press Ctrl+F11) to place a copy of the document on the Windows Clipboard. You can then open an existing document and paste the text in the desired location.

Navigate Through Answer-Gathering Dialogs

To assemble a document, you must answer interview questions. To help you go through the interview easily, you can use the navigation bar, located at the bottom of the dialog pane.

As you answer questions, the icons in the interview outline change color. This can help you identify how much of the interview is left to complete:

None of the questions in the dialog are answered.

At least one of the questions in the dialog is answered.

■All of the questions in the dialog are answered.

ERepeats the dialog, allowing you to enter more than one set of answers.

➡ Allows you to enter a new set of answers in a list.

The final dialog in an interview is usually the **End of Interview** dialog, which reports how many questions are unanswered. This dialog is customizable and also provides options for working with the assembled document.

To navigate through HotDocs dialogs

- 1. At the HotDocs library window, select a template and begin assembling the document.
- 2. The HotDocs assembly window appears. Complete any of the tasks using commands listed in the following table:

То	Do This
Enter an answer for the question	Depending on the type of answer required, type a text or number answer, enter a date, select a check box, or choose an option from a list of options.
	When entering a multi-line text answer, you can control whether the Enter key inserts a line break or a paragraph mark in the assembled Word document. To enter a line break, simply press Enter . To enter a paragraph break, press Ctrl+Enter . (In some situations, the template developer may have specified which break to use.)
	When entering a patterned time of day, enter the hour first, then press the right arrow key to move past the colon to enter the minutes. You can also click the up and down arrows next to the answer field to select the hours and minutes.
Move to another dialog in the interview outline	Click a dialog icon in the interview outline.
Go to the next or previous dialog	Click either the Next Dialog button or the Previous Dialog button. (You can also press Page Down/Page Up , or Alt+N/Alt+P .)
Go to the next dialog or the previous dialog that contains an	Click either the Next Unanswered button or the Previous Unanswered button. You can also press Ctrl+Page Up/Ctrl+Page Down .

unanswered variable

Go to the first or Click either the **First Dialog** button or the **Last Dialog** button. **the last dialog in the interview**

Finish the interview and view the assembled document

View additional information about a specific answer or the contents of a dialog

View where the answer is merged into the document (This option is available to Word users only.)

template

Click the Finish Interview button. (Where you view the assembled document depends on the Finish Interview action you've specified.

Click the **View Resource** button.

Click the **Go to Answer in Document** button. HotDocs displays the **Document** tab and moves the cursor to the first place in the document where the answer is merged. However, if you place your curser in the answer field of an answer that is not directly used in the document HotDocs will not display the document tab and you will see an error message.

Hide the Clear Interview outline (View menu).

Preview an
assembled text
document
before sending
it to the word
processorClick the Document Preview tab.View anClick the Form Document tab.

assembled form document

View a list of all Click the Question Summary tab. questions in the

View a list of all Click the Answer Summary tab. the questions and their

answers in the interview	
View a spreadsheet of	Click the Variable Sheet tab.
the variables	
and their current	
answers	

HotDocs **updates the interview** as you enter answers in a dialog. If you experience a significant delay when moving from one answer to the next, you can have HotDocs update the interview less frequently.

Sometimes the format of your answer may be changed. To verify these changes, you can specify an option that forces HotDocs to warn you of answer format changes before moving to the next dialog.

To display the different tabs along the bottom of the window (instead of the top), clear **Tabs at Top** in the **View** menu.

View Resources During Assembly

Template providers can include helpful information with dialogs and questions in the interview. When such information is available, it appears in the resource pane. (To view the resource pane, either choose

Resource Pane (View menu) or click the **Resource** button in the assembly window toolbar.)

To view resource information

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. Navigate to a dialog that shows a resource.
 - If the resource pane is already displayed, the information automatically appears there. (You can change the size of the resource pane by selecting the top border of the pane and dragging it up or down.)
 - If the resource pane isn't displayed, you can either click the **Resource** button in the toolbar or click the **Resource** button that appears next to the answer field.

Resources help you answer questions in the interview. For help using HotDocs, see Get Help Using HotDocs.

At a Glance: The Save Answer File Dialog Box

8	×
B	Q
OK Cance	:k
	ु ि OK Cance

When you come to save an Answer File, either during assembly or after assembling a template, you will be presented with the **Save Answer File** dialog box.

In the first field A you can enter a file name for the answer file and using the \square **Browse** button B the right of this field you can change where the file will be saved on your computer. If you do not change where the file is to be saved it will be saved to the **Answers** folder in your **HotDocs** folder usually in **My Documents.**

In the next field **G** you can enter a Title for the answer file. This will be the name displayed when you see the answer file in **Answer File Manager** or when you use it in a HotDocs Interview.

In the last field **D** you can enter a brief description of the answer file. This will display with the answer file in **Answer File Manager** and may help you to identify the answer file you need the next time you come to use it.

Below these text fields is a check box . Checking this box will have HotDocs save the answer file in a format that is compatible with older versions of HotDocs (see Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs in the Quick Start Guide). This can be useful when you intend to share the answer file with someone who may be running a different version of HotDocs to you.

To learn more about saving answers files follow the link below:

• Save an Answer File During Assembly

Save an Answer File During Assembly

While assembling a document, you can save the answers you have entered. This allows you to save your work and perhaps start a different assembly without closing the assembly window. Saving answers also allows you to use the information you enter with other templates, thus saving you time.

To save answers during assembly

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. At the assembly window, click the **Save Answers** button. If you are using an existing answer file, the new answers are saved. If you are using a new answer file, the **Save Answer File** dialog box appears where you can perform any of the following tasks:

То	Do This
Enter the file name	At the File name field, enter a name. When you click OK , the answer file will be saved to the default <i>Answers</i> folder and a reference to it will be added to Answer File Manager.
	To save the answers to a location other than the default <i>Answers</i> folder, click the Rowse button and navigate to the location.
Enter the title that identifies the answer file in Answer File Manager	At the Title field, enter a name or accept the suggestion HotDocs makes.
Add information to help identify the file	At the Description field, type notes about the answer file's purpose or contents. (The description will appear at the Properties tab of the Answer File Manager. When searching for specific answer files, you can search for text in the description.)
Save the answer file so it is compatible with versions of HotDocs earlier than HotDocs 2009	Select HotDocs 2005-2008 answer file . (See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs.)

See Introduction: Use Answer Management for information on using Windows Explorer instead of Answer File Manager.

The Save Answers button may be disabled because the template provider wants to prevent the answer file from being altered. You can assemble a document using new answers, but when you close the assembly window, the new answers are automatically discarded.

Switch Answer Files During Assembly

While assembling a document, you may want to use another answer file. At any point during the interview, you can select a different answer file, then continue the interview using the new answers.

To switch answer files

- 1. At the HotDocs library window, select a template and begin assembling a document. (See Assemble a Text or Form Document.)
- During the interview, click Open Answers. The Open Answer File dialog box appears. (Depending on the file management settings you have selected, a Windows Explorer dialog box or a window from your document management program may appear instead. See Manage Answer Files.)
- 3. Select an answer file and click **Open**.
- 4. If prompted to save the old answers, click **Save**, **Save As**, or **Don't Save**, depending on your needs. (See Save an Answer File During Assembly.)
- 5. The assembly window appears again, using the newly selected answer file.

When viewing the answer library, you can sort the answer files. To do this, select a folder and click 2, sort. To search for a specific answer file, select the **Find** check box, then type the text for which you are searching in the text field.

Use Repeated Dialogs During Assembly

Dialogs can be repeatedly asked to gather several sets of answers for the same questions. A repeated dialog can appear as a series of dialogs or as a spreadsheet.

Once you have created a list of answers, you can use commands in the toolbar to move records within the list.

You can change the way repeated dialogs are represented in the interview outline. (See Display Repeated Dialogs Using a Special Icon.)

HotDocs updates the interview as you enter answers in a dialog. If you experience a significant delay when moving from one answer to the next, you can have HotDocs update the interview less frequently. (See Update Your Interview Outline and Document.)

To enter answers in a series of dialogs

- 1. At the repeated dialog, enter answers for the questions.
- Click the Add Another button to add another repetition of the dialog to the series. The repetition number next to the dialog icon increases each time you enter a new set of answers. (You can also press Alt+A to add new repetitions to a list.)
- 3. To move to the next dialog after entering all the answers in the list, click **▶Next** in the navigation bar. (You can also click on the next dialog icon in the interview outline, or press **Page Down** or **Alt+N**.)

When entering a list of answers within another list of answers, you can press **Alt+Left Arrow** to return to the parent dialog. (You can also choose **Parent Dialog** from the **Navigate** menu.)

To enter answers in a repeated spreadsheet

- 1. At an answer-gathering dialog that includes a repeated dialog formatted as a spreadsheet, click in the first spreadsheet cell and enter your answer.
- 2. Press **Tab** to move to the next cell in the row (or use the mouse to click in the next cell) and enter your answer.
- 3. Repeat this process for each answer.

To work with repeated answers

- 1. Answer questions in the repeated dialog.
- 2. Select the record with which you want to work. (If you are editing records in a dialog series, select the dialog icon in the interview outline. If you are editing records in a spreadsheet, click on the row you want to edit.)
- 3. Use the options described in the following table to complete additional tasks:

То	Do This
Remove a specific dialog from the list of answers	Select the dialog you want to delete and either click the P*Delete Repetition button, or choose Delete Repetition from the Edit menu.
Insert a new dialog between existing dialogs	Either click the Head Insert Repetition button, or choose Insert Repetition from the Edit menu.
Move an existing dialog up in the list of answers	Select the dialog you want to move and click the [∃] ⁺ Move Repetition Up button (located on the Edit toolbar).
Move an existing dialog down in the list of answers	Select the dialog you want to move and click the ∃+Move Repetition Down button (located on the Edit toolbar).

Sort the contents of a	Click in the column on which you want to sort and click the $\frac{1}{2}$ Sort
spreadsheet dialog in	button (located on the Edit toolbar). The Sort Spreadsheet dialog box
alphanumeric order	appears where you can specify your sorting preferences. If the template
	developer has specified any sorting options, answers will be merged in
	the document using those sort options. Otherwise the sort order you
	specify here will be used in the assembled document.
Edit the contents of	Click in the row you want to edit and either click the 🛁 Edit Row
the row in a regular	button, or right-click and choose Edit Row from the shortcut menu.
dialog	

Use Inserted Dialogs During Assembly

In some templates, one dialog may contain other dialogs. For example, a dialog may ask for information about a client; then, if the client is married, an inserted dialog may ask for information about the spouse.

To use an inserted dialog

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. Navigate to a dialog that contains an inserted dialog.
- 3. Select the inserted dialog and provide information for the answer fields, based on the following information:
 - If the dialogs are not grouped: For each inserted dialog, HotDocs displays an icon in the dialog pane and in the interview outline. If you don't answer questions in the inserted dialogs, HotDocs counts them as unanswered questions and warns you about them.
 - If the dialogs are grouped for single selection: HotDocs adds an option button next to each inserted dialog icon. Until you select one of the option buttons or click the child dialog icon or title, no icon for the inserted dialog appears in the interview outline. If you don't select an option button and don't answer any questions in the inserted dialogs, HotDocs doesn't count them as unanswered questions.
 - If the dialogs are grouped for multiple selection: HotDocs adds a check box next to each inserted dialog icon. Until you select the check boxes or click the child dialog icon or title, no inserted dialogs appear in the interview outline. If you don't select any check boxes and don't answer any questions in the inserted dialogs, HotDocs doesn't count them as unanswered questions.

You can hide all inserted dialogs in the interview outline by selecting **Collapse All** (**View** menu) or by pressing **Alt+-**.

HotDocs updates the interview as you enter answers in a dialog. If you experience a significant delay when moving from one answer to the next, you can have HotDocs update the interview less frequently. (See Update Your Interview Outline and Document.)

View Individual Rows in a Spreadsheet

Some documents may require a list of information. To generate a list, template providers create a repeated dialog, which may be formatted as a spreadsheet. This format may be awkward to work with when there are large numbers of questions (or columns)—the spreadsheet may extend beyond the viewable area. However, to see all the questions at once, you can view one row of the spreadsheet in dialog format.

To answer a single set of answers in a spreadsheet

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. At the assembly window, navigate to a repeated dialog that is formatted as a spreadsheet.
- 3. At the spreadsheet, click *i* **Edit Row**. A pop-up interview appears. (You can also right-click and choose **Edit Row** from the shortcut menu.)
- 4. Type answers in the answer fields, then click **Next**. The answers are saved and a new set of answer fields appears.
- 5. After providing all the necessary answers, close the pop-up interview by clicking **Finish**.

Select Clauses During an Interview

A clause is a section of text, usually a commonly used paragraph, and a clause library is a collection of clauses. You can select clauses from a clause library and use them to assemble text documents. (See Introduction: Use Clauses and Clause Libraries.)

In HotDocs, there are two places where you may view a clause library and select clauses for inclusion in a document—at the template library or during the interview for a text template:

• When a clause library is referenced in a template library, you assemble the clause library and then choose where the assembled clause text is merged. You can choose to send the assembled clauses to a new, empty document; copy the assembled text to the Windows Clipboard (so you can paste it into an existing document); or paste the assembled text directly into an open word

processor document. (See Assemble a Clause Library from the Template Library for instructions on using this method.)

• When a clause library is asked during the interview of a text template, the clause text is automatically inserted into that assembled document. (Instructions for this method are included below.)

To select clauses from an inserted clause library

- 1. At the HotDocs library window, select a template that contains an inserted clause library and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. Navigate to the dialog where the clause library is located. The clause library appears in the dialog pane.
- 3. Optionally, perform either of the following tasks:
 - Select the **Find** check box to find all of the clauses with specific text in their title or description. In the text field, type the text for which you are searching.
 - Click QPreview to view the clause text in a pop-up dialog box. To return to the clause library, click the X in the upper-right corner of the pop-up dialog box.
- From the Available list, select each clause you want to use and click Select to add it to the Selected list. To remove a clause, select it, then click Remove. (Press Ctrl or Shift to select multiple items.)
- 5. Optionally, reorder the clauses in the **Selected** list by selecting and dragging a clause to the desired position. The clauses will be inserted into the document in this order.
- 6. When all the necessary clauses are in the **Selected** list, click **Next** on the navigation bar.
- 7. Answer any questions in the interview.

Answer Date Questions

To make it easier to answer Date variables, HotDocs provides three different ways to enter a date:

- Type the letter **t**, then move to the next answer field. HotDocs converts the letter to the current date.
- Type a date using a common format, such as **MM/DD/YYYY**. HotDocs converts the format to match the format specified for the template.
- Click the Calendar button to access the Calendar dialog box. Then use the Next or Previous arrows to find the desired month and year. Select a day, and click OK.

HotDocs requires a four-digit year for Date variables. However, you can enter a two-digit year, and then let HotDocs convert it to a four-digit year. You can control which years are converted to

1900-century dates, and which are converted to 2000-century dates. See Control How HotDocs Handles Two-Digit and Four-Digit Years.

The **Calendar** button is not available when a Date variable is in a dialog repeated as a spreadsheet. However, if you click **Edit Row**, the pop-up interview will show the calendar button.

HotDocs may format dates in the interview different from the way they will appear in the assembled document. The appearance in the interview is controlled by a setting in the HotDocs Options dialog box. (See Change the Way Dates Appear in Answer Fields.) When a date appears in the document, however, it is formatted according to the format assigned to that variable.

HotDocs updates the interview as you enter answers in a dialog. If you experience a significant delay when moving from one answer to the next, you can have HotDocs update the interview less frequently. (See Update Your Interview Outline and Document.)

View an Answer's Variable Name

Sometimes as you complete an interview, you may need to ask the template provider about a specific question asked in the interview. Being able to identify the question by variable name may help the template provider better understand the kind of information you need. You can have HotDocs show you the name of the variable.

To view the variable name for a question in the interview

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. Navigate to the dialog that contains the question.
- 3. Place your cursor in the answer field, right-click, and choose **Variable Name** from the shortcut menu. A message box appears, showing you the variable name.

Use the End of Interview Dialog

The **End of Interview** dialog appears after the last dialog in each interview. At this dialog, you can view a report of unanswered questions, as well as access commands for working with the assembled document.

To work with assembled documents at the End of Interview dialog

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After completing the interview, the **End of Interview** dialog appears. The following table describes the commands available here:

То	Do This
Return to the first unanswered question in the interview	Click H Go to the first unanswered question in the interview.
Send the assembled text document to the word processor so you can view it in its final format	Click Send the assembled document to the word processor. You can choose the default word processor to which assembled documents are always sent, or you can let HotDocs choose the word processor each time based on the file type. See Change Your Default Word Processor.
View the assembled form document	Click the View the assembled form document at the Form Document tab button. (See View the Assembled Form Document.)
Insert the assembled document into an open word processor document	Click Click Paste the assembled document into the open word processor document. This command will paste the document text into a document already open in your word processor. If there is not a document already open, HotDocs will create a new document.
Copy the document text	Click Copy the assembled document to the Clipboard.
available when assembling Microsoft Word documents)	For DOCX templates the assembled document is copies in plain text format.
Save a copy of the assembled document	Click Save the assembled document in a file .
Save an assembled text or form document as a PDF file	Click Save the assembled document as a PDF.
	For complete instructions on saving assembled documents as PDF files, see Save Assembled Text Document as PDF.
Close the assembly window without saving a copy of the assembled document	Click Close this window without saving the assembled document.

Choose which buttons are displayed on this dialog. *Click* Choose which buttons are displayed on this dialog. *Interview* dialog

When you assemble multiple documents, the *End of Interview* dialog also notifies you that other assemblies are ready to begin.

To keep the *End of Interview* dialog from appearing, clear **End of Interview Dialog** at the **View** menu of the assembly window. Then, when you click **Next** at the last dialog in the interview, HotDocs will either send the assembled document to the word processor or HotDocs Filler, or it will display the **Document** tab of the assembly window. To specify which action HotDocs performs, see **Control What Happens When You Finish an Interview**.

You can choose which options appear in the *End of Interview* dialog. See Customize the End of Interview Dialog.

Spell Check Answers

You can spell check text answers you have entered at any time during an interview.

To spell check your answers

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Check Spelling** button. If HotDocs finds any unrecognized words, the **Check Spelling** dialog box appears. In the **Answer being checked** field, HotDocs shows a section of the text that contains the misspelled word, which is highlighted. Options for working with the misspelled word appear below it.
- 3. Make your selection, based on the options described in the following table:

То	Do This
Ignore the current instance of the word and continue spell checking	Click Ignore .
Ignore all instances of the word and continue spell checking	Click Ignore All .
Correct only the current instance of the word and continue spell checking	Select an existing replacement from the Change to list (or type the replacement in the Change to field) and click Change .

Correct all instances of the misspelled word and continue spell checking	Select an existing replacement from the Change to list (or type the replacement in the Change to field) and click Change All .
Add the word to your personal dictionary so that the spelling checker will not question the word again	Click Add .
Display additional spelling alternatives for the unrecognized word	Click Suggest . A deeper search takes longer but produces better possible replacements. If the button is unavailable, HotDocs is searching at the deepest level.
Change your spell checking options, such as which words the spelling checker looks at and how it determines whether a word is a possible replacement	Click Options . (You can also change your options at the HotDocs Options dialog box. See Change Your Spell Checking Options.)

Working with Assembled Text Documents

Preview the Assembled Text Document

After assembling a text document, you can print the document, send it as an e-mail attachment, or save it as a word processor file. However, once you perform any of these actions, the document is no longer associated with HotDocs. That means any changes you make to the text of the document will not be updated in the answer file or template. Because of this, it is helpful to preview the assembled document in the assembly window and make sure the information in the document is accurate before you perform any of the aforementioned tasks.

While the information in the **Document Preview** tab is shown accurately, the document may not be formatted correctly, particularly in WordPerfect documents. To view the actual formatting, send the document to the word processor. (See Send the Assembled Text Document to the Word Processor.)

Some template providers may not allow you to edit answers or text at the **Document Preview** tab. If not, these commands will be disabled.

To preview an assembled text document

- 1. At the HotDocs library window, select a template and begin the assembly. (See Assemble a Text or Form Document.)
- After providing the required answers, click the **Document Preview** tab. The assembly window changes to show a preview of the document. By default, answers that have not been entered are marked with asterisks (for example, ****Employee Name****). You can choose a different unanswered format, however. See Format Unanswered Variables in a Document for details.
- 3. (Word users) Optionally, to edit an answer or document text, click the **Highlight Fields** button, then double-click on an answer field or section of editable text. (See Edit Answers at the Document Preview Tab and Edit Document Text at the Document Preview Tab for details.)
- 4. Once you have verified that the document is correct, use any of the following commands to work with the assembled document:

То	Do This
Save the assembled text document to disk	Click the Save Document button. (See Save an Assembled Text or Form Document.)
Send the assembled text document to the word processor so you can see it in its final format	Click the Send Document button. (See Send the Assembled Text Document to the Word Processor.)
Print the assembled text document	Click the Print Document button. (See Print an Assembled Text Document or Print an Assembled Form Document.)
Close the assembly window	Choose Close (File menu)

If an assembled text document contains any cross references, including a table of contents or index, you may need to update the references once you send the document to the word processor. To do this, select all of the document text (press **Ctrl+A**) and then press the **F9** key.

Edit Answers at the Document Preview Tab

The ability to edit answers at the **Document Preview** tab is available to Microsoft Word users only. Additionally, this option is available only if the template provider has allowed it.

If you are viewing the **Document Preview** tab, you can edit answers you've entered. Additionally, you can move from an answer in the document to its question in the interview and vice versa. Seeing answers in the context of the document helps you pinpoint errors and makes it easy for you to correct them without having to go back to the interview.

To help you review answers in the document, you can use the Document Navigation Bar. Just like the Interview Navigation Bar, this toolbar can help you move between answers, unanswered fields, and matching answers.

Some template providers may prohibit you from changing answers at the **Document Preview** tab. If so, the buttons and commands that control this feature will be disabled.

To change answers while previewing the document

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Document Preview** tab. The assembly window changes to show a preview of the document.
- 3. Click the **Highlight Fields** button. This assigns colors to both answered questions and unanswered questions in the document. (You can change the default colors assigned to answered fields as well as unanswered fields. See Set Properties for Viewing Answers and Editable Text in the Assembled Document.)
- 4. Double-click on an answer field. A pop-up interview appears, showing the dialog in which the question is used.
- 5. Change your answer and click **Next** or **Finish**. The pop-up interview closes and the answer is updated in the document.
- 6. Use any of the following commands to help you edit answers in the document:

То	Do This
Jump to the place in the interview where the question is asked	Place your cursor in the answer field and click the Go to Answer in Interview button. You can also right-click on the answer field and select Go to Answer in Interview from the menu. While viewing the Interview tab, you can go to the place in the document where the answer is merged. To do this, place your cursor in the answer field and click the Go to Answer in Document button. Because of scripting in the template, sometimes you will not be able to move between the Interview and Document tabs in this manner. In the Document tab, if you place your curser in the answer field of an answer generated from a computation, the Go to Answer in Interview button in the tool bar will be greyed out. If you right-click in the answer field the menu will show Computed Value and you will be unable to select it. Similarly, in the Interview tab, if you place your curser in the answer field of an answer that is not directly used in the
	message.
Move between answers in the document	Click Next or Previous to move to the next or previous answer in the document.

Click the **Next Unanswered** or **Previous Unanswered** button to move to the next unanswered or previous unanswered question in the document.

Press **Ctrl+M** or **Ctrl+Shift+M** to move between matching answers in the document (for example, those answers in the document that are exact matches). (You can also press **Shift** as you click the **Next** or **Previous** button.)

Click **Hend** or **Heginning** to go to end of the document or the beginning of the document.

If you experience a delay when changing answers in your document, click the **Document Instant Update** button (so it isn't pressed in). Then, when you change an answer, HotDocs will just update the field you are editing (and any other fields that merge the exact same answer). Be aware, however, that turning **Dinstant Update** off may cause some of the information in the document to be out of date if you change answers. (You can tell when the document is out of date by the appearance of the **Dinstant Update** button. To update the document, click the button again. (The document will automatically update if you attempt to save it, print it, or send it to the word processor.)

Once your cursor is in an answer field, you can press **Enter** and the pop-up interview will appear.

If the template provider has allowed it, you can also edit the text of the document. See Edit Document Text at the Document Preview Tab for details.

At a Glance: The Document Text Editor

🛃 Document Text Editor	×
🗛 🖾 Save & Close 🖉 🛃 🖛 🐃 🎸 🗈 🛍 🛍 🖓 🏰 🛣 🤹 🖉 ¶ 🦃 ⊘ Cancel 🔞	
Normal → Arial → 10 → B I U E = = = #	
8	<u>^</u>
B	
	-
I € F	

The **Document Text Editor** will appear when you click on editable document text while on the **Document Preview** tab of the HotDocs Interview.

The window is split into two parts, at the top is the Toolbar A and at the bottom is the workspace B where you can type to edit the text.

The Toolbar A itself is split into two parts. The lower part has standard word processor tools such as font, styles and justification. The upper part of the toolbar has more specific HotDocs buttons including:

- Save and Close: Saves your changes, closes the Document Text Editor, and merges the new text in the assembled document.
- **Restore Original Text:** Discards any changes you've made to the text and restores the original document text.
- **Print:** Prints a copy of the text you are editing
- **• Undo:** Removes any changes you have made to the script.
- **Redo:** Reapplies any changes you have made to the script.
- **Cut:** Removes the selected text and copies it to the Clipboard.
- **Copy:** Copies the selected text to the Clipboard.
- **Paste:** Pastes the selected text on the Clipboard at the cursor position in the script.
- **A** Find: Displays the Find dialog box where you can specify the word or text string for which you are searching.
- **B** Find Next: Finds the next instance of the word or text string for which you are searching.
- Label{eq:Find and Replace: Searches for a specific word or text string and replaces it with word or text string you specify.
- *Ruler Guide:* Show or hide the ruler guide below the toolbar.

- **¶Paragraph Marks:** Show or hide the paragraph marks in the editable text. HotDocs always adds an extra paragraph mark to the end of editable text which cannot be deleted.
- **Spell Check:** Checks the spelling within the Document Text Editor only.
- **Cancel:** Closes the Document Text Editor without saving any changes.
- **Whelp:** Opens the relevant page of the HotDocs Help File.

To learn more about using the document text editor follow the link below:

• Edit Document Text at the Document Preview Tab

Edit Document Text at the Document Preview Tab

The ability to edit the text of an assembled document is available to Microsoft Word users only. Additionally, this option is available only if the template provider has allowed it. (You can tell if the developer has made text editable by the color of the document text. By default, editable text is green.)

After you finish answering questions in an interview and before you send the document to the word processor or save it, you should review it. As you do, you may see that you need to change the language of a particular paragraph, or you may need to change a word or phrase to fit your particular needs. If the template developer has allowed it, you can edit the document text while viewing the **Document Preview** tab. Changes you make can be saved in an answer file so that if you reassemble the document, those changes will be reapplied. (You can always send the document to the word processor and make the changes there. However, those changes are specific to the word processor and will not be saved to HotDocs.)

When you edit the text of the assembled document, the text appears in a pop-up window called the **Document Text Editor**. This editor provides many of the text-editing tools you need, including font options, spell checking, printing, and copying and pasting. As you edit, you can add new text or remove existing text. You can also change the formatting of text.

You cannot edit variable (or answer) fields at the **Document Text Editor**. To edit answers, return to the assembly window and make the change at the **Interview** tab.

To edit the text of the document

- 1. At the HotDocs library window, select a Microsoft Word template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Document Preview** tab. The assembly window changes to show a preview of the document.
- 3. Click the \checkmark Highlight Fields button. Text in the document that can be edited changes color.

- 4. Place your cursor in an editable section of text and click the **≝Edit Document Text** button (or double-click on the text). The **Document Text Editor** appears. (If there are multiple sections of text you can edit, HotDocs may ask you which section you want to edit.)
- 5. Edit the text using the commands in the **Document Text Editor** toolbar. (See Document Text Editor Toolbar for a description of each command.)
- 6. When you are finished making your changes, click Save and Close. The editor closes and your changes are merged in the assembled document.

When you are finished reviewing the document, you can save your answers and the changes you made to the document are saved to the answer file. Then, if you reassemble the document using that answer file, the document will update with the changes you made.

To move between sections of editable text, click the \bigvee Next Editable Text or the \bigwedge Previous Editable Text buttons in the navigation bar.

You can choose specific colors for marking editable and edited text in the document. See Set Properties for Viewing Answers and Editable Text in the Assembled Document.

When viewing the document in the **Document Text Editor**, if you click the **¶Show Codes** button, you will see an extra paragraph mark at the end of the document. This paragraph mark is merged by the text editor and will be deleted when you save your changes and the text is inserted back into the document. Because of this, do not modify the paragraph mark.

View an Assembled Document in Markup View

If you are not a template developer, please click here for a brief explanation of template development terms. Understanding these terms will help you understand how the assembled document is marked up.

As you assemble Microsoft Word documents, there may be times when you want to assemble only a partial document and then submit it to an attorney or peer for revisions or corrections. To make the document easier for a non-HotDocs user to review, you can change the formatting of the document to Markup View and then provide the reviewer a copy of it.

When displaying markup, HotDocs replaces unanswered fields and instructions in the document with markup, which includes markers (such as brackets) and variable names. How the resulting markup fields look depend on the type of fields they are:

• **Answer fields:** By default, answer fields are marked using brackets. If the template developer has assigned a comment to the variable, the comment text may be used as the field name. If no comment is used, HotDocs will use either the title or the name the template developer assigned to the variable.

- **Conditional and repeated text blocks:** By default, conditional and repeated text blocks are marked using brackets. If a comment is assigned to these text blocks, the comment can be merged in the field label. If no comment is assigned, HotDocs will simply merge the field markers around the text block. The text block can be marked using an annotated footnote or endnote, depending on your preferences.
- **Insertion fields:** HotDocs will merge the file name of the template being inserted.

Once you assemble the document using as many answers as needed, you can switch to Markup View and save a copy of the document. You can then provide a copy of it to the reviewer.

To view a document in Markup View

- 1. Define the Markup View options you want to use in the document.
- 2. Assemble the Word document for which you want to generate markup.
- 3. Answer only the questions whose answers you want to appear in the assembled document. Leave all other questions unanswered. (These unanswered questions are the fields that will be marked using the scheme you define in step 1.)
- 4. Click on the **Document Preview** tab to switch to the preview pane.
- 5. Choose **Markup View** (**View** menu). The document changes to show the marked up document.
- 6. If you send the document to the word processor it will appear in markup view.

By default, merge fields in the document appear in color—red for unanswered questions and blue for answered questions. Sometimes, however, if an answer field is within a text block that is conditional or repeated, it appears (and remains) blue until the condition is met or the REPEAT instruction is answered. Editable text likewise appears using the colors you've defined at HotDocs Options.

Additionally, merge fields in the document are italicized by default. If you've used international characters in your answers, the italicized version of the font used in the document may not display the foreign characters correctly. If this is the case, clear the **Italic** attribute for Markup View.

If the template developer has used merge fields in question prompts, those merge fields can be displayed as markup fields. Click the **View** menu, and then click **Markup View** while in the interview tab. Additionally, you can switch to Markup View while assembling form documents.

To view a non-marked up version of the assembled document again, clear **Markup View** (View menu).

Send the Assembled Text Document to the Word Processor

Once you complete an interview, you often want to view the document in its final format. For text documents, this must be done in the word processor.

To send the assembled text document to the word processor

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Send Document to the Word Processor** button. The word processor appears, showing the assembled document.
- 3. Using the word processor's tools, modify the document. Be aware, however, that any changes you make to answers in the document will not be reflected in the answer file.

If you have more than one word processor installed, you can choose which word processor will be used for viewing assembled documents. See Change Your Default Word Processor for details.

You can preview a text document at the **Document Preview** tab. While this preview shows the information in the document correctly, it may not be formatted correctly—particularly in WordPerfect documents.

If the template provider has allowed it, you may be able to edit the text of the document while still viewing it in the assembly window. For details, see Edit Document Text at the Document Preview Tab (Word users only).

If you are assembling a Plain Text Templates, the document will be sent to your default text editor instead of word processor.

Open a Saved Text Document for Editing

After assembling and saving a document, you can later edit the document using the appropriate program:

- Text documents are saved as word processor files (.DOCX, .DOC, .WPD, or .RTF). To edit these files, you must use a word processor.
- Form documents are saved as HotDocs form files (.HFD or .HPD). To edit these files, you must use HotDocs Filler. See Edit a Saved Form Document for instructions on completing these actions.

To open a saved text document using Windows Explorer

• Using Windows Explorer, locate the document file and double-click it. Your word processor appears, showing the saved document.

To open a saved text document using the word processor
• Using your word processor program's commands, locate and open the desired document file.

When an assembled document is saved, the file is separated from the template file, component file, and answer file that created it. This means that changes you make to a saved document are not reflected in the answer file used to assemble that document. Finally, if you do make changes to the document, you must identify and change each instance of the answer manually, including re-evaluating any computations or conditions affected by changed answers.

Print an Assembled Text Document

Many projects require you to print copies of assembled documents.

To print an assembled text document

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Document Preview** tab. The assembly window changes to show a preview of the document.
- 3. Click the **Print Document** button. The document is printed at the printer you specify.

Save an Assembled Text or Form Document

Some projects may require you to save an electronic copy of every document. Also, you may need to move the document to a disk or laptop, so you can take an editable, printable version of the document to another location.

When you save a text or form document, the file is separated from the answer file, so any changes made to a saved document are not saved to the answer file.

To save an assembled text or form document

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Save Document** button in the toolbar or at the **End of Interview** dialog. The **Save As** dialog box appears.
- 3. Navigate to the folder where you want the document to be saved.

4. In the **File name** field, enter a name for the document and click **Save**.

Save Assembled Text Document as PDF

With HotDocs Developer, you can use the PDF printer driver to save assembled text and form documents as PDF documents. Creating PDF files allows you to take advantage of features available in Adobe Acrobat.

To save a document as PDF

- 1. At the HotDocs library window, select a text template and click Resemble. (See Assemble a Text or Form Document.)
- 2. Answer questions in each dialog.
- 3. After answering all of the questions, choose **Save Document As** (**File** menu). The **Save As** dialog box appears.
- 4. Specify the file name and folder path for the file.
- 5. At the Save as drop-down list, make sure PDF File (*.pdf) is selected.
- 6. Optionally, click **Security** to assign any security options you need to the document. (See Assign Security Options to a PDF Document.)
- 7. Click Save. The Convert to PDF dialog box appears.
- 8. Optionally, you can add information about the document's title, subject, author, or search keywords. Otherwise, click **OK**. The assembled document is saved in PDF format.

Information you type in the **Convert to PDF** dialog box is saved as metadata embedded in the PDF-based file. HotDocs does not use this information, but other programs that work with PDF files may access this metadata. (For instance, if you open a PDF or PDF-based form template in Adobe Acrobat, you can select **Document Properties** (**File** menu) to see the values in these fields.) For form templates, the values input here will also be set in any PDF documents assembled with that template.

Save Assembled Form Document as PDF

With HotDocs, you can use the PDF printer driver to save assembled PDF form documents as PDF documents. Creating PDF files allows you to take advantage of features available in Adobe Acrobat.

To save assembled PDF forms as PDF

- 1. At the HotDocs library window, select a PDF template and click **Assemble**. (See Assemble a Text or Form Document.)
- 2. At the **End of Interview** dialog, click **Save the assembled document as a PDF**. The **Save As** dialog box appears.
- 3. Specify the file path and name for the PDF file, and select **PDF File (*.pdf)** from the **Save as type** drop-down list.
- 4. Optionally, click **Security** to assign any security options you need to the document. (See Assign Security Options to a PDF Document.)
- 5. Click **Save**. The **Convert to PDF** dialog box appears.
- 6. Optionally, you can add information about the document's title, subject, author, or search keywords. Otherwise, click **OK**. The assembled document is saved in PDF format.

If the file you're using to create a PDF document is a form document, the answer fields and other HotDocs features are removed. Only the current answers are shown as static text on the PDF document. If you may need to edit the answers later, you should first save the file as a PDF-based form document (.HPD), then save another copy of the file as a PDF.

Information you type in the **Convert to PDF** dialog box is saved as metadata embedded in the PDF-based file. HotDocs does not use this information, but other programs that work with PDF files may access this metadata. (For instance, if you open a PDF or PDF-based form template in Adobe Acrobat, you can select **Document Properties** (**File** menu) to see the values in these fields.) For form templates, the values input here will also be set in any PDF documents assembled with that template.

At a Glance: The PDF Security Settings Dialog Box

PDF Security Settings	? 💌	
ecurity Level None 40-bit RC4 encryption 128-bit RC4 encryption	(compatible with Acrobat 3.x and later) n (compatible with Acrobat 5.x and later)	
Basswords		
Master password (red	uired to change security settings)	
	Confirm:	
User password (required to open document)		
	Confirm:	
C Printing (Content Protection	
Fully allowed	✓ Allow content copying and extraction	
Cow resolution only	\checkmark Allow access for visually impaired	
Not allowed		
Content Editing Allow content editing Allow page-level editin Allow annotations	g (insert, delete, rotate	
[✓] Allow form filling		
	OK Cancel	

If you select **Save Document As...** from the **File** menu while assembling a template, then you can view the **PDF Security Settings** dialog box by choosing **PDF File(*.pdf)** from the **Save as Type** drop-down list and clicking the **Security** button.

In the first set of options A you can decide on the Security Level you would like from the following options:

- None: Assigns no security options to the PDF document you are creating.
- **40-bit RC4 Encryption:** Allows you to assign minimum security options to the document, including disabling the ability to print the document, changing the contents of the document, copying or extracting the contents of the document, and adding or changing comments in the document. While security is less, the document is compatible with versions of Acrobat back to version 3.
- **128-bit RC4 Encryption:** Allows you to assign more complicated security options to the document. This level of security, however, makes it so the document can't be viewed in any version earlier than Acrobat 5.

In the second set of options **B** you can set Master and User passwords for the file. You can set the Master password by ticking the check box and entering your chosen password in both the Password and Confirm fields below. The file then requires a password when editing the contents of the document. You can set the User password by ticking the check box and entering your chosen password in both the Password and Confirm fields below. The file then requires the intended recipient to enter a password when opening the document. Make sure you communicate this password to the intended recipient.

In the third set of options **C** you can set the printing options. **Fully allowed** lets the user to print a copy of the document, **Low resolution only** lets the user to print only a low-resolution copy of the document, and **Not allowed** keeps the user from printing a copy of the document

In the fourth set of options **D** you can set the content protection options. Check the first box to allow the intended recipient to copy content from the document and check the second box to allow visually impaired recipients greater access to the document.

In the final set of options **b** you can set the Document Editing options by ticking the check boxes:

- **Allow content editing**: Allows the intended recipient to edit the content of the document, including making changes to the text.
- **Allow page-level editing:** Allows you or the intended recipient to make page-level edits in the document.
- Allow annotations: Allows the intended recipient to annotate the document.
- Allow form filling: Allows the intended recipient to fill form questions in the document.

To learn more about setting options for PDF Security follow the link below:

• Assign Security Options to a PDF Document

Assign Security Options to a PDF Document

When creating PDF documents, you may want to protect the content of the document. You can do this by assigning security options to the document.

To assign security options at the HotDocs Save As dialog box

- 1. Assemble a document. (See Assemble a Text or Form Document.)
- 2. Choose Save Document As (HotDocs File menu). The Save As dialog box appears.
- 3. Click the Save as type drop-down button and choose PDF File (*.pdf).
- 4. Click the Security button. The PDF Security Settings dialog box appears.

- 5. Make your selection, based on the following information (for additional information about each of these options, see the Adobe Help file):
 - Select **None** to leave the document unprotected. No security options will be applied and the user can edit the document (using Adobe Acrobat Professional).
 - Select **40-bit RC4 encryption** to apply less-stringent security options to the document (and make the document compatible with all versions of Adobe starting Adobe 3.x). (Selecting this option affects which options are available in the rest of the dialog box.)
 - Select **128-bit RC4 encryption** to apply more stringent security options to the document (and restrict use of the document to Acrobat 5.x and later users). (Selecting this option affects which options are available in the rest of the dialog box.)
 - Select **Master password** and enter (and confirm) the required password to require the user to enter a password when changing security options for the document.
 - Select **User password** and enter (and confirm) the required password, require the user to enter a password when attempting to open the document.
 - Select one of the printing options: Fully allowed, Low resolution only, and Not allowed to allow varying levels of printing.
 - Clear **Allow content copying and extraction** to keep users from copying the text in the document and pasting it into other applications.
 - Select **Allow access for visually impaired** to allow the document to be read out loud by Acrobat.
 - Select **Allow content editing** to let users can modify the pages of the document, including adding or removing pages, or rotating pages to let users make changes to the document.
 - Select **Allow page-level editing (insert, delete, rotate)** to let users make changes to page properties of the document (such as inserting pages).
 - Select **Allow annotations** to let users make comments or other notes in the document.
 - Select **Allow form filling** to let users enter information in Adobe fields on the form.

Working with Assembled Form Documents

Introduction: Assemble a Form Document

In most cases, you assemble a form document exactly as you assemble a text document. (See Assemble a Text or Form Document.) However, HotDocs form documents provide some additional capabilities. For example, you can enter answers directly in the form document, moving from field to field. You can also print a blank copy of the form that can be completed by hand. Finally, you can create and modify an addendum to save answers that require more space than the form provides.

At a Glance: HotDocs Filler

HotDocs Filler - [Form1]		- • ×
Eile Edit View Field Tools Window Help Menu Bar		_ & ×
6] 📂 🗟 🚄 🔺 📘 💺 🖑 🔍 🖽 📜 🖪 🗉 🕢	Toolbar	
	·	<u>^</u>
Workspace		=
Hobble Creek Publishing		
GROUP TERM LIFE INSURANC		TION
Please print all information in black ink.		non
Section 1: Employee Information		
Name (Last, First, Middle Initial)		Gender
Street Address	City	State
Street Audress	City	State
Email Address		
Social Security Number	Marital Status	
	Single Married	Separate
Section 2: Coverage Information		Ŧ
		4
		NUM

When you open HotDocs Filler to view assembled form documents you will see the window laid out with one main Workspace, a Toolbar and above that a Menu Bar. Using HotDocs Filler you can edit the answers and the form, save and print the document.

From the Toolbar you can access the following buttons:

- **d**HotDocs Library: Opens the HotDocs library (if it's not already open) and brings it to the front.
- **Derived** Opens a form template file for editing, or opens a saved form document for printing or editing.
- **Save:** Saves the current form. If the form has never been saved, HotDocs will ask for the path and file name to use.

- **difference** Print: Prints a copy of the form.
- Field Properties: Displays the Field Properties dialog box during direct-fill assembly of form documents. This allows you to modify the appearance of answers in that field.
- AAnswer Font: Changes the font for the selected field or fields.
- **Fill Tool:** Allows you to place your cursor in a form's answer field and type an answer.
- **Select Tool:** Either creates a new field or selects an existing field during direct-fill assembly of a form document.
- **Scroll Tool:** Scrolls form documents up and down, and side to side during direct-fill assembly.
- **QZoom Tool:** Changes the view of the form document during direct-fill assembly, increasing the magnification (or decreasing it when used in combination with the Shift or Ctrl keys).
- **Fit Page Width:** Displays the entire width of the current form page in the window.
- **I** Fit Page Height: Displays the entire length of the current form page in the window.
- **Show Fields:** Shows or hides the field backgrounds.
- **Show Thumbnails:** Shows or hides a thumbnail view of each page in the form at the left of the window.
- **Previous Page:** Moves to the page just before the current page in the form.
- **Next Page:** Moves to the page just after the current page in the form.
- In HotDocs Options: Opens the HotDocs Options dialog box where you can specify several preferences that control how HotDocs works
- **Whelp:** Opens the relevant page of the HotDocs Help File.

In the Menu Bar you can access further options not available on the Toolbar.

To learn more about editing assembled documents using HotDocs Filler follow the link below:

• Edit a Saved Form Document

Enter Answers Directly at the Form Document Tab

When assembling text documents, you answer questions in the interview and those answers are merged into the document. When assembling form documents, however, you can either enter your answers in the interview or you can enter your answers directly on the form document. This latter method of assembly is called direct-fill assembly. In some cases, the template provider designs templates so that the only method for assembly is by direct-fill.

To enter your answers directly into the fields

- 1. At the HotDocs library window, select a form template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. If you're not viewing it, click the **Form Document** tab. The assembly window changes to show the form document.
- 3. Click on a field and enter an answer.
- 4. Press **Tab** to move to the next field, or use the mouse to click on another field.

As you tab between fields, HotDocs will skip any conditioned fields that have been resolved in such a way that it isn't necessary to answer the field.

5. Optionally, you can perform any of the following tasks:

То	Do This
Create a new field on the document	Click the Select Tool button and draw the field. (See Create a Form Field.) To enter text in the field click on the J Fill Tool and then click in the field.
Add additional answers to a table	If the template developer has included it, click the Hable Wizard button next to the field.
Answer questions that compute the answer for a field or control the inclusion of other answers	If the template developer has included it, click the Answer Wizard button next to the field.
Show or hide the colored fields	Click the Show Fields button.
Quickly browse through the pages of a form document	Click the Show Thumbnails button. This displays small images of each page in the document in the left margin of the form view. You can click one of these images and go to that page immediately.
Jump to a specific page of the assembled document	Choose Go To (Edit menu) and enter the page number of the page you want to view.
Move through the answers in a document	Place your cursor in a field and click one of the following navigation buttons: (If the Form Navigation Bar isn't showing, choose it at the View menu.)
	 Clicking Next or Previous moves you to the next or previous answer in the document.
	 Clicking the Next Unanswered or Previous Unanswered button moves you to the next unanswered question in the document
	 Pressing Ctrl+M or Shift+Ctrl+M moves between matching answers in the document (for example,



those answers in the document that are exact matches). (You can also press **Shift** as you click the **Next** or **Previous** button.)

 Clicking ▶ End or ▶ Beginning to go to the end of the document or the beginning of the document.

To access answer wizards, table wizards, and calendars during direct-fill assembly, right-click on the field and choose the option from the shortcut menu. For example, to view the calendar for a date field, right-click on the field and choose **Calendar** from the shortcut menu.

Some answers may be too large to fit in the field. If this happens, HotDocs warns you by displaying the **Overflow Status** dialog box. See Check Form Fields for Answer Overflow. Similarly, you can specify when this warning should appear. See Set Rules for Handling Answer Overflow.

At a Glance: The Field Inactive Warning Dialog Box



When entering answers at the form document tab, you will see the **Field Inactive Warning** dialog box when you try to enter an answer into a restricted field.

Here you have three options:

- **Override:** Choosing **Override** will allow you to type any answer into the field regardless of the restrictions placed upon it, however this answer will not be saved in the answer file and will not be used in any calculations. After clicking the **Override** button you will see another warning tab explaining this.
- Wizard: If the field has an answer wizard attached to it then choosing Wizard opens that answer wizard. From there you can answer the questions necessary to properly create the answer that the restricted field requires. You can also open answer wizards by clicking on the Answer Wizard button next to the field.

• **OK:** Choosing **OK** will close the warning dialog without altering the answer in the answer field.

To learn more about entering answers at the form document tab follow the link below:

• Enter Answers Directly at the Form Document Tab

View the Assembled Form Document

Before saving the assembled form document, it is a good idea to review the document. You can do this at the **Form Document** tab of the assembly window. Doing this allows you to make changes to answers in the document and have those changes automatically updated in the answer file.

To view assembled form documents

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After answering the questions in the interview, click the **Form Document** tab. The assembly window changes to show the assembled document.
- 3. Optionally, select a field and enter a different answer. (See Enter Answers Directly at the Form Document Tab.)
- 4. Once you have finished entering answers, complete any of the following options:
 - Click the Save Document button to save the assembled form document to disk. (See Save an Assembled Text or Form Document.)
 - Click the IPrint Document button to print the assembled form document. (See Print an Assembled Form Document.)
 - Click Send Document To > Mail Recipient (File menu)to attach the assembled form document to an e-mail message. (See Attach a Text or Form Document to an E-mail Message.)
 - Click **Close** (**File** menu) to close the assembly window. HotDocs prompts you to save any changed answers.

See Introduction: Assemble a Form Document for a list of changes you can make to an assembled form document.

Assemble a Fillable PDF Document

Some form templates you assemble can be saved as fillable PDF documents. Fillable PDF documents allow users to enter answers on the form while editing the form in Adobe Acrobat or Reader.

The process of assembling a fillable PDF is nearly the same as the process for assembling a regular form document—you can enter answers in the interview or type your answers directly on the form at the **Form Document** tab. You can also choose to save the assembled document as a HotDocs form document, a regular PDF, or a fillable PDF. If you plan to save the assembled document as a fillable PDF, however, there are a few things you should understand:

- Adobe Acrobat/Reader does not support the creation of an addendum when an answer overflows its field. If a field overflows in HotDocs, Adobe may attempt to force the answer to fit in the field when you save it as a fillable PDF. If the answer doesn't fit, it will simply appear cut off. If possible, you should review the assembled document for any field overflow before sending it to Adobe so that you can minimize the chance answers won't appear correctly in the field. See Check Form Fields for Answer Overflow for details.
- When viewing the document in HotDocs, some answers may appear differently than when viewing the assembled document in Adobe Acrobat/Reader. This is because the underlying Adobe fields may have formats or patterns assigned to them that are different than what was assigned in HotDocs.
- If you create additional fields on the assembled document, any text you enter in the fields will appear as static text when you save the document as a fillable PDF.
- If you change the properties of an existing fillable field, those changes will not be honored when you save the document as a fillable PDF.

To assemble a fillable PDF document

- 1. Follow the instructions for assembling a form document in Assemble a Text or Form Document.
- 2. Complete the interview. (See Navigate Through Answer-Gathering Dialogs.)
- 3. At the Form Document tab, review your answers.
- 4. Choose **Check for Overflow** (**Tools** menu) and resolve any fields that overflow.
- 5. Click the **Save Document** button. The **Save As** dialog box appears
- 6. Click the Save as type drop-down button and select Fillable PDF Form from the list of options.
- 7. Enter a name in the File name field and click Save. The document is saved.

Once you save the document, you should review it to make sure the answers appear correctly.

When creating a fillable PDF document, form designers can enable usage rights within the document. These rights allow Adobe Reader users to save copies of the document with changes they have made to the document—including entering or changing answers they have entered in the fillable PDF. However, when a rights-enabled fillable PDF is converted to template format, the document is changed in such a way that usage rights are no longer in effect. This means that you will not be able to save changes you make when editing the assembled document in Adobe Reader.

Override an Answer in a Form

In a HotDocs form document, fields are designed to hold certain types or formats of information. However, there may be times when you need to enter a type of answer that HotDocs won't allow. You can override the field type and enter any answer you need.

For example, a field with a Number variable attached to it is designed to hold only digits. However, if the number you enter (say, \$325,000,000) is too large for the field, but you don't want to send the answer to the addendum, you can override the field and enter \$325 m.

Answers entered in overridden fields are not saved in the answer file. The original answer is saved in the answer file and used in any computations or other fields that require it.

To override an answer

- 1. At the HotDocs library window, select a form template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After entering answers, click the **Form Document** tab. The assembly window changes to show the document.
- 3. Select the field you want to override.
- 4. Click the **Override Field** button and, at the **Override Field** warning message, click **OK**. The assembly window appears again.
- 5. In the field, enter the answer you want, then move to a different field. The overridden field changes to gray.

To remove the overridden answer, select the field and click the **Override Field** button again. The current answer is removed and the saved answer reappears. Removing the override, however, may create answer overflow situations. See Check Form Fields for Answer Overflow for details.

At a Glance: The Print PDF-Based Documents Dialog Box

Print Document	3
Print this form using HotDocs Automator (prints slower, but provides more printing options) Adobe Acrobat (prints faster, but provides fewer printing options)	
OK Cancel	
<u>B</u> Use this selection in the future without asking	

When you are assembling a PDF-based template you can view the **Print PDF-based Documents** dialog box by clicking the **GPrint** button at the **Form Document** tab or by selecting **Print Document...** from the **File** menu.

From the multiple choice list A at the top of the dialog you can select which application you want to use to print the document—HotDocs or Adobe Acrobat. The following provides some guidelines for choosing an option:

When using **HotDocs** to print the document:

- You cannot print a PDF-based form using a PostScript printer driver.
- You should print within the 300 DPI (dots per inch) to 600 DPI range, as printing at higher resolutions may be problematic.
- For the most reliable results, use a PCL-based printer with at least 8 MB RAM. Other types of printers may or may not print as expected.
- You can print the document with or without the answers merged. (You cannot do this using Adobe Acrobat.)
- Printing time takes longer than printing with Adobe Acrobat. (Printing at a lower resolution may decrease the amount of time it takes to print.)

When using Adobe Acrobat to print the document:

- You can print a PDF-based form using a PostScript printer driver. (PostScript printer drivers are always recommended. Non-PostScript printer drivers may print the page off-center by 1/8 inch to 1/4 inch.)
- Documents will print with both form fields and answers—you have no option of printing one or the other.
- Printing time is faster than with the HotDocs print function.
- Both printer options give you options for scaling the content to fit in the printable area.

At the bottom of the dialog there is a check box **B** that you can select to specify the printing option you select as the default selection. HotDocs will not prompt you for this information again.

To learn more about printing options for form documents follow the links below:

- Print an Assembled Form Document
- Choose Default Form Printing Options for a Form

Print an Assembled Form Document

Many projects require you to print copies of assembled form documents. The options available for printing a form depend on which type of form you are printing. Specifically, you can print:

- HotDocs form documents (or .HFDs).
- HotDocs PDF documents (or .HPDs).

To print a form document

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, click the **Form Document** tab. The assembly window changes to show the assembled document.
- 3. Click the 🖾 **Print Document** button.
- 4. If you are printing a HotDocs form document (or .HFD file), select any of the following printing options. (If you're printing a PDF form document, skip to step #5.)
 - To print the form and answers, select **Form with Answers** in the **Output** group.
 - To print only the static text of the form, select Form Only (Blank Form) in the Output group.
 - To print the answers onto a pre-printed copy of the form, select **Answers Only (Use Preprinted Form)** in the **Output** group.
 - To change the double-sided printing options, select the desired option. Single-sided prints the document using one side of each sheet of paper. Double-Sided, Side-to-Side prints the document with the tops of both pages at the same end of the paper, like a traditional book. Double-Sided, Top-to-Bottom, prints the document with the bottom of the second page at the same end as the top of the first page, like a flip-chart.
- 5. If you are printing a PDF form document (or .HPD file), you must choose whether to print using the functionality of HotDocs or Adobe Acrobat. (See Print PDF-Based Documents Dialog Box for tips on which printing option you should choose.)
 - Choose **HotDocs**, which prints at a slower speed but provides a wider range of printing options.

• Choose Adobe Acrobat, which prints at a faster speed but provides few printing options.

Some types of printers have been known to cause problems when printing a form template or document, and should be tested before use with finished products. These printers include:

- Ink jet printers
- Brother printers
- 16-bit postscript printer drivers
- DeskJet printers
- Xerox printers
- "All-in-one" printers (printers that include copying, scanning, printing, and faxing capabilities)

For best results, it is recommended that you use an HP LaserJet with at least two megabytes of memory.

At a Glance: The Printing Properties Dialog Box

Printing Properties	
Printing options Single-sided	
Double-sided, side-to-side	
Ouble-sided, top-to-bottom	
Paper size:	
Paper source:	
OK Cancel	

While editing a form template, you can open the **Printing Properties** dialog box choosing **Printing** from the **Template Properties** list in the **File** menu.

From the multiple choice list \underline{A} at the top of the dialog you can select the from three printing options:

- Single-sided: Prints all pages on separate sheets of paper.
- **Double-sided, side-to-side:** Prints the template using both sides of the paper, so you can turn through the pages like a traditional book.

• **Double-sided, top-to-bottom:** Prints the template using both sides of the paper, so you can lift the pages up, like a flip-chart.

Below this list are two drop-down menus. From the first drop-down list **B** you can select the paper size you would like to use and from the second drop-down list **C** you can select the paper source.

To learn more about setting the printing options for a form template follow the link below:

• Choose Default Form Printing Options for a Form

Choose Default Form Printing Options for a Form

Different forms may require different printing settings. You can specify that a form should be printed single-sided, double-sided side-to-side, or double-sided top-to-bottom. You can also control what size of paper to use as well as which paper tray should feed the paper to the printer.

The option you specify is set as the default in the **Print** dialog box, but users can select a different option.

To specify form printing options

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click File > Template Properties > Printing. The Printing Properties dialog box appears.
- 3. Select the printing settings from the following options:
 - **Single-sided** prints the document using one side of each sheet of paper (default setting).
 - **Double-sided, side-to-side** prints the document double-sided, with the tops of both pages at the same end of the paper. (This option allows you to turn through the pages like a traditional book.)
 - **Double-sided, top-to-bottom** prints the document double-sided with the bottom of the second page at the same end as the top of the first page. (This option allows you to turn through the pages like a flip chart.)
 - **Paper size** specifies the dimensions of the paper used to print the document.
 - Paper source specifies what tray of paper is used to print the document.

Paper size and source settings are recommendations that will be passed on to the printer if possible. If printing via acrobat then paper size, paper source and duplex options will revert to your defaults.

You can specify the print setting for all forms you create at the **HotDocs Options** dialog box. (See Introduction: HotDocs Options.)

You can use command-line options to control the paper size and paper source. (See Paper Size and Paper Tray.)

Edit a Saved Form Document

After assembling a form document, you often need to save the document to disk. Once saved, you can reopen the document using HotDocs Filler.

Once a form document is saved to disk, it is no longer associated with the answer file. This means that changes you make to answers in the document do not affect the answer file. If you want to save changes you're making to the answers, re-assemble the form document and change your answers in the interview. Then save your answer file again.

To work with a saved form document in Filler

1. Using Windows Explorer, locate and double-click the appropriate form document. HotDocs Filler appears, showing the form document.

То	Do This
Change an answer in a	Click on a field. The field becomes active, allowing you to enter an answer.
field	You can edit answers in a saved form document, but you must re-evaluate any computations or conditions affected by changed answers. Similarly, changes you make are not reflected in the answer file used to assemble the document.
Create new fields	With the Select Tool button selected, press the left mouse button and move the mouse pointer to draw a rectangle. To type in this field, select the Fill Tool button.
	The fields you create in Filler have no variable associated with them. They are simply text fields where you can type information.
Print the document	Click the d Print button.
Open a different HotDocs form file	Click the Open button, and then locate the desired file. (You can also press Ctrl+O .)

2. Optionally, you can perform any of the following tasks:

3. Once you have the assembled form document open for editing, there are several other things you can do. For example, you can:

- Change the field type. (See Change the Field Type.)
- Change the size of a field. (See Resize Form Fields.)
- Change the position or margins of a field. (See Move a Field on a Form and Change the Borders and Margins of a Field.)
- Create a new answer field. (See Create a Form Field.)
- Rotate text in a field. (See Rotate Answers in a Field.)
- Convert answers on the form to bar code format. (See Convert Answers on the Form to Bar Code Format.)
- Insert a graphic into a field. (See Insert a Graphic File in a Form.)
- Cross out static text on the form. (See Cross Out Static Text on the Form.)
- Circle static text on a form. (See Circle Static Text on a Form.)

Create a Form Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

The first step in automating a form template is creating HotDocs fields at each place on the form where a user's information must be merged.

When you create a field, HotDocs determines the type of field you are creating based on the size of the field—if a field is smaller than a certain dimension, HotDocs creates a check-box field. If a field is larger, then it creates an edit field. (You can define these dimensions at HotDocs Options. See Set Properties for New Edit Fields.) Additionally, once you create the field, you can make it a Resource hyperlink or a Control field.

Sometimes when you create a field, HotDocs can detect the borders of the underlying form and adjust its size to fit within those borders. This helps ensure the field fits best in the space allotted. See Detect Borders to Create or Resize a Field.

Once a field is created, you can attach a variable to it.

Please see Check Fillable Fields in a PDF Template for information on working with fields in a template created from a fillable PDF.

To create a form field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the **Select Tool** button.
- 3. Position the mouse pointer at one end of the intended field.

- 4. Press and hold down the left mouse button, then drag the pointer to the opposite corner of the field.
- 5. Release the mouse pointer. The field is created.
- 6. Optionally, click **Detect** to have HotDocs adjust the size of the field to more closely match the underlying static line or lines.
- 7. Optionally, you can customize the appearance of your fields in the following ways:
 - Attach a variable to the field. (See Attach a Variable to a Field.)
 - Change the field type. (See Change the Field Type.)
 - Adjust the size. (See Resize Form Fields.)
 - Change the field's position on the form. (See Move a Field on a Form.)
 - Make a field conditional so the variable is asked only if a condition is true. (See Make a Field Conditional.)

To move multiple fields once they have been created, select the fields and press the arrow key that indicates which direction you want to move them. To move fields more quickly, press the **Shift** key while pressing the arrow keys.

You can also create a standard-size field by double-clicking on the form. Or, you can click to set the field cross-hair and press **F8**.

Select a Field

These instructions can also be used to select fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

In order to work with a field, you must first select it. When you select a field, handles appear on the field borders showing that you can edit the borders or other properties. You can select a single field to work with, or you can select multiple fields to group them or to make the same change to all of the fields.

To select fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the **Select Tool** button
- 3. Click on the field you want to work with.
- 4. Optionally, to select multiple fields, either press the Ctrl key while clicking each field, or press the Shift key, then press the mouse button and drag the mouse pointer to create a bounding frame. All fields included in or touched by this frame will be selected.

5. Use any of the following additional methods for selecting fields or canceling the selection, as described in the following table:

То	Do This
Select all the fields on the current page	Click Select All (Edit menu). (You can also press Ctrl+A .)
Cancel the selection of all selected fields	Click outside the fields, or press the Esc key.
Cancel the selection of only one of a group of selected fields	Hold down the Ctrl key and click that field.
Add fields to your group of selected fields without canceling the selection of those already selected	Hold down the Ctrl key and click the new fields.

If you have trouble selecting the field you want, the field may be in a group. If this is the case, you must first ungroup the fields by clicking **Ungroup** (**Field** menu). (See **Ungroup Form** Fields.)

Click the **Show Fields** button to show and hide field colors.

Create a Check-Box Field

These instructions can also be used to create check-box fields at the **Form Document** tab of the assembly window and in HotDocs Filler.

Some forms include check boxes for users to mark. Check boxes can represent either true/false (or yes/no) options, but they can also represent several predefined options for users. By default, HotDocs merges an X in a check box to indicate that it has been selected, but you can define a different check-box character.

Please see Check Fillable Fields in a PDF Template for information on working with fields in a template created from a fillable PDF.

To create a check-box field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Where the check box appears on the form, double-click within the field borders, or click the **Detect Field** button. A check-box field is created.
- 3. Attach a variable to the field. Your options include True/False and Multiple Choice.

When assigning a True/False variable, if you want to merge a character other than an X, indicate the character by specifying it as a variable format. See Customize a True/False Variable for details. To change the font for the field, see Set Properties for New Check-Box Fields. When assigning a Multiple Choice variable, you should group the fields before attaching the variable. See Group Form Fields So Answers Can Flow From One Field to Another for details.

4. Optionally, specify a condition to make the field dependant on other answers in the form. (See Make a Field Conditional.)

You can change the default measurements used to identify new fields as check boxes. You can also specify a default character other than X to be used when selecting check-box fields. (See Set Properties for New Check-Box Fields.)

If HotDocs creates an Edit field, rather than a Check-box field, select the field, click th	e
Field Properties button, and choose Check box as the Field type.	

Detect Borders to Create or Resize a Field

These instructions can also be used to detect field borders both at the **Form Document** tab of the assembly window and in HotDocs Filler.

When you create fields on a form, you often use the underlying static text as a guide as to where the field should be placed. Often, you want the borders of the fields to match the borders of the static text. To help you align these borders, you can use the **Petect** feature. This feature is useful both when you create a new field and when you need to resize a field to fit within its allotted space.

If HotDocs is unable to automatically create or resize a field to the size and position you want, you must create the field manually. Several factors may contribute to these difficulties:

- **Insufficient surrounding features:** To detect a field, HotDocs searches for surrounding features, such as lines, text, or graphics. If there are insufficient surrounding features, HotDocs may have difficulty detecting a field. In such cases, HotDocs creates a field of the default size.
- Label text: When label text is inside the field area and there is enough space between the text and the bottom of the field area, the **Detect** command will extend the field up to the bottom of the label text. If you want the field to occupy the area to the left or right of the label, or if you want to include the label inside the field, you must create the field manually.

• **Field not completely visible:** If part of the intended field is not visible (for example, it's scrolled out of the window), HotDocs attempts to scroll to detect the field. It is recommended that you use a zoom level such as 🔛 **Fit Page to Width** so HotDocs can find the entire field on the screen.

To detect the borders of the underlying static text

- 1. Edit a form template. (See Edit a Form Template.)
- 2. On the form, complete one of the following steps:
 - If the field is already created, select the field and click the **Detect** button.
 - If you are creating a new field, place your cursor where the new field should be created and click the **Detect** button.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Move a Field on a Form

These instructions can also be used to move fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

As you create fields on a form, you will frequently need to move the fields. You can do this using the mouse or keyboard. You can also specify a precise location using the **Field Properties** dialog box.

To move a field using the mouse

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Position the mouse pointer over the selected field. The cursor changes.
- 3. Hold down the mouse button and drag the field to the new position.

To move a field using the keyboard

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select a field. (See Select a Field.)
- 3. Press the arrow keys to move the field one unit of measurement in that direction. (To move the field more quickly, hold down the **Shift** key as you press the arrow keys.)

To move a field using the Position/Size dialog box

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select a field. (See Select a Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. Click the **Additional** tab. The view changes to show the positioning options.
- In the **Position** group, enter the desired distance from the left and top margins of the page in the Left and **Top** boxes. (You can also click the up or down arrows for each field to change the distance.)

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

If a field you want to move is part of a group, you must first ungroup the fields. (See Ungroup Form Fields.)

The **Field Properties** dialog box includes the **First**, **Previous**, **Next**, and **Last** buttons to move you between fields. When you have multiple fields selected, these buttons are not available.

Align Two or More Fields

When creating fields on a form template, you frequently need to align fields, either vertically or horizontally. Other times, you may need to adjust the height or width of a group of fields so they match each other. Aligning fields can give forms a more professional appearance, as well as minimize problems the user may experience when tabbing between fields during direct-fill assembly.

To align fields in relation to each other

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the fields you want to align. (See Select a Field.)
- 3. Click the Align button. The Align Fields dialog box appears. (You can also right-click and choose Align from the shortcut menu.)
- Select an option in the Horizontal alignment group to align the fields horizontally, or the Vertical alignment group to align the fields vertically. The Example box previews the selected alignment option.
- 5. Click **OK**. The template appears again, and the selected fields are repositioned.

To size fields in relation to each other

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the fields you want to size. (See Select a Field.)
- 3. Click the Align button. The Align Fields dialog box appears. (You can also right-click and choose Align from the shortcut menu.)
- 4. Select the options you need from the **Width** group and/or the **Height** group. The **Example** box previews the selected resizing options.
- 5. Click **OK**. The template appears again, and the selected fields are resized.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Change the Borders and Margins of a Field

These instructions can also be used to change the borders and margins of fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

You can change the borders and margins of a form template field. Borders control the thickness of the field box, and margins control the distance between the border of the field and the answer text.

To change field borders or margins

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field whose borders or margins you want to change.
- 3. Click the **Field Properties** button. The **Field Properties** dialog box appears.
- 4. Click the Layout tab. The view changes to show layout options.
- 5. In the **Borders** and **Margins** groupings, make your selections. (Remember, borders affect the thickness of field boxes, while margins affect the distance between the border of the field and the answer in the field.)

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Copy One or More Form Fields

These instructions can also be used to copy and paste fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Frequently, you need to copy fields on a form. For example, say you need to create a group of fields that all have the same properties. Rather than creating each field individually and modifying its properties, you can create a single field, assign the properties, and then copy the field to create the others in the group.

When a field is copied to a new place in the same template, all the field's properties (size, font, line formatting, fill order, etc.) are copied with it. Similarly, when variables, conditions, or REPEAT instructions are attached to fields, they are also attached to the copied fields. However, if you copy fields from one template to another, the variables, dialogs, and other components used in computations, conditions, or instructions are not copied to the new template's component file. You must copy these items manually. (See Create and Edit Multiple Components Simultaneously or Use One Component File for Multiple Templates.)

When you copy fields to a new location, the cursor position tells HotDocs where to paste the copied fields. If you copy a single field, the lower-left corner of the field will be at the cursor position. Likewise, if you copy two or more fields at the same time, the lower-left corner of an invisible boundary box drawn around all of the copied fields will be at the cursor position. If there is no cursor, the field is pasted on top of the original.

You can copy fields on only one page at a time.

To copy and paste one or more fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field or fields you want to copy. (See Select a Field.)
- Choose Copy (Edit menu). The field is copied to the Windows Clipboard. (You can also select the field and press Ctrl+C, or you can right-click on the field and choose Copy from the shortcut menu.)
- 4. Place the mouse cursor where you want the lower-left corner of the copied field to be, and then click the mouse to set the cross-hair.
- 5. Choose **Paste** (**Edit** menu). The copied field is pasted at that location. (You can also press **Ctrl+V**, or you can right-click on the field and choose **Paste** from the shortcut menu.)

You can also copy a field by holding down the **Ctrl** key while dragging the field to its new location.

If you need to automate an updated version of the template, it may be easier to replace the static content, rather than copy all of the fields. (See Update the Underlying Text in a Form Template.)

If you are working in a fillable PDF template, remember that linked fields are associated with underlying fillable fields in the PDF. If you plan to save the assembled document as a fillable PDF, answers associated with linked fields will only be merged where there are fillable fields on the form. This means that you cannot copy a linked field to a new location on the form and expect the answer to appear there in the resulting fillable PDF. If you want to use the same variable somewhere else in the form, create a regular HotDocs field and assign it the same variable name as the fillable field. Note, however, that in the fillable PDF, the answer will appear static.

Resize Form Fields

These instructions can also be used to resize fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

You may need to adjust the size of a field to fit it into the available space on the form. You can change a field's size using the mouse, the keyboard, or the **Position/Size** tab of the **Field Properties** dialog box.

To resize a field using a mouse

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Click a handle or field border. (When the mouse pointer is on a handle, the pointer changes to a double-arrow, indicating the directions the border can be moved.)
- 4. Hold down the mouse button and drag the border to a new position.

To resize a field using the keyboard

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Press the **Page Up** or **Page Down** keys to move the top border of the field up or down, and the **End** or **Home** keys to move the right border of the field right or left. (To resize the field more quickly, hold down the **Shift** key as you press the keys.)

To resize a field using the Field Properties dialog box

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.

4. Click the **Additional** tab and, in the **Size** group, enter the desired distance from the left and top borders of the field in the **Width** or **Height** boxes. (You can also click the up or down arrows for each field to change the size.) The field size is changed.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Change the Tab Order of Fields

These instructions can also be used to change the tab order of fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

When determining the order to ask questions in the interview, HotDocs begins with the top-left field of the form and processes all of the fields until it reaches the bottom-right field. This processing affects two things—the order questions are asked in the default interview and the order in which the user is able to tab through fields on the form.

Because of this default field ordering, sometimes you may find your dialogs being asked out of order. You may also find that the tab order during direct-fill assembly isn't working as you expect. Additionally, when automating a nontypical table, you may need to change the order in which fields are asked.

Once you specify a fill order for one field, you must specify the order for every field you want processed after that field. In effect, when you specify a fill order for just one field, you set that field to be answered last. This is because all the other fields are still set to *Row 0*, *Column 0*— they are lower than the *Row 1* or *Column 1*, so they will be filled first. Therefore, to change the order of the fields in the middle of the form, you must also change the order for every remaining field.

The following rules control field ordering:

- Fill order is determined first by row, then by column.
- All field values are initially set the same—Row 0, Column 0.
- When fields have different row numbers, fields with lower row numbers are asked first, for example, Row 0, Column 0; Row 1, Column 0; etc.
- When fields have the same row numbers, but different column numbers, fields with lower column numbers are asked first, for example, Row 1, Column 0; Row 1, Column 1; etc.
- Fill order is treated separately for each page of the template—you cannot specify a fill order that runs from one page to another.

To change the tab order for fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. Click the **Additional** tab. The view changes to show ordering options.
- 5. In the **Row** box, type the row number.
- 6. If you need to specify the order for fields in the same row, type a number in the **Column** box.
- 7. Optionally, at the **Order** dialog box, you can click the **First**, **Previous**, **Next**, or **Last** button to save the current field's settings and display the next field's order.

By selecting and ordering different groups of fields, you can use fill order to handle various situations. For example:

- Remove all fill order settings: Select all fields and set the Row and Column numbers to 0.
- **Organize large sections of fields:** Select the desired fields and specify the order.
- Flow answers across fields contrary to the default order (top to bottom, left to right): Set the desired order, then group the fields. (See Group Form Fields So Answers Can Flow From One Field to Another.)
- **Fill table columns in an order different from the static text:** Set the desired order, then group the fields as a table.

To control when a variable is asked without changing fill order, you can use ASK instructions. For example, if you simply want a particular field to be asked first even though it isn't the first field on the form, place an ASK instruction in the top left corner of the page. (See Attach an ASK Instruction to a Field.)

Group Form Fields So Answers Can Flow From One Field to Another

These instructions can also be used to group fields for answer flow-through both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes an answer must begin in one field and continue in other fields. For example, you may have a Text variable whose answer must span two or more pre-printed lines. You can create fields for each line, and then group the fields so that answers flow from one field to another. Fields grouped this way are called run-on groups.

Once fields are grouped, HotDocs treats the group as a single field. This means that the Text variable and any conditions for the variable are assigned to the grouping, not the individual fields in the group.

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot group fillable fields since necessarily, each fillable field is linked to a separate "variable." Answers—both in HotDocs and Adobe—must be entered on individual lines. This means that when you reach the end of one field, you must tab to the next field and continue typing your answer. (This inability to group fields is a limitation of Adobe. If you have access to the form, you may consider replacing individual fields in the group with a single, multiline field. Text you enter may not match the lines perfectly, but you will not have to tab between fields to enter the answer.)

To flow an answer across multiple fields

- 1. Edit a form template or saved form document.
- 2. Create all the text fields across which the answer should flow.
- 3. Select all the fields that will be used for the answer.
- 4. Click **Group** in the **Field** menu. A bounding frame appears around the grouped fields. (You can also right-click and select **Group** from the shortcut menu.)
- 5. Attach a variable to the grouped fields.

When fields are grouped, you cannot change properties for individual fields. You must first ungroup the fields.

A form may require an answer to appear in a series of single-character fields or boxes (for example, a Social Security number). Placing one character or digit in each field requires additional formatting.

You can group fields that appear on separate pages in the form.

By default, HotDocs asks questions in the interview by reading fields in the form from left to right, top to bottom. As it encounters a field, it asks the variable associated with it. If the variable is linked to a dialog, it asks the dialog instead. When directly filling the form, HotDocs tabs through the form fields using this same method. However, sometimes grouped run-on fields can create problems with the tab order. To ensure that text in a run-on group flows in the correct order, you may need to specify a fill order for fields in the group.

Ungroup Form Fields

These instructions can also be used to ungroup fields both at the **Form Document** tab of the assembly window and in HotDocs Filler.

When fields are grouped, you cannot modify the individual fields. You must first ungroup the fields.

To ungroup the fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select one of the grouped fields. The whole group is selected. (See Select a Field.)
- 3. Click **Ungroup** (**Field** menu). (You can also right-click on the field and choose **Ungroup** from the shortcut menu.)

Some properties that were assigned to the group, such as text, margin, and line format properties remain applied to the individual fields that made up the group. Other properties, such as a field order or a condition, are lost until you regroup the fields. Regrouping the fields restores the properties that were previously applied to the group.

Properties of a table, including the name of the repeated dialog, are attached to the first field in a table. Because of this, you should not delete the first field in the group before you regroup the fields. Also, do not close the form before you regroup the fields. If you do either of these things, these properties are lost.

Change the Field Type

These instructions can also be used to change the field's type at the **Form Document** tab of the assembly window and in HotDocs Filler.

When you first create a field, HotDocs creates it as either an edit or check-box field, depending on the field's height and width. Check-box fields are usually used to mark a yes/no response, while edit fields are used for questions that require text, date, or number answers. In addition, you can assign some fields as control fields, which means users can't access or edit the field during direct-fill assembly. You can also create resource fields, which can display helpful information about the form.

Once you choose a field type, you can determine the type of answer that is merged in the field. Your options include text, graphics, and bar codes. You can also use fields to cross out or circle static text on a document.

The following are the different types of fields you can create:

Type of Field	Default Color	Description
Edit	Yellow	Allows users to enter any type of answer, including text, dates, numbers, multiple choice options, or computed values. It is the most commonly used field type.
Check Box	Blue	Allows users to select from two or more options. Usually check-box fields are associated with True/False variables or Multiple Choice variables. Answers in a check-box field are usually indicated by an X or other character. (See Create a Check-Box Field.)
Resource	Orange	Allows users to view helpful information about the form while directly filling the form. Resource fields appear as hyperlinks on the form. When users click the link, a pop-up window containing the useful information opens. (See Create a Resource Link on a Form.)
		Resource fields provide help for the form in general, instead of for a specific variable or dialog.
Control	Green	Allows you to complete "behind-the-scenes" tasks in the template, such as inserting templates or setting the values of variables. If the field contains answer or example text, it will be visible to users, but users won't be able to access the field.

If you are working with a fillable PDF template, linked fields appear using a light blue color. This distinguishes them from regular HotDocs fields. See Check Fillable Fields in a PDF Template

To change the field type

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a new field (See Create a Form Field) or select an existing field. (See Select a Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. At the **Type** tab, make your selection, based on the information in the table above.

You can change the default colors used for form fields at the HotDocs Options dialog box. (See Change Colors in HotDocs Forms.)

Change the Font Used for a Field

These instructions can also be used to change the field's font properties both at the **Form Document** tab of the assembly window and in HotDocs Filler.

You can choose the font that is used for answers on a form. When choosing the font, you can also indicate the style (for example, bold or italics), size, effects, and color that are used.

As is always the case when working with fonts, if the form requires a specific font be used, you must ensure the font is installed on all computers where the form will be viewed.

To change the font for a field or group of fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field or fields whose font you want to change.
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. At the **Type** tab, click **Font**. The **Font** dialog box appears.
- 5. Make your font selections.

You can specify default font properties for all new fields you create at **HotDocs Options**. See **Set Properties for New Edit Fields** for more details.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Format Lines and Paragraphs of Text in a Form Field

These instructions can also be used to format how text appears in a field both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes you need to format the text within a multi-line text field. For example, perhaps you need to indent the first line of a paragraph, or maybe you need to adjust the number of lines that fit in an inch of vertical space. To do this, you can adjust the line format of a field.

To format lines and paragraphs

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a multi-line field.
- 3. Attach a multi-line Text variable to the field. (See Customize a Text Variable.)

- 4. With the field selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 5. Click the **Layout** tab. The view changes to show the different options for formatting the field.
- 6. In the **Line Format** group, complete one of the following options:

То	Do This
Indent the first line of text in the paragraph	Enter how much space to include between the margin of the field and the first character of the answer in the First line indent box. (You can either type the number or click the up or down arrows.)
Force a certain number of lines to appear in an inch of space in the field	Enter the number of lines in the Lines per inch box.
Indicate how many lines can appear in the field	Enter the number of lines in the Maximum lines box. If the answer contains more lines than is specified, the field will overflow.
Indicate the maximum number of characters that can appear in a given line of text	Enter the number of characters in the Max chars/line box. When the number of characters exceeds this limit, text will wrap to the next line.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Rotate Answers in a Field

These instructions can also be used to rotate answers in an answer field both at the **Form Document** tab of the assembly window and in HotDocs Filler.

By default, answers in fields appear in horizontal rows and can be read from left to right. You can rotate text so answers will read from bottom to top, top to bottom, or upside-down from right to left.

To rotate text in a field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the fields you want to rotate. (See Select a Field.)
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. Click the **Layout** tab. The view changes to show field layout properties.
- 5. In the **Rotation** group, select a degree of rotation: **0°**, **90°**, **180°**, or **270°**, for example:



You can enter test text in the test answer box (of the **Type** tab) to preview the selected rotation. See Preview the Formatting of Answers in a Form Field.

When you change the rotation, HotDocs views the bottom of the letters as the bottom of the field. This means that aligning the text horizontally or vertically may have different effects.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Keep Contents of Form Fields from Printing

These instructions can also be used to cross out static text on a form both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes you want field text to appear on the form during direct-fill assembly, but you don't want the text to appear when you print a copy of the form. For example, if you create a resource field, you may not want the resource field text to appear on the form when you print it. You can select an option that keeps this text from printing.

To designate that a field's contents shouldn't be printed

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create or select the field whose text you want to appear only on the form.
- 3. Click the **Field Properties** button. The **Field Properties** dialog box appears.
- 4. At the **Type** tab, select **Non-printing field**.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Cross Out Static Text on the Form

These instructions can also be used to cross out static text on a form both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes a form includes static text that must be crossed out, depending on answers the user enters during the interview. For example, say a form includes a list of medical conditions. Instructions on the form tell you to cross out any conditions that do not apply to you. You could manually cross out these conditions once you've printed the form; however, HotDocs allows you to create a strike-through field, which crosses out the text for you, depending on answers you enter during the interview.

A strike-through field, which is transparent, overlays the static text. You can choose the character that will be used to cross out the text.

To create a strike-through field in a form

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field that covers the static text on the form. (See Create a Form Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears. (You can also right-click and select **Field Properties** from the shortcut menu.)
- 4. Click the **Type** tab.
- 5. In the **Field** type group, select the type of field you want to associate with the strike-through field.
- 6. In the **Display type** group, select **Strike-through**.
- 7. In the **Variable** box, insert a True/False or Multiple Choice variable. This variable sets the conditions when static text should be crossed out. (See Attach a Variable to a Field.)
 - **True/False variable:** Select an example format that merges the strike-through character when the variable is true or false, as appropriate. (See Customize a True/False Variable.)
 - **Multiple Choice variable:** Type the strike-through character in the **Merge Text** field of the option that should cause the strike through, and type **NONE** in the other choices. (See Customize a Multiple Choice Variable.)

You can use the alignment controls on the **Field Properties** dialog box to position the strike-through text more accurately.

To change the font properties of the strike-through character, click the **Font** button at the **Field Properties** dialog box and make your changes there.
Circle Static Text on a Form

These instructions can also be used to circle static text on a form both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes instructions on a form ask users to circle a preprinted option. Rather than make users print the form and manually circle the option using a pen, you can create a form field that overlays a circle on the text. You can attach a variable to the field so that the user can select which option should be circled during the interview. Creating circle fields not only preserves the professional look of the form, but allows the answer associated with the selection to be saved in the answer file.

To create a field that circles text

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field that covers the static option on the form you want to select. (See Create a Form Field.)
- Click the Field Properties button. The Field Properties dialog box appears. (You can also right-click and select Field Properties from the shortcut menu.)
- 4. Click the **Type** tab.
- 5. In the **Field type** group, select the type of field you want to associate with the circled field.
- 6. In the **Display type** group, select **Circled**.
- 7. Assign a True/False or Multiple Choice variable to the field. (See Customize a True/False Variable, Attach a Multiple Choice Variable to a Group of Check Boxes, and Customize a Multiple Choice Variable.)
- 8. Optionally, to control the width of the circle's border, click the **Layout** tab and change the **Line thickness** in the **Circle** group.

To preview the circle (see Preview the Formatting of Answers in a Form Field), make sure you enter text in the **Answer text** box of the **Type** tab.

Insert a Graphic File in a Form

These instructions can also be used to insert graphics in a form both at the Form Document tab of the assembly window and in HotDocs Filler.

Sometimes you may need to include an image on your form, such as a signature or seal. You can create a field and assign as one of its properties a graphic file. Supported file formats include .JPG, .BMP, and .PNG. Image files should be saved to the same folder as the template.

You control the conditions under which the image appears on the form. For example, you can attach a Multiple Choice variable or a True/False variable that merges an image file depending on which option a user chooses. Additionally, you can make the field a control field so that users can't access the field directly.

To create a graphic field using a variable

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field (See Create a Form Field) and attach either a Multiple Choice or True/False variable to it. (See Customize a Multiple Choice Variable or Customize a True/False Variable.)
- 3. If you're using a Multiple Choice variable, edit the variable and, in the **Merge Text** column, enter the file names of the graphics, depending on which options the user chooses.
- 4. If you're using a True/False variable, edit the variable and, in the **Format** box, enter the file name of the graphic, either preceded or followed by a forward slash (to indicate yes/no status).
- 5. Click **OK** at the variable editor and at the **Variable Field** dialog box.
- 6. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 7. At the **Type** tab, select **Image** from the **Display type** group.
- 8. Optionally, change the Field type to Control to restrict the user's ability to modify the field. (Change the Field Type.)

To create a graphic field without using a variable

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field. (See Create a Form Field.)
- 3. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. At the **Type** tab, select **Image** from the **Display type** group.
- 5. Click the **Open** button next to the **Image file name** box. The **Open** dialog box appears.
- 6. Locate and select the desired graphic file and click **Open**. The path and file name appear in the box.
- 7. Optionally, change the Field Type to Control to restrict the user's ability to modify the field. (Change the Field Type.)
- 8. Optionally, click the ^{Show} Variables button in the Automator toolbar to view the image.

The image is visible at the **Form Document** tab.

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot assign an image to a linked field. Instead, use a regular HotDocs field.

Convert Answers on the Form to Bar Code Format

These instructions can also be used to enter bar code text both at the **Form Document** tab of the assembly window and in HotDocs Filler.

You can create a two-dimensional bar code field that displays a single answer or group of answers. This may be useful if the organization to which you submit your documents requires certain data in the document be in bar code format. This allows the organization to scan the information and save it in some data retrieval system, such as a database.

For example, say you must submit a form to the court. When the court files the form, rather than manually entering case information (such as party names, case numbers, etc.), it can simply scan the bar code and have the information automatically entered in the system.

Typically, bar codes display data in a pattern of lines and formats, which should be readable by most hand-held laser scanners or charge-coupled device (CCD) scanners. The scanner should be able to read the size of the bars, which is 0.01 inch.

Use the following tips when working with bar codes:

- The format HotDocs uses for bar codes is PDF417, or Portable Data File 417. It is one of several recognized 2-D bar code formats.
- Bar codes in HotDocs can store about 1,500 characters, including numbers and other standard text symbols.
- Printing a form that has a bar code requires a high-density printer, such as a laser printer.
- Bar code fields can overflow, but all overflow properties assigned to the field are ignored. When too much text is entered for a bar code, no bar code appears. If you are having trouble getting the bar code to fit in the field, you can enlarge the field or reduce the field margins. (See Resize Form Fields or Change the Borders and Margins of a Field.)

There are two methods for creating a bar code:

- **Create a bar code using a variable:** When the answers you want to convert to a bar code must be supplied by the user, you can create a variable, such as a Computation variable, which will then merge the answers in the field and convert them to bar code format.
- **Create a bar code using predefined text:** If the bar code text is always the same, regardless of who is completing the document, you can enter that text at the **Field Properties** dialog box.

To create a bar code using a variable

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field and attach a variable to it, such as a Computation variable. (See Attach a Variable to a Field.)
- If using a Computation variable, create a script that will create a text string that includes all the answers you want encoded. Separate each answer by using a delimiter character, such as a tilde (~) or a vertical bar (|). (See Customize a Computation Variable and Introduction: Instruction and Expression Models.)

For example, Court Branch or District + "|" + Court Mailing Address + "|" + Court City + "|" + Court Zip Code.

- 4. Click **OK** at both the variable editor and the **Variable Field** dialog box. The template appears again.
- 5. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 6. At the **Type** tab, select **Bar code** as the **Display type**.
- 7. Optionally, click the **Bar Code** tab and adjust any of the bar code settings. (See Understand Bar Code Settings.)

When the user assembles the document, the answer will be encoded as a two-dimensional bar code.

To create a bar code using predefined text

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field. (See Create a Form Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. At the **Type** tab, select **Bar code**.
- 5. Type the data you want encoded in the **Bar code text** box. As you type the text, HotDocs displays the bar code in the underlying form field. (You may need to click the **Show Variables** button in the Automator toolbar to hide variable names.) (If you are entering a group of answers that must be interpreted by the bar code scanner as individual answers, you must separate each answer with some sort of delimiting character, such as a tilde (~) or vertical bar (]).)

For example, Superior Court of Ada County | 12 N. Elm Street | Boise | 99999.

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot assign a bar code to a linked field. Instead, use a regular HotDocs field.

For information on changing the bar code specifications, see Understand Bar Code Settings.

To access the **Field Properties** dialog box, you can also right-click and select **Field Properties** from the shortcut menu.

Flow a Single Answer Across Two or More Pages in a Form

These instructions can also be used to flow an answer across two pages both at the **Form Document** tab of the assembly window and in HotDocs Filler.

At times, a single answer field will start at the bottom of one page and continue to the top of the next page. However, HotDocs will not allow you to create a single field that spans across both pages. You can, however, create two fields—one on each page—and then link them using the **Group Fields** command.

To group fields on separate pages

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create the first field in the series of spanned fields. (See Create a Form Field.)
- 3. Assign a Text variable to the field. (See Customize a Text Variable.)
- Select the field and click the Aries Field Properties button. The Field Properties dialog box appears.
- 5. Click the **Overflow** tab, and in the **Group name** box, enter a name. Usually, this is the name of the variable, but it can be any text as long as it is the same for each field in the group.
- 6. On the following page(s), create the fields that will contain any overflow. Do not assign variables to these fields.
- 7. Repeat the preceding steps for each field in the group, using the same group name as you assigned to the first field.

Now, answers that do not fit in the first field will overflow to the next field named in the group.

To span text across multiple fields on the same page, see Group Form Fields So Answers Can Flow From One Field to Another.

At a Glance: The Overflow Status Dialog Box

Overflow Statu	5	? - ×
The hover	highlighted field overflo flow, you can modify th erties for the field	ws. To resolve the ne overflow
Overflow pro	perties	
Shrink the	answer to fit in the fie	ld
Send the	answer to the addendu	m Split answer
Or you can		
Edit] Edit your answer to	fit it in the field
Resize	Resize the field to m	ake it larger
<u>O</u> verride	Override the answer	in the field
Click Ignore to click Cancel to r	proceed without resolv eturn to the highlighter	ing the overflow or d field.
6	Ignore Can	icel

When entering answers at the form document tab, you will see the **Overflow Status** dialog box when you enter an answer that is too large for the field.

At the top of the dialog there are two check boxes A you can select to decide the overflow properties. You can select **Shrink the answer to fit in the field** to reduce the answer's font size to the minimum font size specified in the **Field Properties** dialog box for this field or you can select **Send the answer to the addendum** to send the answer information to an addendum at the end of the document, and inserts an addendum reference in the field. The **Split answer** option only becomes available after selecting **Send the answer to the addendum** and send only the portion of the answer that does not fit in the field to the addendum.

Otherwise you can choose from the three option buttons below. Click the **Edit** button betton betton below. Click the **Edit** button consistent the field so you can shorten the answer, click the **Resize** button constant to select the field so you can click a field border or handle and resize the field (resizing the field to make it larger may resolve the overflow), or click the **Override** button constrained to override the field properties, and place the cursor in the highlighted field so you can edit the answer. When override is used, you can enter any type of answer.

To learn more about when the overflow status dialog box appears follow the links below:

- Check Form Fields for Answer Overflow
- Enter Answers Directly at the Form Document Tab
- Control Warnings During Assembly

Check Form Fields for Answer Overflow

HotDocs can compare the size of a field and the length of the answer. If an answer uses more space than is available in that field, HotDocs warns you and lets you resolve the overflow. Your options for resolving the overflow include reducing the answer's font size, sending answers to the addendum, editing the answer, resizing the field in which the answer is merged, and overriding the field to enter a differently formatted answer.

After you complete an interview, you can manually check for answer overflow. Also, when you direct-fill assemble a form document, HotDocs automatically checks each field when you move to another field. Finally, when you print a document, HotDocs checks for any unresolved answer overflows.

To manually check for field overflow

- 1. At the HotDocs library window, select a form template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After answering the questions in each dialog, click the **Form Document** tab to view the assembled document.
- 3. Click **Check for Overflow** (**Tools** menu). HotDocs begins comparing the size of each field and answer.
- If an answer is too long, HotDocs selects the field and displays the **Overflow Status** dialog box. Information about the status of any answers that overflow is displayed at the top of the dialog box.
- 5. Make adjustments as explained in the following table:

То	Do This
Reduce the answer's font size to the minimum size allowed for the field	Select Shrink the answer to fit in the field . The answer's font size is reduced to a smaller point size. If the answer still doesn't fit in the answer field, you must choose another option for resolving the overflow.
Send the answer to the addendum and insert cross- reference text in the answer field	Select Send the answer to the addendum . The answer is moved to the addendum. HotDocs then merges text in the field that indicates the answer can be found in the addendum. Select Split answer if you want only the part of the answer that doesn't fit in the field to be cent to the addendum.
Change the text of the answer, such as reduce	the field to be sent to the addendum. Click Edit . HotDocs highlights the field that contains the answer so you can edit the text.

the number of words in the answer	Once you edit your answer, click back on the Overflow Status dialog box to have HotDocs the overflow.
Make the answer field larger	Click Resize . HotDocs selects the field so you can click one of its handles to resize it.
	Once you resize the answer field, click back on the Overflow Status dialog box to have HotDocs recheck the overflow.
Override any	Click Override and enter the text you choose in the field.
formats of the field and enter any text in the field	Overriding a field does not change the original answer. In fact, HotDocs continues to use the original answer in any calculations or scripts that require it. It also saves the original answer to the answer file. Choosing to override a field simply allows you to define only the text that needs to appear on the form.
	Once you override the answer field, click back on the Overflow Status dialog box to have HotDocs recheck the overflow.
Resolve the overflow at a later time	Click Ignore . The Overflow Status dialog box is closed and you are able to work with other answer fields in the document. If you click on the overflowing answer field again, or if you check for answer overflow, you will be asked to resolve the overflow again
Resolve the overflow on your own	Choose Close . HotDocs highlights the overflowing field so you can change the answer in whatever way you choose

Once you resolve the overflow, you can click **Close** to close the dialog box. Any other fields that overflow will display a similar dialog box.

You can prevent HotDocs from checking for field overflow during assembly. See Set Rules for Handling Answer Overflow for details.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Customize the Look of the Addendum

These instructions can also be used to customize the look of the addendum both at the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes when assembling a form document, an answer is too long for the form field. One way to resolve this overflow is to send the overflowing text to an addendum, which is a section of the form designed to display such answers. You can modify the appearance of the addendum—for example, you can choose what page margins to use, define what text appears in the headers and footers, and choose the font properties for the text that is used in the addendum. You can also define how pages are numbered in the addendum.

To change the addendum format

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click File > Template Properties > Addendum. The Addendum Properties dialog box appears.
- 3. Complete any of the following steps:

То	Do This
Define how much white space there is between the addendum page edges and the text in the addendum	Enter the dimensions in the Page margins group.
Define the text that appears at the top of each addendum page	Enter the text in the Header text box.
	To learn how to merge page numbers in the header, see Use Answer Overflow and Addendum Text Codes.
Define the text that appears at the bottom of each addendum page	Enter the text in the Footer text box.
	To learn how to merge page numbers in the footer, see Use Answer Overflow and Addendum Text Codes.
Change the font properties (including font face, size, and style) of the header or footer text	In the respective Header properties and Footer properties groups, click the Font button and make the changes at the Font dialog box.
Define how much vertical space the header or footer text requires	In the respective Header properties and Footer properties groups, enter a number in the Height box.
Change the alignment of the header or footer text	In the respective Header properties and Footer properties groups, click the Alignment button and choose your alignment option.

Insert a variable in the header or footer text	In the respective Header properties and Footer properties groups, click the **Variable Field button. The Variable Field dialog box appears, where you can define the variable whose answer you want merged in the header or footer.
Define the font properties (including font face, size, and style) for answers that appear in the addendum	In the Addendum entries group, click the Font button and make your changes at the Font dialog box.
Define how much space there should be between the addendum label and the answer	In the Addendum entries group, enter a number in the Indentation box.
Define how much space there should be between each answer in the addendum	In the Addendum entries group, enter a number in the Space between box.
Make the addendum appear as a pleading paper	Select Number lines to format as pleading paper.

Once an answer has been sent to the addendum, you cannot edit the actual addendum—you must either modify your answers at the **Form Document** tab or **Interview** tab, or you must send the addendum to the word processor (choose **File > Send Addendum To > Word Processor**). You can also send the addendum to the Windows Clipboard (choose **File > Send Addendum To > Clipboard**) to paste it into a different program for editing.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Define Overflow Properties for a Regular Field

These instructions can also be used to change the overflow properties for a field both at the **Form Document** tab of the assembly window and in HotDocs Filler.

When an answer is too large to fit in its field, HotDocs, by default, warns you and asks how to handle the overflow. You can customize the field overflow properties, controlling how HotDocs automatically handles answer overflows.

For example, you can have HotDocs automatically create an addendum, where each answer that overflows is sent. In the field that overflows, HotDocs can insert cross-reference text that points you to a specific

item in the addendum. You can designate the text that is used both for the cross-reference and for the label in the addendum.

Another option is for you to have HotDocs automatically reduce the answer's font size. You can set a minimum point size and control if the answer is automatically reduced to this size before displaying an overflow warning message. (Be careful reducing the font size since others may find the smaller size harder to read.) (See Change the Font Used for a Field.)

How you define overflow properties depends on the type of field you are creating. For example, in an addendum, grouped fields and tables are handled a little differently from regular fields. See Define Overflow Properties for Run-on Fields and Define Overflow Properties for a Table.

To set overflow properties for regular fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field whose overflow properties you want to define.
- 3. Click the **Field Properties** button and click the **Overflow** tab.
- 4. Complete any of the following steps:

То	Do This
Automatically shrink an answer that overflows to a specific point size	Select Shrink answer as needed and enter a point size in the points box.
Automatically send an answer that overflows to the addendum	Select Send answer to addendum . (To send only part of a multi-line answer to the addendum, select Split answer . (See Split a Multi-Line Answer Between the Form and the Addendum.)
Define the text that appears in the answer field when an answer is sent to the addendum	Enter the text in the Cross-reference text box. (See here for a list of reference numbering codes you can use.)
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the Addendum label text box.
Make an answer the first item on its addendum page	Select Begin addendum entry on a new page .
Make an answer the last item on its addendum page	Select Begin new page following addendum entry.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Split a Multi-Line Answer Between the Form and the Addendum

When creating multi-line fields, you can select an overflow option that sends either all of the answer to the addendum, or only the part of the answer that doesn't fit in the field to the addendum.

If you allow the user to choose whether to split the answer between the form and the addendum, you can customize the reference and label text, based on their selection. For example, if the user chooses to send the entire answer to the addendum, you can specify *See Addendum 1* as the cross-reference text. However, if the user chooses to send just part of the answer, you can specify *Continued in Addendum 1* as the reference text.

To choose overflow options for a multi-line answer

- 1. Create a multi-line field. (See Create a Form Field.)
- 2. Select the field and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 3. Click the **Overflow** tab. The view changes to show overflow options.
- 4. Complete any of the following steps:

То	Do This
Always send the answer to the addendum without	Select Send answer to addendum.
prompting the user	To send just the part of the answer that doesn't fit in the field to the addendum, select Split answer .
Specify the text that	
will be merged in the	Enter the text in the Cross-reference text and Addendum label text
field as well as the	boxes, respectively.
addendum if the user	
chooses just to send	To specify alternate text that will be merged in the field and addendum
to addendum	if the user chooses to split the answer between the form and the addendum, type a vertical bar () and then enter the alternate text after the bar. For example:
	Cross-reference text: See Addendum 1 Continued in Addendum 1
	Addendum label text: Addendum 1 (con't) Addendum 1

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF

document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Attach a Text or Form Document to an E-mail Message

After assembling a document, you can attach it to an e-mail message and send it to another user.

Internet-based e-mail services are not compatible with this feature. You must have an e-mail program, such as Microsoft Outlook, installed on your computer.

To attach a document to an e-mail message

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- After providing the required answers, click Send Document To > Mail Recipient (File menu). The Send Document To Mail Recipient dialog box appears.
- 3. Type a name for the attachment, then click **OK**. An e-mail message window appears.
- 4. Complete the e-mail message and send it.

Creating Template and Document Summaries

Introduction: Question and Answer Summaries

Some projects may require a printout of the information in the template. A Question Summary lists each question followed by a blank space so an answer can be written in, as if being filled out like a questionnaire. An Answer Summary lists each question and the associated answer. These printouts can be concise records of exactly what information was used when the document was assembled.

The following is an example of an Answer Summary:

Assembling Documents

Employment Agreement - HotDocs Developer	- • •
<u> </u>	
🗐 🛛 🖌 🔂 🖌 🖓 🚱 🖉 🕼 🖓 🖓 🖓 🖏	
∠ ∽ ∾ ở ฿ ฿ ฿ ฿ ฿ ฿ ฿	
Interview Document Preview Question Summary Answer Summary	
Answer Summary Template: Employment Agreement Answer File: Alice Chamberlain Employee Information Employee Name Alice Chamberlain Employee Gender (X) Female	
Agreement Information	
Agreement Date	
Company Representative	-
For Help, press F1	NUM at

You can specify a one-column or two-column format for your summaries. In one-column format, dialogs, questions, and answers are displayed as an indented outline. In two-column format, the information is presented as a table, with each answer appearing on the same row as its question. (See Change the Way Question and Answer Summaries Appear.)

If you're assembling a form document, you can create a questionnaire by printing a blank copy of the form itself.

View a Question Summary

You can create a concise list of the questions that appear in an interview. This can be useful when you want to create a questionnaire, gathering information from one person so that a second person can enter the information using HotDocs. A question summary can also give a quick overview of what questions will be asked during an interview.

You can format a question summary to appear as an indented bullet list, or as a table with one column for questions and a second column for answers. (See Change the Way Question and Answer Summaries Appear.) Also, in a question summary, you can control which conditional sections are included. (See Control Which Variables are Asked in a Question Summary.)

To assemble a question summary

- 1. Assemble a text or form document. (See Assemble a Text or Form Document.)
- 2. At the assembly window, click the **Question Summary** tab. The assembly window shows the HotDocs variables and other information about the template. (To hide or show the tab, choose **Question Summary Tab** at the **View** menu.)
- 3. Optionally, complete any of the following tasks:

То	Do This
Print a copy of the summary	Click the Print button.
Attaches the summary (in HTML format) to an e-mail message to send to another person	Choose Send Summary to Mail Recipient (File menu).
Send the summary to the word processor so you can edit it or format it	Click the Send Summary to Word Processor button.
Save a copy of the summary as an HTML document	Click the 日 Save Summary button.

View an Answer Summary

An answer summary is a list of the questions and the answers in an assembled document. It provides a way to quickly scan and review information, rather than reading through a document or paging through dialogs in the assembly window.

You can format an answer summary to appear as an indented bullet list, or as a table with one column for questions and a second column for answers. (See Change the Way Question and Answer Summaries Appear.)

To view an answer summary

- 1. At the HotDocs library window, select a template and begin assembling a document. (See Assemble a Text or Form Document.)
- 2. After answering the questions, click the **Answer Summary** tab. The assembly window shows the HotDocs variables and the answers you have entered in the interview.

3. Optionally, perform any of the following tasks:

То	Do This
Print a copy of the summary	Click the Print button.
Attaches the summary (in HTML format) to an e-mail message to send to another person	Choose Send Summary to Mail Recipient (File menu).
Send the summary to the word processor so you can edit it or format it	Click the Send Summary to Word Processor button.
Save a copy of the summary as an HTML document	Click the 日 Save Summary button.

View the Variable Sheet

The Variable Sheet helps you see the relationship between variables and answers. It shows the variables in the interview as well as the answers, variable types, and prompts. While viewing the Variable Sheet, you can sort the different columns.

To view the Variable Sheet

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. After providing the required answers, choose Variable Sheet Tab (View menu).
- 3. Click the **Variable Sheet** tab. The assembly window changes to show the Variable Sheet.
- 4. Optionally, complete any of the following tasks using toolbar buttons and menus:
 - To sort the different lists in alphanumeric order, click one of the column headings.
 - To save the variable sheet as an HTML document, click the **Save Variable Sheet** button.
 - To view the variable sheet in the word processor, click the Send Variable Sheet to Word Processor button.
 - To attach the variable to an e-mail message, click **File > Send Variable Sheet To > Mail Recipient**.
 - To copy the contents of the Variable Sheet into a spreadsheet program, such as Microsoft Excel, click File > Send Variable Sheet To > Spreadsheet.

When sending Variable Sheet data to a spreadsheet, HotDocs automatically looks for Microsoft Excel. If you do not have Excel installed, you will be prompted to copy the contents from the Clipboard to another spreadsheet program.

Saving Answer Files

Introduction: Save Answers

When you assemble a document, you can save the answers you enter in an answer file, which can then be used later to recreate the same document or assemble other documents that require the same information.

For example, perhaps you are creating several estate planning documents that require much of the same information, like names, addresses, and important dates. You can save your answers in an answer file and then use that answer file when assembling other estate planning documents. This eliminates the need to re-enter information you've already provided. Answer files are saved in XML format.

You can organize your answer files with the Answer File Manager, or you can use Windows Explorer to manage your answer files. (See Manage Answer Files.)

See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs for details on making answer files compatible with versions of HotDocs prior to HotDocs 2009 and HotDocs 10.

Create a New Answer File

Often, you want to save answers you enter during an interview so they can be used with other interviews. Saving answers in an answer file and then reusing those answers keeps you from having to enter the same information multiple times.

To create a new answer file by assembling a document

- 1. At the HotDocs library window, select a template and begin assembling a document with a new, untitled answer file. (See Assemble a Text or Form Document.)
- 2. Answer the questions in the interview and click the **Save Answers** button. The **Save Answer File** dialog box appears.
- 3. In the **File name** field, enter a file name. (To save the answer file in a folder other than the default *Answers* folder, click the **Browse** button and select a location.)
- 4. In the **Title** field, enter a name for the new answer file. This name will identify the answer file in Answer File Manager.

- 5. Optionally, type a note in the **Description** field to help identify the answer file. (The description will appear at the **Properties** tab of Answer File Manager. When searching for specific answer files, you can search based on text in the description.)
- 6. If the answer file will be used with HotDocs 2005-2008, select **HotDocs 2005-2008 answer file**. (See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs.)

To save an existing answer file as a new answer file, choose **Save As** (**File** menu) and enter a new name for the answer file. To create a new, empty answer file while in Answer File Manager, click the **New Answer File** button and enter the appropriate information about the file you want to create.

By default, answer files are saved in the answer file folder specified during installation (for example *C:\Documents and Settings\Username\My Documents\HotDocs\Answers*). To change this default location, see Change HotDocs Program File Locations.

Create a Default Answer File

When certain answers are used frequently in a specific document, you can have those answers automatically inserted each time you assemble the document. To do this, create a default answer file for that document.

A default answer file must be saved to the same folder as the template's component file. It must also have the exact same file name as the component file, but the file name extension must be either .ANX or .ANS. In most cases, the file name and path for the component file are identical to the file name and path for the template file. (The exception would be if the template is pointing to a shared component file, in which case you would use the shared component file name.) You can find this information by selecting the template at the HotDocs library window and viewing the **Properties** tab.

To create a default answer file for a document

- 1. At the HotDocs library window, select a template and begin assembling the document, using a new answer file. (See Assemble a Text or Form Document.)
- 2. Type information for only the answers you want to be default answers.
- 3. Click the **Save Answers** button. The **Save Answer File** dialog box appears.
- 4. At the **File name** field, click the **Browse** button and locate the folder that contains the component file. Then enter a name that matches the component file's name, followed by an answer file name extension. For example, if the component file is named *Invoice.cmp*, the answer file name would be *Invoice.anx*.
- 5. In the **Title** field, type a name for the default answer file.
- 6. Optionally, in the **Description** field, type a description for the answer file.

 If the answer file will be used with HotDocs 2005-2008, select HotDocs 2005-2008 answer file. (See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs.)

The next time you assemble the document, HotDocs automatically inserts the answers from the default answer file into the template's answer fields. You can then modify the answers as necessary and save a new answer file. (See Create a New Answer File.)

Suggest an Answer File for Every Assembly

You may use a certain answer file more frequently than any others. You can select an option that will automatically suggest that answer file every time you assemble a document.

To suggest an answer file for every assembly

- 1. At the HotDocs library window, select a template and click ***** Assemble. The Answer File dialog box appears.
- 2. Click the **Open Answer File** button. The **Open Answer File** dialog box appears.
- 3. Choose an answer file and click **Select**. The **Answer File** dialog box appears again.
- 4. Select Automatically select this answer file for use with the next assembly, then click OK.

The next time you assemble any document, the same answer file will automatically be selected at the **Answer File** dialog box. To change the answer file, click the **Open Answer File** button and choose a different file.

Save an Answer File During Assembly

While assembling a document, you can save the answers you have entered. This allows you to save your work and perhaps start a different assembly without closing the assembly window. Saving answers also allows you to use the information you enter with other templates, thus saving you time.

To save answers during assembly

- 1. At the HotDocs library window, select a template and begin assembling the document. (See Assemble a Text or Form Document.)
- 2. At the assembly window, click the Save Answers button. If you are using an existing answer file, the new answers are saved. If you are using a new answer file, the Save Answer File dialog box appears where you can perform any of the following tasks:

То	Do This
Enter the file name	At the File name field, enter a name. When you click OK , the answer file will be saved to the default <i>Answers</i> folder and a reference to it will be added to Answer File Manager.
	To save the answers to a location other than the default <i>Answers</i> folder, click the Rowse button and navigate to the location.
Enter the title that identifies the answer file in Answer File Manager	At the Title field, enter a name or accept the suggestion HotDocs makes.
Add information to help identify the file	At the Description field, type notes about the answer file's purpose or contents. (The description will appear at the Properties tab of the Answer File Manager. When searching for specific answer files, you can search for text in the description.)
Save the answer file so it is compatible with versions of HotDocs earlier than HotDocs 2009	Select HotDocs 2005-2008 answer file . (See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs.)

See Introduction: Use Answer Management for information on using Windows Explorer instead of Answer File Manager.

The Save Answers button may be disabled because the template provider wants to prevent the answer file from being altered. You can assemble a document using new answers, but when you close the assembly window, the new answers are automatically discarded.

Switch Answer Files During Assembly

While assembling a document, you may want to use another answer file. At any point during the interview, you can select a different answer file, then continue the interview using the new answers.

To switch answer files

1. At the HotDocs library window, select a template and begin assembling a document. (See Assemble a Text or Form Document.)

- During the interview, click Open Answers. The Open Answer File dialog box appears. (Depending on the file management settings you have selected, a Windows Explorer dialog box or a window from your document management program may appear instead. See Manage Answer Files.)
- 3. Select an answer file and click **Open**.
- 4. If prompted to save the old answers, click **Save**, **Save As**, or **Don't Save**, depending on your needs. (See Save an Answer File During Assembly.)
- 5. The assembly window appears again, using the newly selected answer file.

When viewing the answer library, you can sort the answer files. To do this, select a folder and click 2, sort. To search for a specific answer file, select the **Find** check box, then type the text for which you are searching in the text field.

At a Glance: The Send Answers to Mail Recipient Dialog Box

Send Answers to Mail Recipient	? 💌
Attachment name:	
B HotDocs 2005-2008 answer file	
	OK Cancel

While assembling a template you can view the **Send Answers to Mail Recipient** dialog box by choosing **Send Answer to..** and then **Mail Recipient** from the **File** menu.

In the **Attachment name** field A you can specify the answer file name used for the e-mail attachment. You can also check the box B to send an answer file that would be compatible with HotDocs 2005-2008.

To learn more about sending answer files by email follow the link below:

• Attach an Answer File to an E-mail Message

Attach an Answer File to an E-mail Message

After gathering information in an interview, you may need to send those answers to another user. You can do this by attaching the answer file to an e-mail message.

Internet-based e-mail services are not compatible with this feature. You must have an e-mail program, such as Microsoft Outlook, installed on your computer.

To attach an answer file to an e-mail message

- 1. At the HotDocs library window, select a template and begin assembly. (See Assemble a Text or Form Document.)
- 2. After completing the interview, click **Send Answers To** (**File** menu). The **Send Answers to Mail Recipient** dialog box appears.
- 3. In the **Attachment name** field, enter a name for the answer file attachment. Make sure to leave the file name extension on the attachment name.
- 4. Optionally, if you want the answer file to be compatible with versions of HotDocs prior to HotDocs 2009, select **HotDocs 2005-2008 answer file**.
- 5. Click **OK**. HotDocs opens an e-mail message and attaches the answer file.

Using Answer File Manager

Introduction: Use Answer Management

By default, each answer file you create or use is managed through the Answer File Manager. The Answer File Manager is similar to the HotDocs template library in that it allows you to organize your answer files according to project specifications. It also lets you sort and search for answer files. (See Introduction: Use HotDocs Libraries.)

At times you may prefer to not use the Answer File Manager. For example, you can manage answer files using a document manager instead of Answer File Manager. Or, if you have multiple users accessing the same set of answer files, you can use Windows Explorer, since only one person can have write access to Answer File Manager. (All other users have read-only access, which keeps them from saving or adding answer files to the answer library.)

HotDocs 6 through HotDocs 2008 supported binary answer libraries (or answer libraries with the .HAL file name extension). Starting with the release of HotDocs 2009, however, HotDocs now supports both binary answer libraries and XML-based answer libraries (which use the .HDL file name extension). By default, HotDocs will continue to use binary answer libraries as long as the file properties of items in the library use characters that are compatible with your system's default language. If you define answer file properties that contain foreign characters, HotDocs

will create an XML-based answer library (so that it can properly display these characters in the **Properties** tab of the library window.)

New of 🖹 🔝 🖄 Add	Remove 1 Import Sort Properties 1 Tool Bar
Answers	Properties Contents Tabs
A 1st Source	
A memorandum	Answers
A test1.anx	
A Fund Type Test.anx	
A WCAB 247631477 BAILEY	
- A Deighton	
A table test	Properties Pane
Addendum test	
A Silks L21624-1 RENNISON	
A Fee Schedule	
A Memorandum	
A SIL T10946-1 Gibbons	
A testing	
A PRM FLA0012. Flaherty	
A SIL T10946-1 Gibbons	
Answer File List	

At a Glance: The Answer File Manager

After clicking the Answer File Manager button on the toolbar (or **Tools** menu) in your HotDocs Library the **Answer File Manager** window opens.

The window is split into two main panes. On the left is the Answer File List, this contains a list of all the answer files you have saved. You can select an answer file from this list by clicking on it.

The second main pane in the **Answer File Manager** window is the **Properties Pane**. When you select an answer file from the list the answer file properties are displayed in the **Properties Pane**. These include the **File name**, **File type**, **File path** and the usage **History**. To view the contents of the answer file go to **Tabs** at the top of the **Properties Pane** and click on the **Contents** tab. This will display a list of all variables, answers and variable types within the answer file.

To access further options for your selected answer file you can use the Toolbar at the top of the window. It contains the following options:

- Answer File: Displays the New Answer File dialog box where you can specify the type of answer file you want to create as well as provide a file name, title, and description for the file.
- **Cut:** Removes (or cuts) the selected answer file to the Windows Clipboard so you can paste it someplace else in the library.
- **Copy:** Copies the selected answer file to the Windows Clipboard so you can paste it someplace else in the library.
- **Paste:** Pastes any items currently saved on the Windows Clipboard to the location you specify in the library.
- Add: Displays the Add Item dialog box where you can add existing answer files or folders to your answer library.
- **Remove:** Removes the selected answer file or folder. When you delete an answer file, you can choose whether to delete just the reference to the file in the library as well as to delete the actual file from disk.
- Mimport: Allows you to locate an answer file and automatically copy it to the default Answers folder. If managing your answer files using Answer File Manager, the file will also be added to the manager.
- $2 \downarrow$ **Sort:** Sorts the items in the currently selected folder in the order you have specified.
- **MAnswer File Properties:** Displays the Item Properties dialog box where you can modify information about the item.
- **Weighted Help:** Opens the relevant page of the HotDocs Help File.

To find out more about using **Answer File Manager** follow the links below:

- Introduction: Use Answer Management
- Open and Close Answer File Manager
- Add an Answer File to Answer File Manager
- Work with Answer File Manager
- Add, Modify, and Delete Folders in Answer File Manager

Open and Close Answer File Manager

When you use Answer File Manager to manage your answer files, you have greater control over the organization of your files (see Work with Answer File Manager). To use Answer File Manager, however, you must first open it.

To open Answer File Manager

• At the HotDocs library, click the Answer File Manager button. Answer File Manager appears.

To close Answer File Manager

• At Answer File Manager, click the **X** in the upper-right corner of the dialog box.

You can also open Answer File Manager by choosing **Answer File Manager** at the **Tools** menu. To close Answer File Manager using the keyboard, press **Alt+F4**.

Add an Answer File to Answer File Manager

Answer File Manager allows you to organize your answer files. When you save your answers after an interview, a reference to the file is automatically added to Answer File Manager. When you need to use answer files that someone else has created (or if you have created them while using Windows Explorer to manage your files), you must add the files to Answer File Manager.

There are two ways to add an answer file to an answer library. You can import the answer file or you can add it using the **BAdd** button.

To import an answer file

- 1. At the HotDocs library window, click the Answer File Manager button. Answer File Manager appears.
- 2. Click Mimport. The Import Answer File dialog box appears.
- 3. Browse to and select the file you want to import. HotDocs confirms you want to import the file.
- 4. Click **Yes**. The answer file is added to **Answer File Manager**. The file is also automatically included in any answer file drop-down lists.

To add an answer file to Answer File Manager using Add

- 1. At the HotDocs library window, click the Answer File Manager button. Answer File Manager appears.
- 2. Click the **Add** button. The **Add Item** dialog box appears.
- 3. In the **File name** field, enter the path and file name of the answer file you want to use (or click **Browse** to locate and open the file).
- 4. In the **Title** field, enter a name for the answer file. This title will identify the answer file in Answer File Manager.
- 5. Optionally, in the **Description** field, enter a description to help you identify the answer file. (This description appears at the **Properties** tab, and can be used when searching for specific answer files.)

By default, answer libraries are saved in binary format, which does not support the use of international characters. If you attempt to add an answer file that contains international characters in its properties to Answer File Manager, HotDocs will prompt you to choose whether to create a new, XML-based answer library (which will support these characters) or convert the foreign characters to an approximate character that can be read in binary format. If you choose to create a new XML-based library, HotDocs 2005-2008 will continue to use the older binary-format library, while HotDocs versions after 2009 will use the XML-based library.

If the answer file is located in a folder other than the default *Answers* folder, you can import the file, which copies the file to the default *Answers* folder and adds it to Answer File Manager. To do this, click **Import** and then locate the answer file.

When adding files to the library, press Ctrl or Shift to select multiple files at once.

Work with Answer File Manager

Answer File Manager helps you organize answer files, much like a HotDocs library helps you organize templates and clause libraries.

When you save answer files, they are added to Answer File Manager, where you can organize them and modify the file properties. You can also view an answer file's history—a list of dates and times the answer file was used, including the template with which it was used. Finally, the Answer File Manager lets you preview the answers currently saved in an answer file.

By default, answer libraries are saved in binary format, which does not support the use of international characters. If you are using a binary answer library and you either add an answer file that contains foreign characters in its properties to the library, or you change the properties or an existing file to include foreign characters, HotDocs will prompt you to either convert the answer library to XML format or manually remove the characters from the file's properties. If you choose to convert the answer library to XML format, you will have two answer libraries— one that works with HotDocs 2008 and earlier and one that works with HotDocs 2009 and later. (A third option allows you to keep the library in its current, non-XML format, but HotDocs replaces characters that are not compatible with your system's default language with the closest approximation available. Usually this means HotDocs will use question marks or square box characters.)

To display Answer File Manager

- 1. At the template library, click the **Answer File Manager** button. The **Answer File Manager** dialog box appears.
- 2. Complete any of the tasks described in the following table:

То	Do This
Create a new, empty answer file	Click the New button. (See Create a New Answer File.)
Add an existing answer file to the library	Click the Add button. (See Add an Answer File to Answer File Manager.)
	To add multiple files one at a time, press Ctrl or Shift to select several files.
Remove a reference to an answer file and optionally delete the actual file from disk	Select the answer file, then click Aremove .
Simultaneously copy an answer file you receive to the <i>Answers</i> folder and add it to the answer library	Click Import . The Import Answer File Name dialog box appears, where you can browse to the answer file and select it. When you click OK , the file is copied to the <i>Answers</i> folder and added to the library.
Sort answer files in alphanumeric order, based on answer file titles	Select a folder, then click the $2 \downarrow$ Sort button. The contents of the folder are reorganized.
Search for a specific answer file	Select Find , then type the text from the answer file's title or description for which you are searching in the text field. Only files that have matching text in their title or description are shown.
View the answers currently	Select the answer file, then click the Contents tab.
saved in an answer file	In the Contents tab, you can sort the columns by clicking the column headings.
View the history of when the selected answer file was used to assemble various templates	Select the answer file, then click the Properties tab.
Change the title, path, file name, description, or format of an answer file	Select the answer file, then click the Properties button and make the necessary changes.
	Changing the file path does not change the location of the actual answer file. If you change the file path without moving the actual file to the new location on disk, that answer file will not be accessible.

Add a folder so that you can better organize groups of answer files	Click the Add button, then select Folder at the Type drop- down field. (See Add, Modify, and Delete Folders in Answer File Manager.)
Modify a folder's title or description	Select the folder, then click Properties . (See Add, Modify, and Delete Folders in Answer File Manager.)
Delete a folder	Select the folder, then click Remove . HotDocs asks what you want to do with any answer files within the folder. (See Add, Modify, and Delete Folders in Answer File Manager.)
Move an answer file into a folder	Double-click the folder to open it, then select the answer file and drag it to the folder.
Change the format of an answer file so it is compatible with a specific version of HotDocs	Select the answer file, and then click Properties. The Item Properties dialog box appears where you can select HotDocs 2005-2008 answer file . (See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs for details.)
Use Windows Explorer to access answer files instead of Answer File Manager	Change the setting at the HotDocs Options dialog box. (See Manage Answer Files.)

Add, Modify, and Delete Folders in Answer File Manager

To help organize your answer files, you can place them in folders and subfolders within Answer File Manager.

To add a new folder

- 1. At the HotDocs library window, click the Answer File Manager button. The Answer File Manager appears.
- 2. Click the **Add** button. The **Add Item** dialog box appears.
- 3. In the **Type** drop-down list, select **Folder**. The dialog box changes to show the options needed for new folders.
- 4. In the **Title** field, enter a folder name.
- 5. Optionally, in the **Description** field, enter notes to explain why the folder was created, or what kinds of answer files it contains. (This description appears at the **Properties** tab, and can be used when searching for specific answer files.)
- 6. Click **OK.** The folder is added to the answer library. You can work with the folder using the following commands:

То	Do This
----	---------

Move an answer file to a specific folder	Double-click the folder to open it, then select an answer file and drag it to the folder.
Modify the title or description	Select the folder and click the Properties button. The Item Properties dialog box appears, where you can change the title or description.
Sort the contents of a folder in alphanumeric order	Select the folder and click $2 \downarrow $ Sort . The answer files are rearranged based on title.
Remove a folder	Select a folder and click Remove . If you want to remove the items in the folder as well, select Remove the unselected items contained in the selected folders . Otherwise, any files referenced in the folder will be moved to the next folder level.

Using Published and Web-Based Files

Introduction: Use Published Files on the Internet

When you install HotDocs, the installation program sets up HotDocs as a helper program for Microsoft Internet Explorer. This makes it possible for the browser to communicate with HotDocs. When you click on a hyperlink to an auto-install or auto-assemble file, the browser starts HotDocs and sends it the file. Then HotDocs begins installing or assembling the file.

In HotDocs, you can also add auto-assemble files from the Internet or from a corporate intranet to your HotDocs library. This allows you to access the files without starting your browser. If you will be unable to connect to the Internet at the time of assembly, you can download and cache copies of the auto-assemble files for use offline. If you add the URL for a Web page to a HotDocs library, you can start your browser from within HotDocs and display that Web page.

In HotDocs, if you are using a template that has been enabled for answer uploading, you can upload the answers you used to assemble a document to a Web server. (Note, however, that the URL of the Web server must be specified by the template developer.)

Finally, if you use a browser other than Internet Explorer, you may need to manually register HotDocs to work with your browser.

Register HotDocs to Work with Web Browsers

If HotDocs is registered as a helper program with your Web browser, you can click on a HotDocs autoinstall (.HDI) or auto-assemble (.HDA) file listed on a Web page and have your browser start HotDocs and pass the file to it.

The HotDocs installation program automatically registers HotDocs with Microsoft Internet Explorer. If you use a different Web browser, you will need to register HotDocs as a helper program yourself. Specific instructions for your browser should be available in your browser's online help system. (Search for the keywords MIME types, file types, helper programs, or associations.)

To register HotDocs with a Web browser (other than Internet Explorer)

- Find your browser's list of MIME types. A MIME type is an association that tells the browser what to do when you download a particular type of file. Each entry in the list of MIME types consists of a MIME type, one or more associated file name extensions, and an action to take on files of this type—usually either by displaying the file in the browser, by using a plug-in to view it, or by starting a helper program. Depending on the browser, each entry may also have a descriptive name.
- 2. For HotDocs auto-assemble files, make an entry in the MIME types list using the following settings:

Option	Value
Name (optional)	HotDocs Auto-Assemble File
MIME type	application/x-hotdocs-auto
Extension	hda
Launch Application	hd_dispatch.exe /ha=
	By default, the launch application executable is found in C:\Program Files\HotDocs 6.

3. For HotDocs auto-install files, make an entry in the MIME types list using the following settings:

Option	Value
Name (optional)	HotDocs Auto-Install File
MIME type	application/x-hotdocs
Extension	hdi
Launch Application	hd_dispatch.exe /hi=
	By default, the launch application executable is found in C:\Program Files\HotDocs 6.

4. Save these settings and restart your browser.

If you upgrade or change browsers after installing HotDocs, you may need to reinstall HotDocs so it will be registered correctly with the new browser.

Add an Auto-Assemble File on the Internet to a HotDocs Library

You can add an auto-assemble file located on the Internet to the HotDocs library. You must know the URL for the file's location on the Internet or intranet. When you select the auto-assemble file at the library window, and click the **Assemble** button, HotDocs downloads the file and begins assembling.

To make sure you always have the most current version of the auto-assemble file, you must be connected to the Internet at the time you assemble it. Otherwise, HotDocs uses the version of the file that has been cached.

To add an Internet-based auto-assemble file to a library

- 1. At the HotDocs library window, select the folder in which you want the auto-assemble file and click the **add** button. The **Add Item** dialog box appears.
- 2. In the **Type** drop-down list, select **Auto-Assemble File**.
- 3. At the **File name** field, type the URL for the file's location on the Internet or intranet, or click the Browse button to locate the file on your local disk or network and click OK.

Make sure the URL is entered correctly. If you don't include **http://** or **https://** or **ftp://** (or if you are missing a slash or a colon), HotDocs will treat the URL as a regular file path.

- 4. In the **Title** field, type a title for the auto-assemble file. The title identifies the file in the template library.
- 5. Optionally, type a description in the **Description** field. The description appears at the **Properties** tab

The first time you use an auto-assemble file that is stored on a Web server to assemble a document, HotDocs caches the HDA on your hard disk. Then, each time you use the file, it checks to see if a newer version of the file is available and, if so, automatically downloads it. You can also force HotDocs to check for newer versions by choosing **Refresh Cache** from the **Tools** menu. (See Download Auto-Assemble Files for Use Offline.)

Download Auto-Assemble Files for Use Offline

In HotDocs, you can download the latest version of an auto-assemble (.HDA) file and store it on your local drive. When you are assembling, if HotDocs can't connect to the Internet or intranet, it uses the local copy.

To download auto-assemble files for use offline

- 1. At the HotDocs library window, select the auto-assemble file you want to download. You can download all the auto-assemble files in a folder by selecting the folder.
- 2. Click Refresh Cache (Tools menu).

At a Glance: The Install Template Set dialog box

Install Template Set	? 💌
HotDocs is ready to install the template set	
CUKAdditionalTutorials	
Specify where the templates should be installed:	
Public Documents (HotDocs \Templates	
Choose where the library file should be installed:	
Public Documents\HotDocs\Templates\	•
	OK Cancel

You can open the **Install Template Set** dialog box by clicking **Install Templates...** in the **File** menu and navigating to the **.hdi** file you would like to install.

In the first field A you can enter where you would like HotDocs to save the template files on your computer. Either by typing the file path into the text field or by clicking the A **Browse** button and navigating to the correct folder. By default HotDocs will install them in the default **Templates** folder in your Documents.

Using the drop-down list **B** below you can choose to install the library file in the default **Templates** folder, default **Libraries** folder or in the same folder you specified for the templates in the field above **A**.

To find out more about installing template sets follow the link below:

• Install a Template Set

Install a Template Set

Some template sets are published as an auto-install file. Because these templates are published, you can't simply add the templates to an existing library. Instead, you must first install them. When you do, HotDocs installs the library and templates to the locations you specify. You can then access the templates.

To install a template set

- 1. At the HotDocs library window, click **Install Templates** (File menu). The **Install Templates** dialog box appears.
- Locate and select the auto-install (.HDI) file you want to install, then click Install. The Install Template Set dialog box appears.
- 3. At the **Specify where the templates should be installed** field, accept the suggested *Template Sets* folder location, or click the **Browse** button to choose a different location.
- 4. Click the **Choose where the library file should be installed** drop-down button and choose whether to save the library to the shared *Templates* folder, the user-specific *Libraries* folder, or, if you installed the templates to a custom folder in step 3, to the same folder as specified for the templates.
- 5. Click **OK**. HotDocs installs the template set (including the library) to the locations you specified.

	cost effec

At a Glance: The Update Template Sets dialog box

After opening **Dupdate Template Sets** from the **Tools** menu in your HotDocs Library you can see a list of your template sets and when they were last checked for updates. Use the mouse to select the sets you would like to update. Or use the **Select All** and **Clear All** buttons below to select all or none of the sets. When you have chosen the sets you would like to update click the **OK** button.

To find out more about updating template sets follow the link below:

• Check for Template Set Updates

Check for Template Set Updates

If you are using a published template set, HotDocs can check for updates to the set at regular intervals and notify you when updates are available. Updates can include new or revised template files, or they may include an important message from the template provider. Although the frequency of automatic update checks is set by the template provider, you can manually check for template set updates at any time.

To check for template set updates

- 1. At the HotDocs library window, click **Update Template Sets** (**Tools** menu). The **Update Template Sets** dialog box appears.
- 2. Select the template sets for which you want to check for updates, then click **OK**. The **Template Set Update Progress** dialog box appears, showing the progress as HotDocs checks for updates.
- 3. When HotDocs has finished checking for updates, the **Install Template Set Updates** dialog box appears, displaying a list of updates. You can work with this list as described in the following table:

То	Do This
Install specific update items	Select Install for each item you want to install.
Discard an update item to prevent HotDocs from prompting you to install the update	Select Discard for the item. If this box is disabled, the template provider has marked it as a required update and it cannot be discarded. (You can choose to not install the update, but it will be displayed every time you check for updates.)
View a list of discarded updates	Select Discarded updates . The list changes to include discarded items.
View a list of all installed updates	Select Installed updates . The list changes to include installed updates.
Reinstall an update	Select Installed updates , then select the update you want to reinstall from the list.

4. Click **Continue**. HotDocs installs the selected updates.

You can hide the **Template Set Update Progress** dialog box or choose which template sets are checked for updates in the HotDocs Options dialog box. (See Determine How Frequently HotDocs Checks for Template Set Updates.)

Upload Answers

If a template has been enabled for answer uploading, the answers you provide to assemble a document can be uploaded to a Web server. The template developer specifies where and when answers will be uploaded. The developer may also specify that HotDocs automatically uploads the answers.

To upload answers

- If a template is enabled for uploading answers, HotDocs automatically displays the Upload Answers dialog box after you assemble a document and close the assembly window. (If the Upload Answers dialog box doesn't appear, it means either that the template isn't enabled for answer uploading, or that the template developer specified that HotDocs must always upload the answers.)
- Click one of the upload buttons to finish the upload process: Upload Now (uploads the answers immediately), Upload Later (lets you upload either when you start or exit HotDocs), or Don't Upload (doesn't upload the answers at all).

If errors occur when HotDocs tries to upload the answers (or if you click the **Upload Later** button), HotDocs creates a data file that contains the information needed about the files you need to upload. Then, every time HotDocs starts or closes, it checks to see if files are waiting to be uploaded and prompts you to upload them.

If you choose to **Upload Later**, you can then upload the files from the library window at any time by clicking **Upload Answers** (**Tools** menu).

Add a Web Page to a HotDocs Library

You can include a URL in a HotDocs template library, making it possible to open a Web page and download published files. You can also use a URL to link to a support site where information about the template set is available.

When you click a URL in the template library, HotDocs launches a Web browser and displays the specified Web page.

To add a Web page to a template library

- 1. At the HotDocs library, open the folder to which you want to add the URL.
- Click the add button. The Add Item dialog box appears.
- 3. At the Type drop-down list, select Web Address.

4. In the **URL** field, enter the URL. Or, click the **QBrowse** button to locate the address. Then close the browser window to save the URL to the **Add Item** dialog box.

You can prevent HotDocs from checking for field overflow during assembly. See Set Rules for Handling Answer Overflow for details.

- 5. In the **Title** field, enter a title for the URL. The title identifies the item in the template library.
- 6. Optionally, type a description in the **Description** field. The description appears at the **Properties** tab.

Assembling Documents Linked to a Database

Introduction: Assemble Documents Linked to a Database

For the most part, templates linked to database tables are assembled just like regular unlinked templates. However, instead of entering answers to questions, you select a record or records from a database table and HotDocs uses the information in those records and merges them into the assembled document. You can save your record selections in a HotDocs answer file.

Depending on how the template was designed, you may also have the ability to sort and filter records you view in the database table.

Select Records During Document Assembly

Instead of displaying questions during the interview for which you must enter individual answers, HotDocs presents a list of records from which you can choose. The information in the record or records you select will be merged into the assembled document.

In some situations, the template developer may allow you to select multiple records from the table, for example, to create a list of answers.

To select records from a database table

- 1. Assemble a document. (See Assemble a Text or Form Document.)
- 2. When a database table appears, click a single row in the table, or press and hold either **Shift** or **Ctrl** to click multiple rows.
- 3. If selecting a single row, complete the following steps:
- Click **Select**. HotDocs places the record in the **Selected Item** pane of the assembly window. (You can also double-click a record and have it automatically placed in the **Selected Item** pane.)
- Click **Next**. HotDocs merges the answers from the database record into the assembled document.
- 4. If selecting multiple rows, complete the following steps:
 - Click **Select.** HotDocs adds the records you just selected to the **Selected Items** pane of the assembly window. (Additionally, to select multiple records, double-click the cell that uses a thicker border in the last record you selected.)
 - Optionally, select a record and click the **Up** button (moves the record up to the next row in the list of records) or the **Down** button (moves the record down to the next row in the list of records). You can also click the column headings of the **Selected Items** list to automatically sort selected rows by any field. (These options are available if the template developer has given you permission to sort your records. If these buttons do not appear, you cannot change the order of selected rows.)
 - Click **Next**. HotDocs merges the answers from the database into the assembled document.

If the template developer has assigned a filter to the table, you may see different results. For example, if only one record meets the filter criteria, that record will automatically be merged into the document without appearing in the interview. Similarly, if no records meet the filter criteria, HotDocs will ask the questions and allow you to enter your own answers. Finally, in some situations, the template developer may have opted to have any selections returned by the filter automatically merged into the assembled document without first prompting you.

To remove a selected record, you can either select a different record, or you can click **XClear**.

To remove a single selected record from the **Selected Items** list, select the record and click **Clear**. To remove all selected records from the list, click **Clear All**.

Sort Database Records During Assembly

If the template developer has allowed it, you can sort the list of records in the database table to show the records in ascending (*A to Z,1 to 9*) or descending (*Z to A, 9 to 1*) order. You can sort on any column in the table. Additionally, if the template developer has allowed you to select multiple records (in order to create a list of answers), you can choose which order the answers will be merged into the document.

To sort database records in alphanumeric order

1. Assemble a document. (See Assemble a Text or Form Document.)

- 2. When the database table appears, click the heading for the column on which you want to sort. HotDocs sorts the list in alphanumeric order (*A to Z*, *1 to 9*.)
- 3. Optionally, to sort the list in reverse alphanumeric order, click the column heading again. HotDocs lists the entries from *Z* to *A*, *9* to *1*.
- 4. Optionally, if you have selected multiple records and want to control the order the answers are merged into the assembled document, click a record in the **Selected Items** list and then click the **Up** button (moves the record up to the next row in the list of records) or the **Down** button (moves the record down to the next row in the list of records). You can also click the column headings in the **Selected Items** list to automatically sort the selected items by any field.

You may also be able to filter the list of records to show only those you think will be relevant to your selection. See Filter Database Records During Assembly for details.

Filter Database Records During Assembly

If the template developer has allowed it, you can filter the list of records in the database table to show only those records you think are relevant. To do this, you specify conditions that must be met for the records to appear. You can filter based on any field in the table.

To filter a list of records in a database table during assembly

- 1. Assemble a document. (See Assemble a Text or Form Document.)
- 2. When the database table appears, select **Find records where all conditions are met**. This causes filtering options to appear at the top of each database column.
- 3. In the database table, select the drop-down button at the top of the column on which you want to filter and select a comparison operator. A comparison operator allows you to compare two values. If the comparison results in a true statement, the record will be included. If it's false, it will not be included.
- 4. In the field directly below the comparison operator, type the value against which you want to filter. For example, to view only those companies whose names start with the letter **A**, you would click the drop-down button in the **Company** column, select **begins with**, and type the letter **A** in the next cell.
- 5. Click **Apply Filter**. HotDocs filters the list of records to show only those that meet the criteria you specified.

Once the list is filtered, you can select a record or records (see Select Records During Document Assembly) and continue with the interview.

When you apply the filter, only those records that meet every condition you specified will be returned. This is because **<all>** is selected at the **Find records where all conditions are met**

prompt. You can click the drop-down button and select **<any>**, which will return a record, even if only one of the conditions is true.

You can sort a list of records in alphanumeric order. For details, see Sort Database Records During Assembly.

Save Selected Records in an Answer File

If you are using a database to provide your answers during document assembly, you can save your answers using the same steps you follow when saving regular answer files. (See Create a New Answer File.)

However, if you think you may need an exact replica of a document assembled from a template that includes a database component, be sure to save a copy of the document. If you try to reassemble the document later, even if you use the same answer file, the document may not come out exactly the same because the information stored in the connected database may have changed.

Introduction: HotDocs Options

You can change the way HotDocs looks and works by changing your preferences by clicking **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library. These changes affect the program regardless of which template you are using. (Some options, however, can be controlled at the template or component file level.)

For example, from the **HotDocs Options** dialog box, you can specify several options that will help you develop templates more efficiently. You can also control the way HotDocs presents interview information during assembly, and you can control the way the information is presented in an assembled document. Finally, some options let you control where you save your HotDocs files—both program files as well as word processor files.

Settings specified at HotDocs Options are user-specific and are saved in the Current User key of the Windows System Registry. For information on working with the registry, see the Windows Help file.

Interviews and Dialogs

At a Glance: Interviews and Dialogs (HotDocs Options)

HotDocs Options		? 🗙
Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Order Processors File Locations File Management Template Set Updates Plugins	Interviews and Dialogs Interview Outline Display repeat icons in interview ou Automatically expand child dialog entrie Dialog Navigation Next button first in Navigation Bar Select existing answers when tabbi Display warning when clicking Add A Enter key action in single-line fields: Show answer field resource button: When answers are reformatted: Answer Entry Date detection: Date display format: Require four-digit year in dates	Itline es: Completely tab order ing between fields Another button at empty dialog Next dialog For the current field only Don't leave dialog Month Day Year 3 Jun 1990 Century rollover year: 50
		OK Cancel

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you click on **Interviews and Dialogs** you will see the **Interviews and Dialogs Options** dialog box, from here you can set options to change how HotDocs interviews are presented.

The first section **B** has two options you can set for the interview outline. Use the check box to choose if HotDocs will display the special repeat icons in the interview outline and use the drop-down list to decide if HotDocs will automatically expand child dialog entries **Completely**, **Partially** or **Not At All**.

In the second section is the options relate to dialog navigation within the interview. At the top of this section there are three check box options. Using these you can decide if you would like the **Next** button to be first in tab order on the **Navigation** bar, if HotDocs should select existing answers when tabbing between fields, and/or if HotDocs should display a warning when clicking the **Add Another** button at an empty dialog. Below this are three drop down menus. The first drop-down decides whether clicking the Enter key in single-line fields moves the curser on to the **Next dialog**, **Next answer field**, or has **No action**. The second drop-down decides if the answer field resource button is shown **For the current field only** or **For all fields with a resource**. When answers in the dialog are reformatted, the third drop-down list allows you to choose if you **Don't leave the dialog**, **Pause before leaving the dialog** or **Leave dialog anyway**.

The last section **D** contains options for the entering of dates in an interview dialog. The first drop-down list allows you to choose between **Day Month Year** or **Month Day Year** format for date detection. The

next drop-down list allows you to choose the display format for a date in the interview. Below is a check box that let you choose if HotDocs will require four-digit years to be given in date answers. If you uncheck this box then the number field to the right will ungray. You can use this to specify the century rollover year, this decides at what two digit number HotDocs will stop using the current century and start using the previous century to fill in the first two digits. For example if you specify a rollover value of 34, dates entered as 5/14/34 will appear as 14 May 2034. A date entered as 5/14/35 will appear as 14 May 1935

To find out more about setting options for Interviews and Dialogs follow the links below:

- Display Repeated Dialogs Using a Special Icon
- Control How Child Dialogs Appear in the Interview Outline
- Control the Tab Order of Buttons on the Navigation Bar
- Select Existing Answers When Tabbing Between Answer Fields
- Display Warning When Trying to Add a New Repetition
- Control the Functionality of the Enter Key During the Interview
- Display Resource Buttons Next to Answer Fields
- Warn When HotDocs Reformats Date and Number Answers
- Specify How HotDocs Should Process the Date Order
- Control How HotDocs Handles Two-Digit and Four-Digit Years
- Change the Way Dates Appear in Answer Fields

Display Repeated Dialogs Using a Special Icon

When HotDocs starts an assembly, it displays an interview outline and a dialog pane. The interview outline contains a list of all the dialogs in a given template. The dialogs appear as icons, followed by the name of the dialog. If HotDocs encounters a repeated dialog, it can display an icon that more closely represents a repeated dialog: 1. When you click this icon, the interview outline expands to show each repeated dialog as entries underneath the dialog title. These entries are numbered to show which repetition you are on.

To display repeated dialogs using a special icon

- 1. At the HotDocs library, click **Options**.
- 2. Click the **Interviews and Dialogs** folder. The window shows several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the Interview Outline group, select Display repeat icons in interview outline.

To have HotDocs show repeated dialogs in the main level of the interview outline, clear **Display repeat icons in interview outline**.

Control How Child Dialogs Appear in the Interview Outline

As you assemble documents, HotDocs displays the dialogs from the template in the interview outline, which can include inserted dialogs (child dialogs). You can control just how much detail the outline shows for each parent and child dialog: a full view of all dialogs, a view of just the current dialog, or a view of just the parent dialog.

To change the appearance of child dialogs in the interview outline

- 1. At the HotDocs library, click the **Options** button.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the **Interview Outline** group, click the **Automatically expand child dialog entries** drop-down button and select one of the following options:
 - Choose **Completely** to have HotDocs expand the interview outline to show the relationship between all related dialogs.
 - Choose **Partially** to have HotDocs expand the interview outline to show only the dialog you are currently answering.
 - Choose **Not at all** to have HotDocs not expand the interview at all and show only the parent dialog in the interview outline.

These views are controlled when you use the navigation bar or shortcut keys to complete an interview. Once you click in the interview outline, expansion of the outline is controlled by the mouse.

Control the Tab Order of Buttons on the Navigation Bar

Sometimes as you navigate through an interview, you may press the **Tab** key to move between options in a dialog—including the buttons on the Navigation Bar. You can select an option that causes the **Next** button to always be the first button on the navigation bar to which you tab after answering questions in the interview. When this option is cleared and you press **Tab** after the last question in the interview, HotDocs will tab to the first active button in the Navigation Bar.

To control the Navigation Bar tab order

1. At the HotDocs library, click the 🗹 **Options** button. The **HotDocs Options** dialog box appears.

- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the **Dialog Navigation** group, select **Next button first in Navigation Bar tab order**.

Now when tabbing from the last answer field in the dialog to the Navigation Bar, you will automatically tab to the **Next** button, regardless of whether the **First Dialog**, **Previous Unanswered**, or **Previous** buttons are active.

Select Existing Answers When Tabbing Between Answer Fields

During an interview, you can have HotDocs automatically highlight existing text in an answer field when you tab to it. When you do this, you can immediately type a new answer without having to select the text first. If you need to edit part of an answer, rather than replace it entirely, you can clear this option.

To control text selection during the interview

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the **Dialog Navigation** group, select **Select existing answers when tabbing between fields** to have existing text highlighted when you tab to a field.

Display Warning When Trying to Add a New Repetition

When answering repeated dialogs, you click the **HAdd Another** button to add a new, empty repetition to the series of dialogs. However, if you are already in a new, empty dialog, clicking this button has no effect. You can have HotDocs display a warning the reminds you that you must first answer at least one question in the current dialog before you can add a new repetition.

To display a warning when trying to add a new repetition to a dialog

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. Select **Display warning when clicking Add Another button at empty dialog**.

Control the Functionality of the Enter Key During the Interview

You can control the functionality of the **Enter** key as you enter answers in a dialog. For example, if you want to use the **Enter** key to advance to the next dialog instead of the next answer field, you can specify an option that will make this possible.

To specify how the Enter key functions during

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the **Dialog Navigation** group, click the **Enter key action in single-line fields** drop-down button and select an option:
 - Choose **Next dialog** to have the **Enter** key display the next dialog.
 - Choose **Next answer field** to have the **Enter** key move your cursor to the next answer field.
 - Choose **No action** to make the **Enter** key not do anything.

Pressing **Enter** in a multi-line text field will insert a line break, while pressing **Ctrl+Enter** will insert a paragraph end. (This affects Word users only.)

Display Resource Buttons Next to Answer Fields

In addition to viewing helpful information in the resource pane, you can specify an option that will display

a **Resource** button next to answer fields in the dialog pane. (Clicking this button causes the resource text to appear in a pop-up dialog.) You can also control when the button appears.

To have HotDocs display a resource button next to an answer field

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the **Dialog Navigation** group, click the **Show answer field resource button** drop-down and select an option:
 - For the current field only causes HotDocs to display a resource button next to an answer field only when that field is active (or when you are answering it).
 - For all fields with a resource causes HotDocs to simultaneously display resource buttons for all answer fields in the dialog that contain a resource. The buttons appear regardless of which answer field is active.

Warn When HotDocs Reformats Date and Number Answers

When you type an answer in an answer field, HotDocs sometimes reads the answer and reformats it to appear a certain way. For example, if you type *February 6, 2006* in a date field, HotDocs will reformat the answer to appear as 6 *Feb 2006*. Sometimes this reformatting can reveal an incorrectly interpreted answer, especially if the answer you typed is ambiguous. For example, the date 05/06/2006 may either be interpreted as *June 5, 2006* or as *May 6, 2006*.

Depending on where the reformatted answer appears in the dialog, you may not even be aware that it has been reformatted. For example, when you enter an answer in one of the first answer fields in a dialog and then move to the next field, you will most likely see that your answer has been reformatted. However, if you enter an answer in the last answer field and click **Next** to advance to the next dialog, you may not see that the answer has been reformatted.

You can specify an option that forces HotDocs to alert you that an answer has changed before HotDocs advances to the next dialog.

To have HotDocs tell you it has reformatted an answer before advancing to the next dialog

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. Click the **When answers are reformatted** drop-down button and select one of the following options:
 - Choose **Leave dialog anyway** to have HotDocs immediately advance to the next dialog in the interview without showing you the reformatted answer.
 - Choose **Pause before leaving dialog** to cause HotDocs to reformat the variable and redisplay it for one second before advancing to the next dialog.
 - Choose **Don't leave dialog** to cause HotDocs to reformat the variable and stay at the current dialog until you click **Next** again.

This answer reformatting affects only how answers are displayed in the dialog pane. Answers in the assembled document are formatted as the developer specifies or according to the **Default date format** property.

Specify How HotDocs Should Process the Date Order

When you type an answer for a Date variable, the first thing HotDocs does to process the information is to separate what you type into the different parts of a date, for example, the month, the day of the month, and the year. Once it has determined this, it formats the date and then merges the answer into the assembled document. Often, if the user spells out the month when typing a date (for example, *May 6, 2006*), HotDocs can easily distinguish between the different parts. However, if a user enters an all-digit value, such as *05/06/2006*, HotDocs could process the date one of two ways—either as *May 6, 2006* or as *5 June 2006*.

You can specify an option that tells HotDocs the order in which you want months and days of months to be processed—either as *Month Day Year*, or as *Day Month Year*.

To specify a date order format for Date variables

- 1. At the HotDocs library, click the 🗹 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. At the **Answer Entry** group, click the **Date Detection** drop-down list and select a format:
 - Choose **Month Day Year** to have HotDocs assume the first set of numbers in an answer is the month while the second set is the day of the month.
 - Choose **Day Month Year** to have HotDocs assume the first set of numbers in an answer is the day of the month while the second set represents the month.

For HotDocs users in the United States, the default option is *Month Day Year*. For HotDocs users outside the U.S., the default option is *Day Month Year*.

To specify the format for the answer when it is merged into the assembled document, see Specify a Default Date Format.

Control How HotDocs Handles Two-Digit and Four-Digit Years

As a safety precaution, HotDocs requires you to enter four digits when entering the year portion of a date. If you enter a two-digit date, such as 7/9/99, HotDocs responds with this message: "You must enter a four-digit year."

You can override this default option by specifying a century rollover year at the HotDocs Options dialog box. The number you enter controls how HotDocs interprets two-digit years. Two-digit years less than or equal to the number you enter are understood as 2000-century years. Two-digit years greater than the number you enter are understood as 1900-century years.

For example, if you specify a rollover value of 34, dates entered as 5/14/34 will appear as 14 May 2034. A date entered as 5/14/35 will appear as 14 May 1935.

To set a century rollover preference

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. In the **Answer Entry** group, clear **Require four-digit year in dates**. The **Century rollover year** option becomes available, with the default rollover value of **50** selected.
- 4. Type a new two-digit number value in the **Century rollover year** field (or click the up or down arrows to select a number.) Dates you enter where the year is greater than this value will appear as *19??*, while dates where the year is less than this value will appear as *20??* (where *??* represents the year you type at the answer field.)

Change the Way Dates Appear in Answer Fields

As you enter dates during an interview, HotDocs reformats the date to appear a certain way in the answer field. You can change this format. (To change the format of dates in the assembled document, see Specify a Default Date Format.)

To change the way dates appear in an answer field

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Interviews and Dialogs** folder. The window changes to show several options relating to the way variables and dialogs appear and are processed during the interview.
- 3. At the **Answer Entry** group, select a date format from the **Date display format** drop-down list. Your options include **3 Jun 1990**; **June 3, 1990**; **6/15/1990**; and **15/6/1990**.

Whether the 6/15/1990 format or the 15/6/1990 format is available depends on your date detection format.

At a Glance: Appearance (HotDocs Options)

HotDocs Options				? ×
HotDocs Options HotDocs Options Interviews and Dialogs Appearance End of Interview Document Assembly Markup View (Word) HotDocs Models HotDocs Models Personal Information Spell Checking Personal Information Spell Checking Hot Docessors File Locations File Management File Management Plugins	Appearance Dialog font: Resource font: Outline font: Item color: Diak Dialog Title Caption Prompt: Answer To *Required to Additional to	(Default Font) (Default Font) (Default Font) og Title Background ext field ext	 Heading Answer Text Button 	
G	Resource 1	Text k for required fields		
B R	estore Default Fo	onts and Colors	OK	Cancel

After opening P HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \blacksquare .

If you double click on **Interviews and Dialogs** you will see a list below showing **Appearance** and **End of Interview**. Click on **Appearance** and the **Appearance Options** dialog box opens, from here you can customize the look of the interview.

At the top of this dialog \mathbf{E} there are three drop-down menus to change the font for the **Dialog**, **Resource** and **Outline**. To the right of each font field there is a number field to choose the text size.

Below these are two drop-down menus \square which you can use to change the color of items in the dialog pane. Find the item you wish to change in the first drop-down list then click the second menu to choose a color from the palette. You will see the changes in the example dialog pane below \square . Below the example dialog pane is a check box \square where you can choose to show an asterisk for required fields.

At the bottom of the page is the **Restore Default Fonts and Colors** button **F**. You can use this to return the all the settings on this page back to their original values.

To find out more about setting options for Appearance follow the link below:

• Customize the Look of the Dialog Pane

Customize the Look of the Dialog Pane

You can change the appearance of the dialog pane. Specifically, you can change background colors and text properties (such as font face, size, and color). You can also change some of the properties of the resource pane as well as the interview outline.

To restore all default properties after you have changed them, at the **Appearance** window, click **Restore Default Fonts and Colors**.

To change the appearance of the dialog, resource pane, or interview outline of the assembly window

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Interviews and Dialogs folder. The folder expands to show subfolders.
- 3. Click the **Appearance** subfolder. The window changes to show a visual representation of the dialog pane.
- 4. Make any changes, based on the following options:
 - To change the font or font size for text in the dialog, click the **Dialog font** drop-down button and choose the font from the list and enter a size in the **Font size** field.
 - To change the font or font size for text in the resource pane, click the **Resource font** drop-down button and choose the font from the list and enter a size in the **Font size** field.
 - To change the font or font size for text in the interview outline, click the **Outline font** drop-down button and choose the font from the list and enter a size in the **Font size** field. (To see these changes take effect, you must start a new assembly.)
 - To change the color of an item in the assembly window, click the **Item color** drop-down button and choose the item from the list. Then click the color drop-down button and choose the color.
 - To have HotDocs mark required questions with an asterisk, select **Show asterisk for** required fields .

In addition to choosing the item whose properties you want to change using the drop-down lists, you can also click on an area in the dialog preview. This automatically selects the item (in the corresponding drop-down list) so you can then change the properties.

At a Glance: End of Interview (HotDocs Options)

Interviews and Dialogs Appearance Ind of Interview Document Assembly Markup View (Word) HotDocs Models Ouestion and Answer Summaries Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins	End of Interview End of Interview Buttons G to the first unanswered question in the interview View the assembled form document at the Form Document tab Send the assembled form document to the word processor Send the assembled form document to the open word processor document Paste the assembled document into the open word processor document Copy the assembled document in a file Save the assembled document as a PDF Close this window without saving the assembled document Copy choose which buttons are displayed on the End of Interview dialog Finish Interview Action Send assembled text documents to the word processor and close Proceed to the Document Preview tab Send assembled form documents to Adobe Acrobat and close Proceed to the Form Document tab Component tab Send assembled form documents to Adobe Acrobat and close Proceed to the Form Document tab Component t
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

After opening $\mathbf{\overline{M}}$ HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you doubleclick on **Interviews and Dialogs** you will see a list below showing **Appearance** and **End of Interview**. Click on **End of Interview** and the **End of Interview Options** dialog box opens, from here you can decide how HotDocs will finish an interview and what buttons appear at the end of interview dialog.

The top set of check boxes **B** enable you to decide which buttons appear on the End of Interview dialog; you can choose as many or as few as you need.

Below that are two sets of check boxes; the top set \square allow you to choose what will automatically happen when you finish an interview for a text document and the bottom set \square allows you to choose what will automatically happen when you finish an interview for a form document.

The option to **Send assembled form document to Adobe Acrobat and close** does not work with .HFT templates as they are not PDF based templates. Selecting this option instead sends them to HotDocs Filler. If you need to send the assembled form to Adobe Acrobat you should save your .HFT templates as .HPT templates before choosing this option.

To find out more about setting options for End of Interview follow the links below:

• Customize the End of Interview Dialog

• Control What Happens When You Finish an Interview

Customize the End of Interview Dialog

Options in the *End of Interview* dialog allow you to work with assembled documents. Such options include sending documents to the word processor, saving them to disk, or closing the assembly window without doing anything. You can choose which of these options appear in the *End of Interview* dialog.

To choose which options appear in the End of Interview dialog

- 1. At the HotDocs library, click the **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Interviews and Dialogs folder. The folder expands to show subfolders.
- 3. Click the **End of Interview** subfolder. The window changes to show which buttons you can choose to have appear in the *End of Interview* dialog.
- 4. In the **End of Interview Buttons** group, choose which buttons you want on the *End of Interview* dialog by selecting the corresponding check box.

Buttons will be included only if they are selected and if they are relevant to the current interview. For example, HotDocs will display the **Go to the first unanswered question in the interview** button only if there is at least one unanswered question in the interview.

To customize the list of buttons while viewing the *End of Interview* dialog, you can right-click anywhere in the dialog and choose **Change Options** from the shortcut menu.

Control What Happens When You Finish an Interview

By default, when you click **Next** at the last dialog in an interview, HotDocs displays the **End of Interview** dialog, which gives you options for working with the assembled document. For example, you can choose to send the assembled document to the word processor or Filler, or you can return to the interview and answer any unanswered questions. However, if the **End of Interview** dialog is hidden (see Use the End of Interview Dialog), clicking **Next** simply displays the assembled document. Where you view the document, either in the assembly window or in the word processor or HotDocs Filler, depends on which options you select.

Likewise, when you click the Ei**Finish** button in the navigation bar, HotDocs displays the assembled document. Again, you can choose where you view the document.

To define how HotDocs should finish an interview

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Interviews and Dialogs folder. The folder expands to show subfolders.
- 3. Click the **End of Interview** subfolder. The window changes to show options for working with the *End of Interview* dialog.
- 4. In the **Finish Interview Action** group, select an option for text documents:
 - To send all assembled text documents directly to the word processor, select **Send** assembled text documents to the word processor and close.
 - To view all assembled text documents directly at the **Document Preview** tab of the assembly window, select **Proceed to the Document Preview tab**.
- 5. Select an option for form documents:
 - To send all assembled form documents directly to HotDocs Filler, select **Send assembled** form documents to HotDocs Filler and close .
 - To view all assembled form documents at the **Form Document** tab of the assembly window, select **Proceed to the Form Document tab**.

Now, whenever the **End of Interview** dialog is hidden and you click **Next** at the last dialog in the interview, HotDocs will perform the action you specified. Additionally, whenever you click the **Finish** button, the same action will be performed.

Document Assembly

At a Glance: Document Assembly (HotDocs Options)

HotDocs Options	? <mark>- ×</mark>
Interviews and Dialogs Interviews and Dialogs Imarkup View (Word) Imarkup View (Word)	toolbars wered warning before saving documents e document when closing assembly window e documents sent to word processor e printing assembled document via word processor raph mark added by the Document Text Editor swer file when beginning an assembly nt mat: June 3, 1990 able placeholder: *** Variable *** a at Document Preview tab Answered text Color: color: adding an assembly to the queue assemblies
	<u>OK</u> Cancel

After opening Reference opening After opening from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you click on **Document Assembly** you will see the **Document Assembly Options** dialog box, from here you can set options to decide how HotDocs assembles a document.

The first section **B** has seven check box options to help you customize the assembly window. By checking these you can choose to auto-arrange toolbars, display the unanswered warning before saving a document, prompt to save the document when closing the assembly window, prompt to save documents sent to a word processor, prompt before printing assembled documents via a word processor, explain the paragraph mark added by the document text editor, and/or prompt for an answer file when beginning an assembly.

The setting **Explain the paragraph mark added by the document text editor** refers to the final paragraph mark in the text editor shown when editing a SPAN. When ticked, this option displays an explanation of this paragraph mark when editing SPAN fields during an interview. Uncheck this box to suppress the dialog box containing that explanation.

In the next section vou can change settings to change the look of an assembled document. In the text field at the top the default date format is displayed. To change this you just need to delete the text and type in your preferred date format example. using the drop down list below you can chose what an unanswered variable placeholder will look like in the assembled document. Below that is a check box giving you the option to highlight fields at the **Document Preview** tab. At the end of this section are two

drop down lists that allow you to specify the color of different parts of the document. From the first drop down menu you chose a document field from: **Answered Text**, **Unanswered Text**, **Editable Text** or **Edited Text**. Then click the **Color** drop-down list and chose the color you would like this field type to be from the color palette.

The last section **D** has two check box options to change settings for the assembly queue. Checking the top box will have HotDocs prompt you before adding an assembly to the queue and checking the bottom box will have HotDocs produce a list of completed assemblies.

To find out more about setting options for Document Assembly follow the links below:

- Make Assembly Window Toolbars Automatically Wrap
- Display Warning When Questions are Unanswered
- Prompt to Save the Assembled Document When Closing the Assembly Window
- Warn When Printing Assembled Text Documents
- Prompt to Select an Answer File Before Assembly
- Specify a Default Date Format
- Format Unanswered Variables in a Document
- Set Properties for Viewing Answers and Editable Text in the Assembled Document
- Warn When Adding Assemblies to Assembly Queue
- Include List of Completed Assemblies in Assembly Queue

Make Assembly Window Toolbars Automatically Wrap

The HotDocs assembly window includes several toolbars to help you accomplish the work you need. At times, the width of the assembly window won't allow all of the toolbars to be displayed at once—some of the toolbars scroll off the right edge of the window. You can select an option that allows those toolbars to wrap to a new line of the assembly window so you can view all of them at once.

To auto-wrap assembly window toolbars

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the Document Assembly folder. The window changes to show several assembly options
- 3. Select Auto-arrange toolbars.

Display Warning When Questions are Unanswered

As you complete a HotDocs interview, HotDocs keeps track of which questions you have answered. You can have HotDocs display a warning message if you attempt to send the document to the word processor without answering one or more questions. (The warning also appears when you save or print the document.) Leaving questions in an interview unanswered may result in an incorrectly assembled document. Heeding these warnings can help you ensure the accuracy of the document.

To have HotDocs display a warning when variables are unanswered

- 1. At the HotDocs library, click the **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. Select **Display unanswered warning before saving documents**.

If you don't want HotDocs to warn you about unanswered questions, clear **Display unanswered warning before saving documents**. Unanswered questions will still be reported in the *End of Interview* dialog, but HotDocs will no longer display warning messages.

Prompt to Save the Assembled Document When Closing the Assembly Window

Whenever you close an assembly window, either before or after you send the assembled document to the word processor, HotDocs can ask if you want to save a copy of the assembled document.

To have HotDocs prompt you to save the assembled document

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. Select **Prompt to save document when closing assembly window.** HotDocs will prompt you to save a copy of the assembled document whenever you close the assembly window.
- 4. Optionally, select **Prompt to save documents sent to word processor** to have HotDocs prompt you to save a copy of the assembled document when you close the assembly window, even if you have sent the document to the word processor.

Depending on your project needs, you may not need to save electronic versions of your documents. In such cases, you can clear both of these options.

Warn When Printing Assembled Text Documents

When you print an assembled text document from the assembly window, HotDocs opens the document in the word processor, prints it using the default printing options, and then closes the word processor without ever displaying a **Print** dialog box.

Frequently, you may need to change the default printing options, which you can only do by sending the assembled document to the word processor and then by printing from the word processor's **File** menu. To help you remember this sequence of steps, you can have HotDocs warn you when it's about to print a document without displaying the **Print** dialog box. If you choose, you can stop the print job.

To display the Print warning

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. From the Assembly Window group, select Prompt before printing assembled document via word processor.

Prompt to Select an Answer File Before Assembly

You can have HotDocs prompt you to select an answer file when you assemble a document. When it does, you can choose to use an existing answer file; specify a new, untitled answer file; or use the answer file from your last assembly.

To have HotDocs prompt you for an answer file

- 1. At the HotDocs library, click the 🗹 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. Select **Prompt for answer file when beginning an assembly**.

Clear **Prompt for answer file when beginning an assembly** to have HotDocs bypass this option and start the interview immediately using an untitled answer file. Once the interview has started, you can save the current answer file at any time.

Specify a Default Date Format

You can specify a default format for dates that are merged in assembled documents. This default format is used if no format has been assigned by the template provider. It controls how dates are merged into the assembled document—not how dates are formatted for display during the interview. (See Change the Way Dates Appear in Answer Fields.)

To specify a default date format

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. Type a date format in the **Default date format** field. (Use one of the following example formats: **June 3, 1990; 3 June 1990; 03 JUN 90; 06/03/90; or 3rd day of June, 1990**.)

Format Unanswered Variables in a Document

You can specify how an unanswered variable will appear in an assembled text document. (If a variable is left unanswered in a form document, the form field is left empty, regardless of which unanswered variable placeholder you have selected.)

To specify an unanswered variable format

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. At the **Unanswered variable placeholder** drop-down list, make a selection based on the following information:
 - Select *****Variable***** to insert the name of the variable with three asterisks on both sides of the variable name.
 - Select [Variable] to insert the name of the variable between two brackets.
 - Select **Underscores** to insert a blank line where the answer should be.
 - Select Asterisks to insert three asterisks.
 - Select **Nothing** to insert nothing in the unanswered field, not even a space.

Sometimes a template developer may assign a placeholder to be used with a variable. If so, the document may show a placeholder that is different than what you select here.

If you are using Microsoft Word (and the template developer has enabled answer editing at the **Document Preview** tab), you can edit answers while viewing the document. See Edit Answers at the Document Preview Tab.

Set Properties for Viewing Answers and Editable Text in the Assembled Document

If you are using Microsoft Word, you can view answers and assembled document text that can be edited while viewing the **Document Preview** tab. (See Edit Answers at the Document Preview Tab for details.) When doing this, you can control whether the answers and text are marked by a special color, which you can also define.

Some template developers choose to disable these features in their templates. If the options for editing answers and text are not available at the assembly window, this is why.

To always highlight answers and editable text in a document

- 1. At the HotDocs library, click the **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. Select **Highlight fields at Document Preview tab**. This will mark answers in the document with the color you define at the **Color** drop-down. It will also mark editable document text as well as edited document text.
- 4. Optionally, to change the color for answer fields or sections of editable text, click the **Document field** drop-down button, and choose the appropriate option. Then click the **Color** drop-down button and choose a color.

To restore the default colors, choose the field type and then click the **Color** drop-down button. Choose **Default** from the top of the **Palette** tab.

Warn When Adding Assemblies to Assembly Queue

HotDocs will only allow you to assemble one document at a time. If you are currently assembling a document and you attempt to assemble another one, HotDocs will add it to the assembly queue and it won't be assembled until the current assembly is complete. If there are multiple assemblies in the queue, HotDocs will assemble them in the order they are added. (See Use the Assembly Queue.) You can have HotDocs warn you when you attempt to assemble a document and one is already being assembled.

To have HotDocs warn you when you add an assembly to the queue

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. In the Assembly Queue group, select Prompt before adding an assembly to the queue.

Include List of Completed Assemblies in Assembly Queue

As you assemble documents during a given session of HotDocs, you can have HotDocs create a list of each document you have assembled in the assembly queue. This list remains active during a particular session of HotDocs. (A session is the time between when you start HotDocs and when you close it.)

To list completed assemblies in the assembly queue

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Document Assembly** folder. The window changes to show several assembly options.
- 3. In the Assembly Queue group, select List completed assemblies.

If you are assembling large numbers of documents and you don't want HotDocs to track completed assemblies, clear **List completed assemblies**. Then, once assembly has finished, the assembly is removed from the queue.

Document Markup

At a Glance: Document Markup (HotDocs Options)

HotDocs Options						? <mark>×</mark>
HotDocs Options	Markup View (Word) Markers Answer field Insertion field Editable text Labels Use developer Use option tex Use developer Enclose text b Add Footnot	[[c comn ct to la c comn lock la]]]] hents to lat bel multiple bels in mar Numbers	Conditional text Alternative text Repeated text bel answer fields e choice answer fields bel beginning	[][[end of t	P X
	Attributes Bold Options Mark editable V Use minimal mark Apply attribute	It text b arkup es to a	alic locks at the Hotl answer field	Color: Regular Cocs library Preview ta ds when minimal marku	ib p is use K	ed Cancel

After opening HotDocs Options from the toolbar (or Tools menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you click on **Markup View (Word)** you will see the **Markup View (Word) Options** dialog box, from here you can set options for the look and function of a template in markup view.

In the first section of options **b** you can choose the markers used for **Answer Fields**, **Insertion Fields**, **Editable Text**, **Conditional Text**, **Alternative Text** and **Repeated Text** by typing the characters you would like to use into the corresponding fields.

In the next section \bigcirc you can set options for labels by ticking the check boxes. Using the first four check boxes you can choose to use developer comments to label answer fields, use option text to label multiple choice answer fields, use developer comments to label the beginning and/or the end of text blocks, and/or enclose text block labels in markers. The last check box has slightly more complicated options and ticking the last box will ungray the two drop-down menus next to it then you can use these to decide if, at the beginning and/ or end of text blocks, you would like **Footnote Letters**, **Footnote Numbers**. **Endnote Letters**.

In the next section **D** you can choose which attributes to apply to markup fields. Using the three check boxes you can choose to apply **Bold**, **Italic**, and/or a different **Color** scheme. Once the **Color** box has been ticked the drop-down list next to it ungrays, then you can chose from **Regular**, **Nested**, or **Sequential** color schemes. You can further customize these schemes at the Field Colors Options dialog.

The last section contains three check boxes with more general options for Markup View. You can choose to mark editable text blocks, use minimal markup at the HotDocs Library **Preview** tab, and/or apply attributes to answer fields when minimal markup is used.

To find out more about setting options for Markup View (Word) follow the link below:

• Specify How Documents Should Be Marked Up

Specify How Documents Should Be Marked Up

This feature is available to Word users only.

Moving between Markup View and Developer View is supported in Microsoft Word RTF templates only. In Microsoft Word DOCX templates you can click the ¹ Create Markup button to open a Markup document in a new Word window.

As you automate a Microsoft Word template and create within it variable and instruction fields, the template can be complicated to read and understand, especially to someone unfamiliar with automation. At times, however, you may need for a subject matter expert to review the text of the template for accuracy or to make changes or edits. To make the template easier for non-HotDocs users to review, you can provide them a marked up copy of the template.

How the template is marked up depends on your preferences. You can choose which fields HotDocs marks as well as define how much information is included in the field. For example, say you have a template with a series of IF expressions. You can assign comments to each IF Field describing, in layman's terms, the conditions surrounding the inclusion of the text in the document. Then, when you view the template in Markup View, the comments can be used to mark the expressions. The instructions themselves can then be viewed in an annotation.

In addition to viewing markup in the template, you can also view markup at the **Preview** tab of the template library and at the **Document Preview** tab of the assembly window. To view markup in either of these places, choose **Markup View** from the respective window's **View** menu.

To specify how documents should be marked

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Markup View (Word)** folder. The window changes to show several markup options.
- 3. Complete any of the tasks, described in the following table:

То

Do This

Choose the character or characters that should be used to mark fields in the	Enter the opening and closing characters in the fields next to the field type:
template	 Answers field = variable fields
	 Insertion field = INSERT instructions
	• Editable text = SPAN / END SPAN instructions
	 Conditional text = IF/ END IF instructions
	• Alternative text = ELSE IF / ELSE instructions
	• Repeated text = REPEAT / END REPEAT instructions
Mark variable answer fields with any comments entered for the variable	Select Use developer comments to label answer fields . If comments have been entered for a variable field, they will be merged in the markup field.
Use a Multiple Choice variable's merge text options as the label for its field	Select Use option text to label multiple choice answer fields.
If you've named your merge text, the name will be used instead.	
Mark text blocks (such as IF, REPEAT, and SPAN instructions) using	Select Use developer comments to label beginning of text blocks.
developer comments entered for the instruction	To merge the same comment in the closing markup field, select and end of text blocks .
Include both opening and	Select Enclose text block labels in markers.
closing markers when merging a field label (which includes developer comments)	Selecting this option causes HotDocs to use the closing marker after the opening label. The closing marker will also be used at the end of the text block.
By default, when HotDocs marks text blocks, an opening field marker is merged before the	In the following example, [* is the opening marker for conditional text while *] is the closing marker. The following is what will be merged in the marked up document:
opening instruction and	*Insert paragraph if employee must complete a trial period*1
the closing marker is	

Include text block instructions in either a footnote or endnote	Select Add Footnote/Endnote at beginning of text blocks. Then select whether the annotation will be marked using Letters or Numbers .
	To annotate the closing field label, select and end of text blocks .
Apply font attributes, such as bold and italics,	Select Bold or Italic .
to a markup field	Merge fields in the document are italicized by default. If you've used international characters in your answers, the italicized version of the font used in the document may not display the foreign characters correctly. If this is the case, clear the Italic attribute.
Define the color scheme used for marking the different fields in the template	Select Color and click the Color drop-down button. Then choose the color scheme.
Include editable text	Select Mark editable text blocks.
template in Markup View	Editable text blocks are not visible when viewing the markup of an assembled document.
	By default, HotDocs does not show SPAN instructions when you switch to Markup View.
Show only basic markup	Select Use minimal markup at the HotDocs library Preview tab.
template at the HotDocs library	When this option is selected, HotDocs will simply mark fields using field markers. If comments have been assigned to variable fields, those comments will be merged. (They will not be merged for instruction fields, however.)
	To assign font formatting and color to markup fields at the Preview tab, select Apply attributes to answer fields when minimal markup is used .

At a Glance: Field Colors (HotDocs Options)

	Field Colors				
Document Assembly	Click on the text in a cell t cell to change the backgro	to change the text colo ound color.	r. Click a	n the white area o	fa
Markup View (Word) Field Colors	B Regular	Nested		Sequential	*
HotDocs Models	Variable	Level 1		Field 1	
Form Documents	IF	Level 2		Field 2	
Question and Answer Summaries	ASK	Level 3		Field 3	
- Spell Checking	INSERT	Level 4	Ξ	Field 4	Ξ
Template Development	REPEAT	Level 5		Field 5	
Word Processors File Locations	SPAN	Level 6		Field 6	
	Comments	Level 7		Field 7	
Template Set Updates		Level 8		Field 8	
Plugins		Level 9		Field 9	-
~		Level 10		Field 10	-
		Level 11		Field 11	-
		Level 12	-	Field 12	-
		<	•	• III •	
			_		

After opening $\mathbf{\overline{M}}$ HotDocs Options from the toolbar (or Tools menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you double click on **Markup View (Word)** you will see a list below showing **Field Colors**. Click on **Field Colors** and the **Field Colors Options** dialog box opens, from here you can set the colors HotDocs will use in Markup view.

There are three lists **B** for the three of available color schemes. You can choose which color scheme you would like to use on the Markup View (Word) Options dialog. By clicking on any of the words you will see a color palette where you can choose a different text color for that item. By clicking on the white space behind any of the words you will see a color palette where you can choose a different background color for that item.

To find out more about setting options for Field Colors follow the link below:

• Define Field Colors

Define Field Colors

By default, when you insert fields in a HotDocs text template, HotDocs applies certain colors to the fields, depending on the field type. For example, all variable fields will be merged using a specific color, while all REPEAT instructions will be merged using a different color, and so forth.

Depending on the complexity of the template, there may be times when you want to use field coloring to view relationships between SPAN, IF and REPEAT instructions in the template. For example, when working with nested instructions, you may want all first-level instructions to be marked by one color, all second-level instructions to be marked by another color, and so forth. Additionally, you may want to view instructions in the template sequentially, meaning each instruction pair is marked with its own color.

At the **Field Colors** page, you can assign the colors HotDocs will use for marking fields and instructions in a template.

To customize the field colors in a text template

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Markup View (Word) folder. The folder expands to show subfolders.
- 3. Click the **Field Colors** folder. The window changes to show lists of field types and instructions with their corresponding colors.
- 4. Complete any of the following steps:
 - To specify the color for all field types in a template, click that particular field in the **Regular** column and choose the color you want from the palette.
 - To specify the color for each nested level of instruction, click that particular level in the **Nested** column and choose the color you want from the palette.
 - To specify the color for each matching pair of instructions, click that field in the **Sequential** column and choose the color you want from the palette.
- 5. Repeat this process for each field type or level of instruction you want to change.

Once you have made these changes, you must click the **Apply Colors** button within your template to apply the new colors to any existing fields.

HotDocs allows you to assign up to 20 Nested and Sequential colors. If you want to use fewer color schemes, select the Level or Field after the last color you want used and assign black as the text color and white as the highlight color. (So, for example, if you want to show only four levels of nesting, click on Level 5 and assign the black/white color combination.)

If using the keyboard, tab to the row you need to access and press the **Spacebar** to show the foreground color palette. Press **Shift+Spacebar** to show the background color palette.

To restore default colors, click the **Color** drop-down button and choose **Default** at the top of the **Palette** tab.

To choose the colors used for form fields, see Change Colors in Form Fields.

HotDocs Models

At a Glance: HotDocs Models (HotDocs Options)

HotDocs Options		? 💌
B There is a second balages The Document Assembly The Markup View (Word) The Docs Models The Docs Model	HotDocs Models	
	OK Car	ncel

After opening HotDocs Options from the toolbar (or Tools menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you click on **HotDocs Markup** you will see the **HotDocs Markup Options** dialog box, from here you can decide if you want to enforce HotDocs Model naming standards by checking the box . Unchecking this box removes the requirement that component names in HotDocs Models not contain spaces or other special characters. (Names should only contain letters, numbers, and underscores.) If names in your automation contain these non-standard characters, selecting this option will cause HotDocs to display error messages when you either assemble a HotDocs Model, or when you create a template from a HotDocs Model.

To find out more about setting options for HotDocs Markup follow the link below:

• Allow Non-Standard Names in HotDocs Markup

Allow Non-Standard Names in HotDocs Markup

When marking up HotDocs Models, you must adhere to the HotDocs Markup rules when naming variables. For example, variable names can include letters, numbers, and underscores, but they cannot include spaces or other special characters.

Because of this requirement, you may receive errors either when you convert templates to models where spaces and non-alphanumeric characters are used in a variable name, or when you assemble a HotDocs Model that use this naming scheme. If your HotDocs Model contains these non-standard variable names, you can select a HotDocs Option that allows HotDocs to still convert or assemble a model, regardless of how you've named your variables.

To allow spaces and other non-alphanumeric characters in variable names

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **HotDocs Markups** folder. The window changes to show options for working with HotDocs Markup.
- 3. Clear Enforce HotDocs Markup naming standards.

Currently in HotDocs, template developers can include spaces and non-alphanumeric characters in variable names. In some future version, however, HotDocs may require users to follow a newer, more standard practice of naming variables that includes using only letters, numbers, and underscores.

See HotDocs Model Naming Standards for more details.

Form Documents

At a Glance: Form Documents (HotDocs Options)

HotDocs Options	? <mark>- × ·</mark>
 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins 	 Form Documents Measurements: Inches Color theme: HotDocs 6 Default Send assembled PDF-based forms to: Adobe Acrobat Send assembled PDF-based forms to: Adobe Acrobat On Open Highlight Fields Zoom: 100% Show Variables Show Thumbnails While Filling Warn when leaving a field that overflows Warn when leaving a field that overflows Warn when overriding a field Warn when overriding a field Warn when attempting to create a field using the Fill tool Printing Remember the printer used for forms Print PDF-based forms using: Always Ask Recovery (HotDocs Automator only) Create backup copies of original templates Maintain auto-recovery files while editing templates

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you click on **Form Documents** you will see the **Form Documents Options** dialog box, from here you can set options to decide how new form documents appear and function.

This section of options **B** at the top of this dialog allow you to choose some of the main setting for new form documents. Using the three drop-down lists you can select the type of measurements used in the document, which color theme to use for HotDocs Automator; you can choose whether to automatically send PDF-based forms to Adobe Acrobat or HotDocs Filler.

The option to **Send Assembled PDF-based forms to Adobe Acrobat** does not work with .HFT templates as they are not PDF based. If you need to send the assembled form to Adobe Acrobat you should save your .HFT templates as .HPT templates before choosing this option.

The next section C allows you to set options on how the form document first appears upon opening. You can use the three check boxes to decide if you would like HotDocs to highlight fields, show variables, and/or show thumbnails. Using the drop-down list to the right you can also decide on the initial zoom level for the form.

In the next section **D** you can choose which warnings HotDocs can display when filling. You can have HotDocs warn when leaving a field that has overflowed, warn when typing in a restricted field, warn when overriding a field, and/or warn when attempting to create a field using the fill tool.

In the section below gou can set two options for printing form documents. You can tick the check box to have HotDocs remember the printer used for forms and you can choose from the drop-down list whether you use **HotDocs**, **Adobe Acrobat**, or HotDocs should **Always Ask** which program to use to print PDF-based forms.

In the last section **F** there are two check box options that allow you to set options for recovery in HotDocs Automator. Check the first box to have HotDocs create backup copies of the original templates and check the second box to have HotDocs maintain auto-recovery files while you are editing templates.

To find out more about setting options for Form Documents follow the links below:

- Change the Units of Measurement for Forms
- Change Colors in HotDocs Forms
- Send Assembled PDF Documents to Adobe
- Control How Forms Appear When Opened
- Control Warnings During Assembly
- Save Printer Information for Printing Forms
- Specify How PDF-based Forms Should Be Printed
- Protect Forms By Backing Them Up

Change the Units of Measurement for Forms

HotDocs Automator measures the fields you create to identify whether they should be treated as checkbox fields or edit fields. In addition, the units of measurement affect the margins around all fields and the amount of space used by the header and footer sections of the addendum.

To change the units of measurement

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Form Documents** folder. The right pane changes to show options for working with forms.
- 3. Click the **Show measurements in** drop-down button and select **Inches**, **Points**, **Centimeters**, **Picas**, or **Millimeters**.

Change Colors in HotDocs Forms

You can change the colors for field backgrounds and borders in HotDocs Automator . Changes affect all forms edited or assembled using HotDocs. There are several custom color schemes from which you can choose.

To change colors

- 1. At the HotDocs library, click the **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Form Documents** folder. The view changes to show general form options.
- 3. Click the **Color theme** drop-down button and select a color scheme.

To change the colors used for text template fields, see **Define Colors for Fields and Instructions**.

Send Assembled PDF Documents to Adobe

You can select an option that allows you send an assembled PDF-based document to Adobe Acrobat. When this option is selected, you can click the **Send Document to Adobe Acrobat** button and HotDocs will send the form document to Acrobat rather than HotDocs Filler. (If you want to view the assembled form document in Filler, you can still choose this option, too.)

To send assembled PDF-based documents to Adobe

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Form Documents** folder. The right pane changes to show options for working with forms.
- 3. Click Send assembled PDF-based forms to drop-down button and choose Adobe Acrobat.

To always send PDF-based forms to HotDocs Filler, click **Send assembled PDF-based forms to** and choose **HotDocs Filler**.

Control How Forms Appear When Opened

When you open a form in HotDocs Automator, different views of the form can be shown, such as fields, variable names in fields, and thumbnail views of form pages. You can control which views appear, as well as what level of magnification is used. These settings are used each time you open a form.
To control how forms appear when opened

- 1. At the HotDocs library, click the **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Form Documents** folder. The right pane changes to show options for working with forms.
- 3. In the **On Open** group, select options that control the view when you open a form, as described in the following table:

То	Do This
Display the colored fields	Select Highlight Fields.
Display the names of variables attached to each field	Click Show Variables.
Display thumbnail images of each page of	Click Show Thumbnails.
the form	The thumbnails appear in a separate pane in the window. You can click on the thumbnail images to move through the form.
Display forms at actual size.	Click Zoom and select 100%
Match form width to the left and right edges of the window	Click Zoom and select Fit to Width .
Matches form height to the top and bottom edges of the window	Click Zoom and select Fit to Height .

You can hide the fields, variables, and thumbnails by clearing these options.

To change the colors used for form fields, see Change Colors in HotDocs Forms.

Control Warnings During Assembly

When you assemble form documents, situations may arise where your answers may be altered. These situations include when answers overflow the field, or when field settings are different. HotDocs can warn you in such situations.

To control warnings that display during assembly

1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.

- 2. Click the **Form Documents** folder. The right pane changes to show several options for working with forms.
- 3. In the **While Filling** group, make selections based on the following information:
 - Select **Warn when leaving a field that overflows** to display the **Overflow Status** dialog box when an answer needs more space than the field allows.
 - Select **Warn when typing in a restricted field** to display the **Restricted Field** dialog box when you type in a field that can't be answered until other fields have valid answers.
 - Select **Warn when overriding a field** to display the **Override Field** dialog box when you choose to override a field.
 - Select **Warn when attempting to create a field using the Fill tool** to display a message explaining how to create a field when viewing the **Form Document** tab of the assembly window.

You can choose to have no warnings appear by clearing these options. This will not automatically resolve answer overflow or allow you to type answers in inactive fields. It only means no warnings will be displayed during the interview or direct-fill. When you attempt to save or print the document, you must resolve the overflow.

Save Printer Information for Printing Forms

Forms may need to be printed using a specific printer. You can have HotDocs remember which printer you use to print form documents, saving you the task of manually selecting the printer each time, especially when the printer you use for forms is different than your default printer.

To save information about the printer you use to print forms

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Form Documents** folder. The right pane changes to show several options for working with forms.
- 3. From the **Printing** group, select **Remember the printer used for forms**.
- 4. Print a form document, selecting the printer you want to associate with forms.

You can have HotDocs *not* remember printer information by clearing this option. This means HotDocs will suggest the default printer you have selected for *all* printing.

Specify How PDF-based Forms Should Be Printed

When you print an assembled PDF-based form document, you can choose the program that controls the printing process—either HotDocs or Adobe Acrobat.

To specify how PDF-based form documents should always be printed

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the **Form Documents** folder. The folder expands to show several sub-folders, and the right pane changes to show several options for working with forms.
- 3. At the **Print PDF-based forms using** drop-down list, do one of the following:
 - Select **Always Ask** to have HotDocs prompt you for your choice each time you print a PDF form.
 - Select **HotDocs** to always print the assembled document using HotDocs' printing technology.
 - Select Adobe Acrobat to always print the document using Adobe's printing technology.

Click here for tips on which printing option you should choose.

Protect Forms By Backing Them Up

In HotDocs Automator, you can choose safety precautions to protect your work from system failure.

To protect your work from system failure

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Form Documents** folder. The right pane changes to show several options for working with forms.
- 3. From the **Recovery (HotDocs Automator only)** group, make selections based on the following information:
 - Select **Create backup copies of original templates** to have HotDocs create a backup file when you open a form for editing in HotDocs Automator. If ever you need to revert to the file as it was when you created it, you can open the backup file and save it using the original file name.
 - Select **Maintain auto-recovery files while editing templates** to have Automator create a log file that lists all changes made directly in the form. If there is a system failure while you are working, the next time you open the file, it will use the log file to reapply all changes made to the form, so you won't lose any work. (The log file, however, doesn't back up the component file.)

Backup files are created in the same folder as the original file. If the folder is write-protected, the backup file is created in the temporary folder defined by your TMP or TEMP system variable.

The backup file has the same name as the original file but with the extension ~FT, or ~PT. (If, for some reason, the backup file can't be created with that name, it will have a number instead of the F or P in the extension—for example, ~2T.) Each time you open the original file, its current content will be saved to the backup file, overwriting the previous content.

At a Glance: Properties for New Fields (HotDocs Options)

HotDocs Options		? 💌
Pr Document Assembly Markup View (Word) HotDocs Models Form Documents Properties for New Fields Properties for New Forms Question and Answer Summaries Personal Information Spell Checking Properties Sor New Forms File Locations File Locations File Management Penplate Set Updates Plugins	operties for New Fields Edit Fields Alignment / Format Horizontal alignment: Left Vertical alignment: Bottom Set to Top for multi-line fields Lines per inch: 0.0 Max lines 1 for single-line fields Check-Box Fields Check-Box Fields Check-box detection threshold: 0.19* Character: X Minimum size: 3.0 pt Alignment Horizontal alignment: Center Vertical alignment: Bottom	Font Margins Left: 0.01" (*) Right: 0.01" (*) Top: 0.01" (*) Bottom: 0.03" (*) Bottom: 0.03" (*) Right: 0.00" (*) Right: 0.00" (*) Right: 0.00" (*) Bottom: 0.01" (*)
Template Development Word Processors File Locations File Management Template Set Updates Plugins	Check-Box Fields Check-box detection threshold: 0.19" - Character: X Minimum size: 3.0 pt - Alignment Horizontal alignment: Center - Vertical alignment: Bottom -	Eont Margins Left: 0.00" Right: 0.00" Top: 0.00" Bottom: 0.01" Cancel

After opening P HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \blacksquare .

If you double click on **Form Documents** you will see a list open below containing a file for setting the options for the **Properties for New Fields** and the **Properties for New Forms**. Click on **Properties for New Fields** to bring up the **Properties for New Fields** dialog box.

There are two main sections on this dialog. The top section **B** allows you to set options for new edit fields. You can set the horizontal and vertical alignment of text within the field by selecting from the two dropdown menus on the left side of this section. Below this there are three further options for the field alignment and format. Check the first box to have HotDocs automatically set the vertical alignment to **Top** for multi-line fields, you can then set how many lines per inch you would like to type in the multi-line

fields and check the box below to keep the maximum number of lines to 1 for single line fields. To the right of these options you can set the size of the border between the edge of the field and the text by entering measurements into the four fields. At the top right of this section you can set the font for new edit fields by clicking the **Font** button.

The second section in this dialog allows you to set options for new check box fields. In the first number field you can set the field size below which HotDocs will automatically turn the field into a check box. Then you can choose what character HotDocs uses to check the field by typing it into the **Character** field and the minimum size that character can be. Below these options you can set the horizontal and vertical alignment of character within the field by selecting from the two drop-down menus. To the right of these options you can set the size of the border between the edge of the field and the text by entering measurements into the four fields. At the top right of this section you can set the font for new edit fields by clicking the **Font** button.

To find out more about setting options for New Fields follow the links below:

- Set Properties for New Edit Fields
- Set Properties for New check box Fields

Set Properties for New Edit Fields

You can specify the default properties that will be used each time you create an edit field in a HotDocs form. You can control font sizes, text alignment and spacing, and field margins.

To set the properties for all new edit fields

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Form Documents folder. The folder expands to show several subfolders.
- 3. Open the **Properties for New Fields** folder. The window changes to show options that control how edit fields appear.
- 4. Make changes in the Edit Fields group, based on information in the following table:

То	Do This
Set font properties for text in edit fields	Click Font and make your changes at the Font dialog box.
Set the horizontal alignment for text in edit fields	At the Horizontal alignment drop-down list, select Left , Right , Center , or Justified .
Set the vertical alignment for text in edit fields	At the Vertical alignment drop-down list, select Left , Right , Center , or Justified .

	Vertical alignment is only apparent in multiple- line fields.
Align the text in multi-line fields with the top border of the field	Select Set to Top for multi-line fields.
Change the distance between lines of text in a multiple-line field	At the Lines per inch field, type a number (or click the up or down arrows to select a number).
Keep the user from manually entering multiple lines of text in a single-line answer field	Select Max lines 1 for single-line fields.
Changes the distance between the text and the edges of the field	At the Margins group, type a number (or click the up or down arrows to select a number) for the left, right, top, and bottom margins.

The changes you make are not applied to existing fields. The changes won't take effect until you create a new field. To change the appearance of existing edit fields, right-click on a field in the form and select **Field Properties** from the shortcut menu.

Some changes you make to these edit field properties may not affect specific fields because of settings assigned to these fields. For example, if the **Max Lines** setting for a field is set to 1, the **Lines per inch** property will have no effect—this property only affects multiple lines of text, but the **Max lines** setting limits the field to one line of text.

Set Properties for New check box Fields

When you create a field in a form template or document, HotDocs checks the size to see if it should be a check-box field. You can adjust the measurement that is used to distinguish check-box fields from edit fields. If a new field is identified as an edit field, HotDocs assigns standard formatting to the field that controls how answers appear in the field. You can also assign other default properties of a check-box field, such as assign the character that will be used when a field is selected, as well as set the alignment and margins of a field.

To set the properties for all new check box fields

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the **Form Documents** folder. The folder expands to show several subfolders.
- 3. Click the **Properties for New Fields** folder. The window changes to show field detection options that affect what kind of fields are created.
- 4. Make changes in the **check box Fields** group, based on information in the following table:

То	Do This
Set font properties for text in check box fields	Click the Font button, then make changes at the Font dialog box.
Sets the minimum size for edit fields so that all fields created smaller than this size will be	Enter a number in the Check-box detection threshold box.
created as check-box fields.	If you specify 0.00 as the value of this box, no new fields will be created as check-box fields.
Specify the character that will be inserted in a check box	Enter the character in the Character field.
Specify the smallest size that will be used for the check box character	Enter a font size in the Minimum size field.
Set the horizontal alignment for text in check-box fields	At the Horizontal alignment drop-down list, select Left , Right , Center , or Justified .
Set the vertical alignment for text in check-box fields	At the Vertical alignment drop-down list, select Left , Right , Center , or Justified .
Change the distance between the check box character and the edges of the field	At the Margins group, type a number (or click the up or down arrows to select a number) for the left, right, top, and bottom margins.

The changes you make are not applied to existing fields. The changes won't take effect until you create a new field. To change the appearance of existing check box fields, right-click on a field in the form and select **Properties** from the shortcut menu.

At a Glance: Overflow Properties (HotDocs Options)

Setting HotDocs Options

Overflow Properties	HotDocs Options	
 a Interviews and Deladogs b Document Assembly b Markup View (Word) c HotDocs Models c Form Documents c Properties for New Fields c Overflow Properties c Overflow Properties<td> Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Properties for New Fields Properties for New Forms Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins </td><td>Overflow Properties When field overflows Shrink answer as needed to 8.0 points Send answer to addendum Split multi-line answer between form field and addendum When sending answers from table to addendum Send entire table Send complete rows Send individual cells Addendum entry options Cross-reference text: Addendum label text: Table answer column indent: 0.33*</td>	 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Properties for New Fields Properties for New Forms Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins 	Overflow Properties When field overflows Shrink answer as needed to 8.0 points Send answer to addendum Split multi-line answer between form field and addendum When sending answers from table to addendum Send entire table Send complete rows Send individual cells Addendum entry options Cross-reference text: Addendum label text: Table answer column indent: 0.33*

After opening P HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \blacksquare .

If you double click on **Form Documents** you will see a list open below containing a file for setting the options for the **Properties for New Fields** and the **Properties for New Forms**. Double click on **Properties for New Fields** to see **Overflow Properties** appear in a list item below. Click on **Overflow Properties** to bring up the **Overflow Properties Options** dialog box.

The top set of options **B** allow you to decide what will happen when a field overflows. You can choose to have HotDocs shrink the answer to the font size of your choosing (you can change this by entering a number into the points field), send the answer to an addendum sheet, and/or spilt multi-line answers between the form field and the addendum.

The next set of check boxes C allows you to choose what HotDocs will do when sending answers from a table to the addendum. If the table overflows, you can choose to send the entire table to the addendum, send complete overflow rows to the addendum, or send individual overflow cells to the addendum.

The last set of options on this dialog **D** allows you to set options for the Addendum entries. In the first text field you can enter the text you would like HotDocs to place on the form as a cross reference to overflow answers in the addendum. In the second text field you can enter the label you would like to appear in the addendum before an entry and in the number field below you can choose how much space you would like between the label and the entry.

To find out more about setting options for Overflow Properties follow the links below:

- Set Rules for Handling Answer Overflow
- Specify Addendum References and Labels for Answer Overflow

Set Rules for Handling Answer Overflow

An answer overflows when it takes more space than is available in the answer field of a form document. This often requires that some or all of the answer be sent to the addendum. You can set rules that automatically control what happens when answers overflow.

To control what happens when answers overflow

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Form Documents folder. The folder expands to show several subfolders.
- 3. Open the **Properties for New Fields** folder, then select the **Overflow Properties** subfolder. The window changes to show options that control answer overflow.
- 4. Make selections for handling edit field overflow, based on the following options:
 - Select **Shrink answer as needed to** and then enter the point size in the **points** field. HotDocs will attempt to reduce the answer to this font size before warning of overflow.
 - Select **Send answer to addendum** to have HotDocs automatically resolve all overflow by sending answers to the addendum.
 - Select **Split multi-line answer between form field and addendum** to have HotDocs send only the part of the answer that overflows in a multi-line field to the addendum.
- 5. Make selections for handling table overflow, based on the following options:
 - Select Send entire table to send all answers in the table to the addendum.
 - Select **Send complete rows** to send answers from each row that contains a cell which overflows to the addendum.
 - Select **Send individual cells** to send answers from individual cells that overflow to the addendum.

The changes you make are not applied to existing fields. The changes won't take effect until you create a new field.

You can customize the reference that appears in the overflowed answer field. This can help you determine where the answer is located in the addendum. (See Specify Addendum References and Labels for Answer Overflow.)

You can remove the rules that automatically handle answer overflow by clearing these options. This will require you to manually resolve each instance of answer overflow.

Specify Addendum References and Labels for Answer Overflow

In a form document, when an answer overflows its answer field, the entire answer can be sent to an addendum. When this happens, HotDocs inserts text into the answer field, cross referencing you to the addendum for the full answer. You can specify what text is used in creating the cross reference as well as specify the text that identifies the answer once it has been sent to the addendum.

To control what information appears in overflow references and in the addendum

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the Form Documents folder. The folder expands to show several subfolders.
- 3. Open the **Properties for New Fields** folder, then select the **Overflow Properties** subfolder. The window changes to show options that control answer overflow.
- 4. In the **Cross-reference text** field, enter the cross-reference text you want to merge in the field if the answer overflows. (See Use Answer Overflow and Addendum Text Codes.)
- 5. In the **Addendum label text** field, enter the text you want to appear as the label for the entry in the addendum.
- At the Table answer column indent field, enter a number (or click the up or down arrows to select a number) to control the distance between the label for that specific entry in the list of answers and the actual answer.

The changes you make are not applied to existing fields. The changes won't take effect until you create a new field.

If you leave the **Cross-reference text** field empty, HotDocs uses a default reference, *See # in Addendum*.

At a Glance: Properties for New Forms (HotDocs Options)

HotDocs Options
 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Properties for New Forms Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations Plugins
OK Cancel

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you double click on **Form Documents** you will see a list open below containing a file for setting the options for the **Properties for New Fields** and the **Properties for New Forms**. Click on **Properties for New Forms** to bring up the **Properties for New Forms** dialog box.

At the top of this dialog is a check box **B** that allows you to choose if you would like HotDocs to generate a default interview from form fields.

Below that are three drop-down menus C that allow you to set the printing options for new forms. Using the first drop-down list you can choose to print **Single-sided**, **Double-sided**, **side to side** or **Double-sided**, **top to bottom**. In the middle drop-down list you can select a default paper size from a long list of options and in the last drop-down list you can choose a default paper source from a list of options.

To find out more about setting options for New Forms follow the links below:

- Automatically Generate Default Interviews for Form Templates
- Set Paper Size and Paper Source Defaults for Printing Forms

Automatically Generate Default Interviews for Form Templates

By default, when you create a new form template, the only option for assembling it is by entering your answers directly in the fields on the form. You can, however, specify an option that causes HotDocs to generate a default interview for the template, based on variables used in the template. If variables are grouped in custom dialogs, HotDocs will display the dialog instead. (This is how interviews are created for text templates.)

To have HotDocs always create default interviews for all new form templates

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the **Form Documents** folder. The folder expands to show several subfolders.
- 3. Click the Properties for New Forms folder.
- 4. Select Generate default interview from form fields.

The changes you make are not applied to existing templates. The changes won't take effect until you create a new template.

You can generate default interviews on a template-by-template basis (see Have HotDocs Generate a Default Interview), or even create custom interviews for your form templates (see Define a Custom Interview).

You can prevent users from using the interview outline by clearing this option. This will force users to directly fill the document at the **Form Document** tab.

Set Paper Size and Paper Source Defaults for Printing Forms

You can specify the paper size and paper source to be used when forms are printed.

To set paper size and paper source properties for a form

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the **Form Documents** folder. The folder expands to show several subfolders.
- 3. Click the **Properties for New Forms** folder. The window changes to show options that control the properties of newly created forms.
- 4. At the **Sides** drop-down list, select one of the available options:
 - Choose **Single-sided** to print the form on one side of the paper.

- Choose **Double-sided**, **side-to-side** to print the form on both sides of the paper, so consecutive pages read like a book.
- Choose **Double-sided**, **top-to-bottom** to print the form on both sides of the paper, so consecutive pages read like a flip-chart.
- 5. At the **Paper Size** drop-down list, select one of the available options.
- 6. At the **Paper Source** drop-down list, select one of the available options.

Paper size and source settings are recommendations that will be passed on to the printer if possible. If printing via acrobat then paper size, paper source and duplex options will revert to your defaults.

The changes you make are not applied to existing templates. The changes won't take effect until you create a new template.

You can use command-line options to specify these printing properties. See the Paper Tray and Paper Size command-line options.

At a Glance: Addendum Properties (HotDocs Options)

HotDocs Options	
Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Properties for New Fields Addendum Properties Question and Answer Summaries Personal Information Spell Checking Word Processors	B Addendum Properties Page Margins Left: 1.00°
	Font Height: 0.33" 🛬 Alignment: Right 💌
A	Font Indent: 0.33" - Spacing: 0.00"
	Number lines to format as pleading paper
	OK Cancel

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you double click on **Form Documents** you will see a list open below containing a file for setting the options for the **Properties for New Fields** and the **Properties for New Forms**. Double click on **Properties for New Forms** to see **Addendum Properties** appear in a list item below. Click on **Addendum Properties** to bring up the **Addendum Properties Options** dialog box.

There are four sections of options you can set here to change how the generated addendum appears. At the top of the dialog **B** you can set the width of the page margins in the four number fields; **Left**, **Right**, **Top** and **Bottom**.

The next set of options control how the addendum header will appear. In the large text field you can enter the text you would like to appear as the addendum header. Leave the field blank if you do not want a header. Below that there are three more options for the addendum header. You can select the font you would like the header to use by clicking the **Font...** button, you can select the height of the header by entering a number in the **Height** field and you can choose the alignment of the text from the **Alignment** drop-down list. You can choose from **Left, Center, Right** or **Justified**.

Below the header options are a set of options that control the appearance of the addendum footer **D**. These are identical to the options for the addendum header but they control the text at the bottom of every page of the addendum. If you do not want a footer you can leave the text field blank.

The last set of options on this dialog control the look of the addendum entries. You can choose the font you would like to use by clicking the **Font...** button, you can choose how much of an indent you would like between the label and the entry by entering a measurement in the central number field, and you can choose how much space you would like between sets of entries by entering a measurement in the right hand number field. Below these options there is a check box so you can decide if you would like to number the lines of entries to format the addendum as pleading paper.

To find out more about setting options for Addendum Properties follow the link below:

• Customize the Appearance of Addenda

Customize the Appearance of Addenda

You can format the form addendum to fit your project requirements.

To change the addendum's appearance

1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.

- 2. Double-click the **Form Documents** folder. The folder expands to show several subfolders.
- 3. Double-click the **Properties for New Forms** folder, then select the **Addendum Properties** subfolder. The window changes to show options that control the format of the addendum.
- 4. In the **Page Margins** group, enter margin widths (or click the up or down arrows to select a number). These numbers control the spacing between the answer text and the edges of the field.
- 5. In the **Header Properties** and **Footer Properties** groups, specify the header or footer text you want to appear in the addendum:
 - At the **Header text** or **Footer text** field, type the text you want displayed in the addendum header or footer. (See Use Answer Overflow and Addendum Text Codes.)
 - At the **Height** field, type a number (or click the up or down arrows to select a number). This number controls the vertical height of the header section.
 - Click **Font** to display the **Font** dialog box, where you can change font properties of the header text.
 - At the Alignment drop-down list, select one of the available options: Left, Center, Right, or Justified.
- 6. In the **Addendum Entries** group, make the following changes:
 - To change the font properties of the addendum entries, click **Font** to display the **Font** dialog box.
 - To control the distance between the label and the actual answer, enter a number (or click the up or down arrows to select a number) in the **Indent** field.
 - To control the distance between answer tables in the addendum, enter a number (or click the up or down arrows to select a number) in the **Spacing** field.
- 7. Select **Number lines to format as pleading paper** to place lines on the right and left side of the addendum, with numbers along the entire left edge.

The changes you make are not applied to existing forms. The changes won't take effect until you create a new template.

Question and Answer Summaries

At a Glance: Question and Answer Summaries (HotDocs Options)

Setting HotDocs Options

HotDocs Options	? 💌
 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Porm Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Locations File Management Template Set Updates Plugins Anne Percent of width taken by prompt: 30 Question Summaries Honor IF instructions when answered 	
	icel

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you click on **Question and Answer Summaries** you will see the **Question and Answer Summaries Options** dialog box, from here you can set options to decide how question and answer summaries are asked and displayed.

In the first section of options **B** you can choose the format for Question and Answer Summaries in a drop-down list. If you select **One Column Format** then the next section of options **C** becomes available and the third section **D** is grayed out. If you select **Two Column Format** the second section **C** will be grayed out and the third section **D** will become available. The other option in the first section **B** allows you to decide the maximum number of multiple choice options that will be displayed.

If you have chosen **One Column Format** in the first section \underline{B} of the options then the four check boxes \underline{C} allow you to further customize the look of the summaries. Here you can choose to indent the prompts displayed under dialog titles, precede the prompts with a bullet point, indent answers under prompts and precede answers with a bullet point.

If you have chosen **Two Column Format** in the first section **B** of the options then the drop-down list and number field **D** allow you set more options for your summaries. In the drop-down list you can choose how you would like the column borders to look. The choices are **None**, **Plain** or **Sculptured**. In the number field you can set the percentage of column width you would like taken up by the prompt.

The last option on the dialog is a check box that allows you to decide whether you would like HotDocs to honor IF instructions in question summaries when they're answered.

To find out more about setting options for Question and Answer Summaries follow the links below:

- Change the Way Question and Answer Summaries Appear
- Control Which Variables are Asked in a Question Summary

Change the Way Question and Answer Summaries Appear

In question and answer summaries, you can specify a one-column or a two-column format:

- The one-column format displays a simple list of variable prompts and answers (if you're generating an answer summary) or empty answer fields (if you're generating a question summary). Prompts and answers are grouped by dialogs. To make these summaries easier to read, you can indent these lists or use bullets.
- The two-column format displays a list of variable prompts in one column and their associated answers (if you're generating an answer summary) or empty answer fields (if you're generating a question summary) in a second column. Prompts and answers are grouped by dialogs. You can control the appearance of these columns by specifying borders and choosing the width of the column that contains the prompt.

To format question or answer summaries

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Question and Answer Summaries** folder. The right pane changes to show the summary options.
- 3. If you want the summary to appear in one-column format, click the **Format** drop-down button, choose **One-Column**, and select from the following options:
 - Select **Indent prompts under dialog titles** to include a tab between the column's margin and the prompt. Selecting this option makes the dialog titles and variable prompts appear hierarchical.
 - Select **Precede prompts with bullet** to insert a bullet character next to each prompt.
 - Select **Indent answers under prompts** to include a tab between the column's margin and the answer. Selecting this option makes the prompts and the answers appear hierarchical.
 - Select **Precede answers with bullet** to insert a bullet character next to each answer.
- 4. If you want the summary to appear in two-column format, click the **Format** drop-down button, choose **Two-Column**, and select from the following options:

- Select a border style from the **Borders** drop-down list. Your options include **None**, **Plain**, and **Sculptured**.
- Specify a percentage in the **Percent of width taken by prompt** field. This percentage determines how much of the viewable window is used to display the prompt. You can change this number by clicking the up or down arrows, or by typing a number directly in the field.
- 5. Optionally, change the number in the **Maximum multiple-choice options to display** field to control the number of multiple-choice options that appear in a question or answer summary.

Control Which Variables are Asked in a Question Summary

Typically question summaries include all questions in a template. However, some questions may not be asked unless certain conditions are met or certain answers are entered. These kinds of questions are called conditional variables. You can limit question summaries so they show only the questions that are applicable, based on certain answers you provide.

An example of a conditional variable might be the section of an employee agreement template that deals with employee benefits. Questions relating to employee benefits are asked only if the employee qualifies for benefits. If you assemble a question summary for an employee who does not meet the conditions, the benefits questions should not appear in the summary.

To control conditional variables in a question summary

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Question and Answer Summaries** folder. The window changes to show several summary options.
- 3. In the Question Summaries group, select Honor IF instructions when answered.

This setting will take effect when you assemble a document, answer the conditional variables, and then view the question summary.

To once again generate a question summary that shows all the document questions, clear **Honor IF instructions when answered**.

Personal Information

At a Glance: Personal Information (HotDocs Options)

 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Personal Information Spell Checking Personal Information Spell Checking File Locations File Locations File Management Template Set Updates Plugins 	HotDocs Options
OK Cancel	 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Management Template Set Updates Plugins Anagement Precede prompts with bullet Two-Column Format Precede answers with bullet Two-Column Format Borders: None Percent of width taken by prompt: 30 Question Summaries Wonor IF instructions when answered

After opening $\mathbf{\overline{M}}$ **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you click on **Personal Information** you will see the **Personal Information Options** dialog box, from here you can enter details that HotDocs will save as personal information that can be used in the script editor and in templates as Personal Information Variables.

The large field \underline{B} on this dialog contains a list of Personal Information variables and their answers. Using the buttons below \underline{C} you can **Edit** existing Personal Information variables, **Add** new ones and **Delete** Personal Information variables that are no longer useful.

To find out more about setting options for Personal Information follow the links below:

- Create Personal Information Variables
- At a Glance: The Add/Edit Personal Information Dialog Box
- Edit Answers for Built-In Personal Information Variables

At a Glance: The Add/Edit Personal Information Dialog Box

Edit Personal Information	? 💌
Variable:	
	Α
Answer:	
	В
	OK Cancel

After opening I HotDocs Options from the toolbar (or Tools menu) in your HotDocs Library you can choose from a list of option pages. If you click on **Personal Information** you will see the **Personal Information Options** dialog box, from here you can see the details that HotDocs has saved as personal information that can be used in the script editor and in templates as Personal Information Variables. Click the **Add** button, or click on any one of these items and then click the **Edit** button, to display the **Add/Edit Personal Information Dialog Box**.

In the first field A you can enter the name of the Personal Information variable you are creating or editing and in the second field you can enter the answer you want assigned to the Personal Information variable. This answer is stored in the system registry. Once given, you will not be prompted to enter it again.

To find out more about editing Personal Information follow the links below:

• Edit Answers for Built-In Personal Information Variables

Create Personal Information Variables

Personal Information variables gather basic information, such as your name, company, and phone number. This information is stored in the Current User section of the system registry so it can be used in assembled documents without you having to re-enter it.

You can create your own Personal Information variables for use in templates you automate. (See Insert a Personal Information Variable for details on how to do this.) You can also remove any custom Personal Information variables you create.

To create a new Personal Information variable

1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.

- 2. Click the **Personal Information** folder. The right pane changes to show several Personal Information variables.
- 3. With the contents of the **Personal Information** folder displayed, click **Add**. The **Add Personal Information** dialog box appears.
- 4. Type the name of the variable in the **Variable** field.
- 5. If you know the value you want assigned to the variable, type it in the **Answer** field and click **OK**.

To remove custom Personal Information variable you have created, select the variable and click **Delete**. Click **Yes** to confirm the deletion.

Edit Answers for Built-In Personal Information Variables

Personal Information variables gather basic information, such as your name, company, and phone number. This information is stored in the Current User section of the system registry so it can be used in assembled documents without you having to re-enter it.

When you assemble a document using a template that has Personal Information variables, HotDocs asks for any personal information required by the template that you haven't already entered. If you want to specify this information before document assembly, however, or if you want to change existing answers, you can do this at the HotDocs Options dialog box.

Because built-in variables are shared between all users of your workstation, you cannot delete the variable. You can, however, remove any personal information that has been entered for it.

To edit answers for Personal Information variables

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Personal Information** folder. The right pane changes to show several Personal Information variables.
- 3. Select the variable you want to modify and click **Edit**. The **Edit Personal Information** dialog box appears.
- 4. Enter an answer in the **Answer field** and click **OK**. The value is now listed in the **Answer** column.

To remove an answer for a Personal Information variable, select the variable that contains the answer and click **Delete**. Click **Yes** to confirm the deletion.

Spell Checking

At a Glance: Spell Checking (HotDocs Options)

HotDocs Options	? 💌
 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins 	Spell Checking Ignore capitalized words (e.g. Canada) Ignore words in all capitals (e.g. ASAP) Ignore words with mixed case (e.g. InLine) Ignore words with digits (e.g. Win95) Match case when comparing Auth case when replacing Auways suggest replacements Make phonetic suggestions Make typographical suggestions Make split word suggestions Check multiple-line text only User dictionary: C:\Users\HolliCooper.HOTDOCS\Documents\HotDocs\Spelling\Personal [

After opening $\mathbf{\overline{M}}$ **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you click on **Spell Checking** you will see the **Spell Checking Options** dialog box, from here you can set the options for spell checking in HotDocs and specify your user dictionary.

There are 12 check boxes **B** on this dialog to set the spell checking options in HotDocs. You can choose to check or uncheck any of the following:

- Spell check will ignore capitalized words
- Spell check will ignore words in all capitals
- Spell check will ignore words with mixed upper and lower case letters
- Spell check will ignore words with digits
- Spell check will match the case of the unrecognized word when comparing
- Spell check will match the case of the unrecognized word when replacing
- Spell check will always suggest replacement words
- Spell check will make phonetic suggestions for unrecognized words

- Spell check will make typographical suggestions for unrecognized words
- Spell check will make split word suggestions for unrecognized words
- Spell check will only check multiple-line text
- Spell check will use the UK spelling dictionary

Below the check box options you can choose to change your default **User Dictionary** C. Using the **Browse** button to the right of the field you can navigate to a different dictionary file on your computer.

To find out more about setting options for Spell checking follow the link below:

• Change Your Spell Checking Options

Change Your Spell Checking Options

HotDocs can spell check text used in component files as well as answers entered during an interview.

As HotDocs spell checks, it looks at the spell checking settings stored in the HotDocs section of the system registry to determine what words the spell checker looks at and how it decides on words to suggest as possible alternatives to words not in the dictionary. You can change these settings.

To change your spell checking options

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Spell Checking** folder. The contents of the window change to show your spell checking options.
- 3. Select appropriate options, as described in the following table:

То	Do This
Cause HotDocs to ignore capitalized words, such as names, cities, titles, and so on	Select Ignore capitalized words (e.g. Canada).
Cause HotDocs to ignore words that contain all capital letters, such as acronyms or abbreviations	Select Ignore words in all capitals (e.g. ASAP).
Cause HotDocs to ignore words that contain both uppercase and lowercase letters	Select Ignore words with mixed case (e.g. InLine).
Cause HotDocs to ignore words that also contain digits	Select Ignore words with digits (e.g. Win95).
Cause the replacement words HotDocs suggests to match the case of the misspelled word	Select Match case when comparing.

Cause HotDocs to match the case of misspelled words when replacing them	Select Match case when replacing.
Cause HotDocs to provide a list of possible replacements for misspelled words	Select Always suggest replacements.
Cause HotDocs to provide a list of possible replacements based on words that sound like the misspelled word	Select Make phonetic suggestions.
Cause HotDocs to provide a list of possible replacements based on words that are typographically similar	Select Make typographical suggestions.
Cause HotDocs to provide a list of possible replacements that are made up of more than one word	Select Make split word suggestions.
Cause HotDocs to check the spelling only if the text is more than 1 line	Select Check multiple-line text only.
Cause HotDocs to use the British spelling dictionary rather than the English dictionary	Select Use UK spelling dictionary.

When a word is not in the spelling checker's dictionary, HotDocs checks for it in your personal dictionary, *Personal Dictionary.TLX*. If it is not found there, you can add it by clicking **Add** at the **Check Spelling** dialog box.

You can change these settings whenever you are spell checking. To do this, click **Options** at the **Check Spelling** dialog box.

Template Development

At a Glance: Template Development (HotDocs Options)



After opening P HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \boxed{A} .

If you click on **Template Development** you will see the **Template Development Options** dialog box, from here you can set the default options for new templates and decide which warning messages you would like displayed.

The first set of options **B** focus on settings used while developing a template. From here you can choose to have HotDocs auto complete component names while typing and to always automatically generate the name when a dialog element is created. There are two drop-down lists where you can choose what style of component icons you would like displayed in the component lists and whether HotDocs should insert returns after instructions in text template.

The second set of options focus on settings for component file properties for new templates. Using the three check boxes you can choose whether HotDocs marks answers in assembled documents, allows you to edit answers at the document preview tab and whether the component file will be automatically enabled for use with HotDocs Server.

The third set of options **D** allows you to choose which warning messages HotDocs will display. You can choose to have HotDocs warn before automatically converting a template, when you are editing a pointed component file, when variables are automatically removed from dialogs and when the component editor is closed via the title bar.

To find out more about setting options for Template Development follow the links below:

- Have HotDocs Automatically Complete Component Names
- Have HotDocs Automatically Generate Dialog Element Component Names
- Choose Icon Style for Component Lists
- Control When Returns Are Inserted After Instructions in Text Templates
- Enable Document Editing Options for All New Templates
- Enable All New Templates for Use with HotDocs Server
- Display HotDocs Warning Messages

Have HotDocs Automatically Complete Component Names

You can have HotDocs suggest existing component names as you type. HotDocs can also suggest example formats and text patterns.

To turn on Auto Complete in Component Editors

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Development** folder. The window changes to show several different template development options.
- 3. Select Auto complete while typing component names.

Have HotDocs Automatically Generate Dialog Element Component Names

When creating dialog elements, you can have HotDocs automatically name the component for you, based on information about the type of element you are creating. Specifically, HotDocs includes in the name the type of element you are creating, followed by any additional information you enter about the element, such as the display text or caption. For example, if you are creating a Script Link element that links to a Computation variable named *Tax Calculator*, HotDocs will name the component *Script Link Tax Calculator*.

When you select this option, the **Generate name automatically** property will be set at the **Dialog Element Editor** for each new dialog element you create.

To auto-generate dialog element component names

1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.

- 2. Click the **Template Development** folder. The window changes to show several different template development options.
- 3. Select Always generate name automatically when dialog element created.

Choose Icon Style for Component Lists

When HotDocs displays lists of components, it uses an icon, followed by the name of the component. You can change the icon to show either a symbol of the component, or an abbreviation for the type of component it is. For example:

Displays all components with a short text abbreviation representing the component type. For example, this icon represents a Text variable.
 Displays all components with a graphic image representing the component type. For example, this

A Displays all components with a graphic image representing the component type. For example, this icon represents a Text variable.

To change the component icon style

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Development** folder. The window changes to show several different template development options.
- 3. At the **Component type icons in component lists** drop-down list, select either **Graphic** or **Text**.

Control When Returns Are Inserted After Instructions in Text Templates

By default, when inserting instructions in a template, HotDocs inserts a return after the instruction. Then, as the instruction is processed and removed from the assembled document during assembly, HotDocs removes the return it inserted as well.

One reason why HotDocs includes these returns after instructions is to make it easier to view template text in relation to the instructions, particularly when the instruction controls the appearance of an entire paragraph of text.

In this example, returns are inserted after the instruction. This keeps instructions on their own lines and makes it easier to view how the instructions relate to the text.

SHIPPED TO

«IF Different Shipping Address»
«Shipping Name»
«Shipping Street»
«Shipping City», «Shipping State» «Shipping ZIP»
«ELSE»
«Billing Name»
«Billing Street»
«Billing City», «Billing State» «Billing ZIP»
«END IF»

In this example, returns are not included after each instruction. This makes it more difficult to see the conditions in relation to the text.

SHIPPED TO «IF Different Shipping Address»«Shipping Name» «Shipping Street» «Shipping City», «Shipping State» «Shipping ZIP» «ELSE»«Billing Name» «Billing Street» «Billing City», «Billing State» «Billing ZIP»«END IF»

In some situations, however, having this return in the template text may actually make the text more difficult to read, for instance, because the instructions control the appearance of text within a paragraph and not the paragraph as a whole.

In this example, the returns after the instructions force text within a paragraph to a new line. This may be confusing to understand.

Health insurance is provided as one of the benefits of your employment with Hobble Creek Publishing. The insurance policy is through TLC Insurance and will cover you, your spouse, and any minor children. Your insurance coverage will begin «IF DAY OF(Hire Date) = 1» «Hire Date»«ELSE»

«Coverage Start Date»«END IF»

In this example, the returns after the instructions are omitted, allowing all of the text to remain in the same paragraph.

Health insurance is provided as one of the benefits of your employment with Hobble Creek Publishing. The insurance policy is through TLC Insurance and will cover you, your spouse, and any minor children. Your insurance coverage will begin «IF DAY OF(Hire Date) = 1»«Hire Date»«ELSE»«Coverage Start Date»«END IF».

In most cases, you most likely want returns inserted when merging an entire paragraph, and you want returns omitted when inserting just short blocks of inline text. Whatever your preference, you can choose an option that best suits your needs.

To determine whether (and when) HotDocs inserts returns after instructions

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Development** folder. The window changes to show several template development options.
- 3. Click the **Insert returns after instructions in text templates** drop-down button and choose an option, based on the following information:
 - Choose Always to have HotDocs always insert a return character after an instruction. (Remember, HotDocs will remove the return during assembly.)
 - Choose Smart to have HotDocs determine whether it should insert a return character or not. It determines how to insert these returns based on surrounding characters (such as paragraph breaks) in the text. (See Understand How Smart Returns are Inserted After Instructions for a complete description.)
 - Choose **Never** to have HotDocs never insert a return character after an instruction.

WordPerfect 8 and WordPerfect 9 users: Because of some problems with underlying text selection when inserting instructions in a template, you should set **Insert returns after instructions in text templates** to **Always**. Selecting **Smart** or **Never** will cause errors during assembly.

Enable Document Editing Options for All New Templates

Document editing options are currently available for Microsoft Word users only.

Often, as users are reviewing their assembled documents, they may need to see exactly where their answers have been merged. Sometimes they may need to edit or change those answers as well. To allow this functionality for your users, you must specify the component file property for each of the templates in

your set. Rather than set it in each template individually, you can specify an option that will set this property in each new template you create.

Specifically, these options allow the user to highlight where answers in the document have been merged, and edit the answer while at the **Document Preview** tab.

For more information on working with the assembled document, see Edit Answers at the Document Preview Tab.

To set the component file properties for editing text documents

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Development** folder. The window changes to show several different template development options.
- 3. In the **Component File Properties for New Templates** group, select **Mark answers in assembled documents.**
- 4. Optionally, select Enable Edit Answer at Document Preview tab.

Now, in each new template you create, these component file properties will be set.

To set the properties for existing templates, change them at the **Component File Properties** dialog box (see Change Component File Properties) or change the properties in multiple templates at once using Template Manager (see Modify Component File Properties Across Multiple Files).

Enable All New Templates for Use with HotDocs Server

When creating templates for use with HotDocs Server, some features (such as clause libraries and database components) are not allowed. You can have HotDocs warn you of this when you are creating templates and you attempt to use such a feature.

Additionally, when templates are enabled for HotDocs Server, you can test assemble the template in a Web browser as well as publish the files required for the template to reside on the server.

To allow all new templates you create to be used with HotDocs Server

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Development** folder. The window changes to show several different template development options.
- 3. In the **Component File Properties for New Templates** group, select **Enable for use with HotDocs Server**.

Now, in each new template you create, this component file property will be set. To set the properties for existing templates, change them at the **Component File Properties** dialog box (see Change Component File Properties). Or change the properties in multiple templates at once using Template Manager (see Modify Component File Properties Across Multiple Files).

If you don't want every template you create to be enabled for server-based assembly, you can set this option on a per-template basis by changing the specific template's component file properties. (See Change Component File Properties.)

Display HotDocs Warning Messages

Sometimes HotDocs warns you when it is about to perform an operation that could impact the work you are doing. When HotDocs displays these warnings, it gives you the option of suppressing the warnings the next time you perform a similar operation. You can have HotDocs re-display these warnings after you have suppressed them.

To cause HotDocs to display warnings

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Development** folder. The window changes to show several different template development options.
- 3. In the **Warning Messages** group, make your selection, based on the following information:
 - To have HotDocs warn you when it is about to convert a template to the current version of HotDocs, select **Warn before automatically converting templates**. (See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs and Convert a Single Template to a New File Format.)
 - To have HotDocs warn you when you edit a template that is using a shared component file, select **Warn when editing pointed component files**.
 - To have HotDocs warn you when you are moving a variable from one dialog to another, select **Warn when variables automatically removed from dialogs**. (See Use the Same Variable or Clause in Two or More Dialogs.)
 - To have HotDocs remind you that changes are saved when you close a Component Editor using buttons in the title bar, select **Warn when component editors closed via title bar**.

Now, when HotDocs performs related actions, it will display the associated warning.

At a Glance: Script Editor (HotDocs Options)

Setting HotDocs Options

HotDocs Options		
 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Spell Checking Template Development Script Editor Template Manager File Locations File Management Template Set Updates Plugins Automatic indentation Select Font. Select Font. Select font: MS Sans Serif (11 point) 		
OK Cancel		

After opening $\mathbf{\overline{M}}$ HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you double click on **Template Development** you will see a list open below containing a file for setting the options for the **Script Editor** and the **Template Manager**. Click on **Script Editor** to bring up the **Script Editor Options** dialog box.

The first section of options **B** contains four settings for the script editor. The three check boxes allow you to choose if HotDocs will automatically indent script, include less common keywords in the auto-complete list and allow the tab key to insert a tab in scripts. In the field below this you can enter a number for how many characters wide tab will be when used in the script editor (it must be between 1 and 15).

In the second section **G** you can decide whether or not you want syntax coloring in the script editor by checking or unchecking the box. You can then use the list below to set colors and styles for each type.

In the third section D you can select a font and font size for the text in the script editor.

As the text in the script editor will only be seen by the template developer the choices of font and colors are available to help make the editor easier to use rather than for any aesthetic qualities on the final template.

To find out more about setting options for Script Editor follow the link below:

• Change Script Editing Options

Change Script Editing Options

You can control the formatting HotDocs uses when you write scripts. For example, you can designate the font size of the text used in the script editor, as well as associate colors with certain types of script keywords, placeholders, components, or operators.

To change the options available for writing scripts

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the **Template Development** folder. The folder expands to show a subfolder named **Script Editor.**
- 3. Click the **Script Editor** folder. The window changes to show several options for customizing HotDocs scripts.
- 4. Perform any of the following tasks:

То	Do This
Have HotDocs start the next line of a script (after you enter a return) at the same level of indentation	Select Automatic indentation.
Include keywords for months, language codes, and so forth in the auto-complete list	Select Include seldom-used keywords in auto-complete list. (If you'd rather exclude these keywords from the list, clear this option.)
Allow the Tab key to insert a tab in a script, rather than move you to the next field in the component editor	Select Tab key inserts a tab in scripts . Now, whenever you press Tab or Shift+Tab , HotDocs will insert a tab in the script.
	This number must be between 1 and 15.
	If you select this option, you can no longer use keyboard commands to move from the Script field to another field in the component editor. You must use your mouse to click out of the Script field.
Specify the number of characters HotDocs should use when you insert a tab in the script	Enter a number in the Tab width field.

Have HotDocs use colors to differentiate between instruction and expression keywords, operators, and component names	Select Syntax coloring . (You can then either accept the default colors HotDocs assigns to these, or you can designate your own colors at the Colors and Attributes list.)
Assign colors and other formatting	Select the item, click the Color drop-down
attributes to the various parts of a script	button, and choose a color. You can also select
	Bold to make the item bolded, and you can
These "parts" include instruction keywords,	select Italic to italicize the item.
expression keywords, components (such as	
variables and dialogs), text values (literal	To restore the HotDocs defaults, select the
text strings), other values (such as literal	script element, click the Color drop-down
number and date values and language	button, and choose Default from the top of the
codes), placeholders, operators, commented	Palette tab.
scripts, and any syntax HotDocs doesn't recognize.	
Specify the font used for the script	Click Select Font and make your selection. You can also specify the size of the script text.

At a Glance: Template Manager (HotDocs Options)

HotDocs Options		? ×
Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Script Editor Template Manager Word Processors File Locations File Management Template Set Updates Plugins	Template Manager Include supplemental components in component list Show component type column Show component status column Overlay status hint on component icons	
	Ōĸ	Cancel

After opening $\mathbf{\overline{M}}$ HotDocs Options from the toolbar (or Tools menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you double click on **Template Development** you will see a list open below containing a file for setting the options for the **Script Editor** and the **Template Manager**. Click on **Template Manager** to bring up the **Template Manager Options** dialog box.

There are 4 check boxes at the top of the dialog **B**. By checking these you can have HotDocs include supplemental components in the components list, show the component type column in template manager, show the component status column in template manager and overlay a status hint on component icons.

To find out more about setting options for Template Manager follow the link below:

• Set Template Manager Options

Set Template Manager Options

You can change the way Template Manager displays the information it generates about components it is showing.

To change the way Template Manager displays component information

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click on the **Template Development** folder to expand it. The window changes to show several template development options.
- 3. Click the **Template Manager** folder. The window shows Template Manager options.
- 4. Complete any of the tasks listed in the following table:

То	Do This
Expand the list of components to include example formats, patterns, and so forth.	Select Include supplemental components in component list.
Include a column in the component list that shows component types	Select Show component type column.
Include a column in the component list that shows the status of the components	Select Show component status column . The status tells you whether the component is OK (used in both the template and component file), unused (the component is not referenced in any templates), or missing (the component is referenced in a template, but not found in the component file).
Have Template Manager overlay a graphic reminder of what the component's status is on the component's icon	Select Overlay status hint on component icons.

You can also access these options by clicking the **Options** button in the Template Manager toolbar.

Word Processors

At a Glance: Word Processors (HotDocs Options)


After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you select **Word Processors** you will see the **Word Processors Options** dialog box. From here you can change what word processors HotDocs supports and choose your default word processor.

The two lists of word processors at the top of the form B display all the word processors that HotDocs can support. HotDocs identifies which word processors you have installed on your computer and ungrays those word processors only. Any word processors already displaying a check are already supported. If you check an empty box next to any of the ungrayed options then these are supported once you close the **Options** dialog box.

The drop-down arrow C below these lists enables you to select a default word processor from your supported word processors.

While you can choose any version of WordPerfect as the default version HotDocs uses to create a new WordPerfect template, for Word, you are limited to selecting the latest version of Word installed on your machine as the default version HotDocs uses to create new Word templates.

Using the check box options below the drop-down, you can choose to **Send assembled documents to the word processor** that matches their template type (useful if you have both Microsoft Word and WordPerfect installed) and choose to **Launch word before sending a document to the word processor** (to allow Word Macros to run). To find out more about setting options for Word Processors follow the links below:

- Install Support for New Word Processors
- Change Your Default Word Processor

Install Support for New Word Processors

Before installing support for a new word processor, make sure you run the newly installed word processor program at least once. This creates entries for the new word processor in the Windows System Registry that HotDocs needs to properly integrate with the word processor.

When you install a new word processor and you want to use it with HotDocs, you can install support for it through HotDocs Options. If multiple users use the workstation, support for the word processor will be installed for them as well.

To install new word processor support

- 1. Install the new word processor and run it at least once.
- 2. Start HotDocs. (See Start HotDocs.)
- 3. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 4. Click the **Word Processors** folder. The right pane changes to show a list of word processors which HotDocs supports. (Word processors for which you have already installed support are selected, while word processors that are not installed are disabled.)
- 5. Select the word processor for which you want to install support.
- 6. Click **OK**. The **HotDocs Options** dialog box closes.

Once you close the HotDocs Options dialog box, HotDocs runs the Current User Setup, which copies the required files to the correct program folders. (See Understand HotDocs Installation.)

Click here to see a list of which word processors are supported for this release.

To make the newly installed word processor the default word processor, you must manually select it after you install support for it. See Change Your Default Word Processor for details.

You may need to restart the word processor to add the correct HotDocs buttons and toolbars to it.

Change Your Default Word Processor

If you have both Microsoft Word and WordPerfect installed for HotDocs, you must choose which word processor HotDocs should use as the default. The default word processor controls which word processor file type will be suggested when you create a new template. It also determines which word processor will be opened when you finish the interview and send the assembled document to it.

To specify a default word processor

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the Word Processors folder. The right pane changes to show supported word processors.
- 3. Click the **Default word processor** drop-down button and select a word processor.
- 4. Optionally, to send assembled documents to the word processor that matches the template file type, select **Send assembled documents to word processor based on template type**.
- 5. Optionally, to start Word before you send the assembled document to it, select Launch Word before sending to word processor (allows auto macros to run). (This is useful if the underlying template uses AutoExec or AutoOpen macros, since HotDocs explicitly disables the execution of Auto macros in Word before it sends a document to Word after assembly.)

Click here to see a list of which word processors are supported for this release.

At a Glance: Word Processor File Locations (HotDocs Options)

Setting HotDocs Options

HotDocs Options		6	
Interviews and Dialogs Document Assembly	File Locations HotDocs Folders		
Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates	Type of File Folder Program Files C:\Prog Schema Files C:\Prog JavaScript Files C:\Prog Silverlight Files C:\Prog Library Files C:\User Template Files C:\User Template Sets Public III III	gram Files (x86)\HotDocs\ gram Files (x86)\HotDocs\Schemas\ gram Files (x86)\HotDocs\ServerFiles\ gram Files (x86)\HotDocs\ServerFiles\bi rs\HolliCooper.HOTDOCS\Documents\ ns\HolliCooper.HOTDOCS\Documents\ Documents\HotDocs\Templates\ Edit	Ŧ
A Plugins	Reference Paths Keyword Path CUKAdditionalTutorials Publi PUBCMCK bug Publi	n ic Documents\HotDocs\Templates\ ic Documents\HotDocs\Templates\ <u>E</u> dit <u>A</u> dd <u>D</u> elete OK Canc	:el

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you double click on **Word Processors** you will see a list open below containing a file for each of your supported word processors. Click on one to bring up the **Word Processor File Locations** dialog box for that particular word processor, from here you can change the default file locations for that word processor.

From here you can edit three file locations for the selected word processor. In the first field \underline{B} you can change the default documents folder by clicking on the \underline{A} **Browse** button next to it and navigating to the folder you need. In the second field \underline{C} you can change the default folder that HotDocs stores and looks for templates. In the third field \underline{D} you can change the startup folder for the chosen word processor.

To find out more about setting options for Word Processors File Locations follow the link below:

• Change Word Processor File Locations

Change Word Processor File Locations

By default, HotDocs uses certain folders that were specified during installation to store text templates and assembled text document files that are used with HotDocs. These folder names and file paths become the default location HotDocs looks when no specific path is assigned at the file's properties.

At the library, you can check whether a full path is given by selecting the template and clicking the **Properties** button. If a full path is given, HotDocs uses that information when accessing the file. Otherwise, HotDocs uses the default file location specified here.

You can change the locations HotDocs looks for these files.

To change the location of word processor files

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Double-click the **Word Processors** folder so that it expands to show subfolders for each word processor that is installed.
- 3. Click the word processor folder for which you want to make changes. The right pane changes to show the different file locations of the document, template, and startup files (or macro files for WordPerfect users).
- 4. For each file location you want to change, click the **Browse** button next to the file location path and navigate to the new location.

If templates in your library use the default file path for its templates, you must move your templates to this new location or HotDocs will not be able to find them. See Move Items to New Locations on Disk.

File Locations

At a Glance: File Locations (HotDocs Options)

Setting HotDocs Options

HotDocs Options		? <mark>×</mark>
Hotbocs Options Interviews and Dialogs Document Assembly Arkup View (Word) Hotbocs Models Form Documents Question and Answer Summaries Decument Information	File Locations HotDocs Folders Type of File Program Files Schema Files JavaScript Files	Folder C:\Program Files (x86)\HotDocs\ C:\Program Files (x86)\HotDocs\Schemas\ C:\Program Files (x86)\HotDocs\ServerFiles\
Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins	Silverlight Files Library Files Template Files Template Sets ✓	C:\Program Files (x86)\HotDocs\ServerFiles\bi C:\Users\HolliCooper.HOTDOCS\Documents\ C:\Users\HolliCooper.HOTDOCS\Documents\ Public Documents\HotDocs\Templates\ III Edit
	CUKAdditionalTuto CUKAdditionalTuto PUBCMCK bug	Path arials Public Documents\HotDocs\Templates\ Public Documents\HotDocs\Templates\ Edit Add Delete
		OK Cancel

After opening HotDocs Options from the toolbar (or Tools menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you select **File Locations** you will see the **File Locations Options** dialog box. From here you can change where HotDocs stores saved files and you can edit the assigned reference paths of HotDocs Libraries.

From this folder, you can change the locations of the following file types: ANX, CMP, HDI, HDL, HDP, HFD, HFT, HPD, HPT, PDF, TTX. You change text file locations from the Word Processors folder just above the File Locations folder.

In the first list **B** you can see the locations of all the HotDocs folders. By selecting a folder from this list and clicking the **Edit** button below the list you can change where HotDocs will automatically save those files.

In the second list C you can see the file paths of your HotDocs libraries. By selecting a library from this list and clicking the **Edit** button below the list you can change where HotDocs will automatically save those files. You can also click **Add** to add a new library file path to the list.

To find out more about setting options for file locations follow the links below:

- Change HotDocs Program File Locations
- Assign Reference Paths to HotDocs Files

• At a Glance: Word Processors (HotDocs Options)

Change HotDocs Program File Locations

When you install and use HotDocs, the program makes several entries in the System Registry that indicate to HotDocs where to look for files, as well as where to suggest you save files. (See Understand HotDocs Installation.) In most situations, HotDocs uses this information each time you perform a task in HotDocs. For example, each time you create a new template, HotDocs suggests a default location for the new template, based on the information it finds in the System Registry. If you ever need to change these default locations, you can do so at HotDocs Options.

If a full path is specified for the file you are attempting to access, that file path will be used, rather than the default file path. (To check whether a full path is given, select the item and click the **Properties** button.)

You can change the default location HotDocs looks for the following files:

- Library files (HDL files).
- Template files (HFT, HPT, and CMP files). (To change the default location for text templates, see Change Word Processor File Locations.)
- Template Sets (installed HDI files).
- Catalog Files (files used to keep published template sets up to date).
- Answer files (ANX files).
- Form document files (HFD, HPD, and PDF files).
- Publish Settings files (HDP files).

To change the default location of HotDocs files

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **File Locations** folder. The window changes to show a listing of the different HotDocs file paths.
- 3. In the **HotDocs Folders** group, select a file type and click **Edit**. The **Browse for Folder** dialog box appears
- 4. Browse to the new location and click **OK**. HotDocs will use this location as the new default.

You must restart HotDocs for these changes to take effect.

Assign Reference Paths to HotDocs Files

A HotDocs library will work only if each template listed in the library is located in the exact path specified in the **Item Properties** dialog box for that template. This makes it difficult to create a library that will work for multiple users who will not store the templates in the same locations. A reference path lets you create a library that will work, regardless of where users store their templates.

A reference path is a folder path for a template in which the drive letter and some or all of the folder names have been replaced with a caret (^) and a keyword. The keyword is associated with an actual folder path in the user's HotDocs settings. When HotDocs needs to open a template, it replaces the caret and keyword in the folder path for the template with the folder path that is associated with the keyword in the user's HotDocs Options.

Some common uses of reference paths are:

- With distributed template sets which allow users to choose where they install the templates (even if they can only choose the drive to which the templates are installed).
- With templates located on a network drive that is mapped to a different letter for some of the users.

In order for your library to work for users regardless of where they store the templates, you need to enter a reference path for all templates and clause libraries in the library. You should use the same keyword for all the items in the library.

If you know the library items won't all be stored in the same subfolders, use the keyword only in place of the part of the path that is identical for all items. For example, if some of the templates will be stored in a *General* subfolder and others will be stored in a *Family Law* subfolder, you would use the reference path **^keyword\General\template.ext** for those templates in the *General* subfolder and **^keyword\Family Law\template.ext** for those templates in the *Family Law* subfolder (where *keyword* is a placeholder for the keyword you use and *template.ext* is a placeholder for the file name and extension of the template).

To associate a keyword with a path and assign the reference path

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **File Locations** folder. The window changes to show a listing of the different HotDocs file paths.
- 3. In the **Reference Paths** section, click **Add**. The **Add/Edit Reference Path** dialog box appears.
- 4. Type a keyword for the reference path in the **Keyword** field.
- 5. Click the **Browse** button next to the **Path** field, and specify the path to the subfolder you want to use.
- 6. Click **OK**.

Once you have designated the keywords and paths that HotDocs needs to know about your files, you can assign reference paths to the templates. (These templates must be saved to the subfolder you specified in Step 5, above.)

- 7. At the HotDocs library window, select the template for which you want to specify a reference path and click the **Properties** button. The **Item Properties** dialog box appears.
- 8. If the **File name** field contains a folder path as well as a file name, delete the part of the path that is the same for all items in the library.
- 9. Place your cursor before the first letter in the **File name** field and type caret (^), the keyword, and a backslash (\).
- 10. Click **OK**.

When publishing your template set as an auto-install (.HDI) file, HotDocs automatically creates the reference paths and enters the correct information in the user's HotDocs settings during installation.

For instructions on assigning reference paths to multiple library items, see Change the Properties for Multiple Items in a Library.

File Management

At a Glance: File Management (HotDocs Options)

Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Government Word Processors File Locations File Locations Template Set Updates Plugins	File Management B Manage answer files using:	HotDocs Answer File Manager 🔹
U	Document manager ODMA DLL loc	ation:

After opening $\mathbf{\overline{M}}$ HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you select **File Management** you will see the **File Management Options** dialog box. From here you can change how HotDocs manages answer files and assembled documents.

The first option **B** allows you to choose how you would like to manage your answer files. You can choose from:

- HotDocs Answer File Manager
- Windows Explorer

To find out more about setting options for file management follow the links below:

- Manage Answer Files
- Manage Assembled Document Files

Manage Answer Files

To work with answer files, you can either use Answer File Manager, or you can use Windows Explorer. (See Introduction: Use Answer Management.) Answer File Manager uses an answer library (.HAL) file to create, store, and select answer files. Windows Explorer allows you to work with answer files using the common **File > Open** or **File > Save** dialog boxes.

Which option you choose depends on your preference. For example, Answer File Manager works well if you are the only user and you like the functionality available with a library. However, Windows Explorer makes more sense for managing answer files in a multi-user setting. Windows Explorer also lets you see more clearly where the actual answer files are saved.

To change how HotDocs locates and manages answer files

- 1. At the HotDocs library, click the **Options** button.
- 2. Click the File Management folder.
- 3. Click the Manage answer files using drop-down button and select an option:
 - HotDocs Answer File Manager uses an answer library to create, store, and select answer files.

The first user to open an answer library is the only one who has *write access* to it. That means only one person can modify items in the library at any given time. If multiple users must access the answer files, use Windows Explorer to manage the files.

• Windows Explorer uses Windows Explorer to create, save, and select answer files.

Template Set Updates

At a Glance: Template Set Updates (HotDocs Options)

Setting HotDocs Options

HotDocs Options			? <mark>×</mark>
Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Plugins	Template Set Updates Perform scheduled checks for up Show progress window when ch Alert user when update icon is d Select the template sets you want t Title	odates when HotDocs is started ecking for updates lisplayed on the status bar to keep up-to-date: Catalog File	
		OK Car	ncel

After opening **HotDocs Options** from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane A.

If you select **Template Set Updates** you will see the **Template Set Updates Options** dialog box. From here you can change how HotDocs updates your template sets.

There are 3 check boxes at the top of the page , these control when HotDocs updates the template sets and how the update progress will be displayed:

- Check the first box if you would like HotDocs to check for updates every time the library is opened.
- Check the second box if you would like HotDocs to show a progress window while it is checking for updates.
- Check the third box if you would like HotDocs to alert you when there is an update icon displayed on the status bar.

Underneath these options will be a list \bigcirc of all the template sets you can keep updated.

Follow the link below to find out more about setting options for template set updates:

• Determine How Frequently HotDocs Checks for Template Set Updates

Determine How Frequently HotDocs Checks for Template Set Updates

If you are using a published template set and the template publisher provides automatic updates, you can have HotDocs check for these updates each time you start HotDocs. (Please check with your template set publisher to determine if your template set includes automatic updates.)

To check for template set updates

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Template Set Updates** folder. The window changes to show update options.
- 3. Complete any of the following optional steps:

То	Do This
Always check for updates when you start HotDocs	Select Perform scheduled checks for updates when HotDocs is started . (If HotDocs detects an update, it displays a warning icon in the status bar of the HotDocs window.)
Display a window that shows the progress HotDocs makes in installing the updates	Select Show progress window when checking for updates.
Choose the template sets for which you want to check for updates	Select the set in the Select the template sets you want to keep up-to-date list.

Plugins

At a Glance: Plugins (HotDocs Options)

Setting HotDocs Options

HotDocs Options		? ×
 Interviews and Dialogs Document Assembly Markup View (Word) HotDocs Models Form Documents Question and Answer Summaries Personal Information Spell Checking Template Development Word Processors File Locations File Management Template Set Updates Pugins 	B Plugins	
	C Enable Disable Preferences	
	ОК	Cancel

After opening $\mathbf{\overline{M}}$ HotDocs Options from the toolbar (or **Tools** menu) in your HotDocs Library you can choose from a list of option pages in the left hand pane \mathbf{A} .

If you select **Plugins** you will see the **Plugins Options** dialog box. From here you can enable or disable HotDocs Plugins.

In the **Plugin List B** you can see all the HotDocs plugins currently installed and you can click on an item in this list to highlight it.

With any item highlighted you can click the buttons C to **Enable** the plugin, **Disable** the plugin, or set your **Preferences**.

Follow the link below to find out more about setting options for plugins:

• Set Options for HotDocs Plugins

Set Options for HotDocs Plugins

If you are using HotDocs Plugins, you may need to enable or disable one or more at some point. You may also find it useful to set your preferences for each plugin. You can do all of this from the **HotDocs Options** dialog box.

To Enable/Disable a Plugin

- 1. At the HotDocs library, click the 🚾 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Plugins** folder. The window changes to show plugin options.
- 3. Select any of the currently installed plugins from the **Plugin List**.
- 4. Click either the **Enable** or **Disable** button.

To Set Plugin Preferences

- 1. At the HotDocs library, click the 🔽 **Options** button. The **HotDocs Options** dialog box appears.
- 2. Click the **Plugins** folder. The window changes to show plugin options.
- 3. Select any of the currently installed plugins from the **Plugin List**.
- 4. Click the **Preferences** button. This opens the **Plugin Preferences** dialog box.

5.

Automating Text Templates

Creating and Working with Template Files

Introduction: Template and Component Files

The Template File

The first step in automating a new HotDocs template is to create a template file. You can create three types of template files: text templates, form templates, and interview templates.

• You create a text template in Microsoft Word or WordPerfect. You can modify the underlying text of a text template both as you automate the template and as you view the assembled document in the word processor.

HotDocs assembles all Word templates using a DOCX-based assembly process. This means that if your template is in DOT or RTF format, HotDocs must assemble it and then convert it back into Word's native format. By developing the templates in DOCX, you can reduce the number of steps in this process, thus decreasing the amount of time it takes to assemble a document. In larger, more complex templates, this difference in speed will be quite noticeable.

- You create a form template using a Windows program (such as a draw program) to create a static form, or one in which the user will not be able to change or modify the underlying text. Once designed, you then use HotDocs to create a PDF version of the form. Finally, you use HotDocs Automator to place fill-in fields on the form where users' answers will be inserted.
- You create an interview template to gather common information (such as court, attorney, or client information) and save the answers for use in assembling other documents. Unlike text and form templates, users cannot generate documents from an assembled interview template—they can only generate an answer file. (The interview template is really a component file that contains ASK instructions. See Create an Interview Template.)

It is recommended that template names are kept to less than 52 characters and don't use non-ASCII characters. This will reduce the possibility of the template name causing errors if it is enabled for use with HotDocs Server.

The Component File

When you create a new template file, HotDocs automatically creates a companion file called the *component file*. The component file contains information about variables and other components used in the template. Both the template file and the component file are necessary for a template to work. Whenever you copy a HotDocs template—for example, to share a template with another user—you must be sure to copy both the template file and the component file.

HotDocs gives the component file the same name as the template file, except that the file name extension for the component file is .*CMP*. For example, if the template file name is *Insure.rtf* (for a Word template), the component file name will be *Insure.cmp*.

The following image shows a fully automated text template with its component file superimposed and overlapping the template:



Subject to the supervision and pursuant to the orders, advice, and direction of H «Employee Name» shall perform such duties as are customarily performed by one holding such position in other businesses or enterprises of the same or similar nature. Specific iob duties shall include «Job

The component file works automatically in the background—as you create various components in the template, they are automatically stored in the component file. Normally each template uses its own component file, but you can make two or more templates share one component file.

You work with components in the component file using Component Manager. For example, you can copy variables or other components from one component file to another, create variables, and drag variables from Component Manager directly into the template. You can also edit, print, search, and spell check components within your component file. Finally, if you need to edit components in several component files at once, you can use Template Manager.

HotDocs does not support templates with embedded mail merge data. You need to remove this data before assembling the template.

HotDocs Editing and Navigation Toolbar Buttons

Since HotDocs works inside your word processor, you use word processor commands for many basic tasks, such as typing text in the template, assigning font attributes and page formatting, and printing. Tasks that are specific to HotDocs are performed by using the HotDocs buttons. To access these buttons in Word 2007 and later, click the **HotDocs** tab on the Word ribbon. Using the drop-down menu below the HotDocs button on the tool bar you can access more options.

In Word 2003 HotDocs adds an editing toolbar, navigation toolbar and a HotDocs drop-down menu to the word processor window. The buttons described below can be accessed from there.

HotDocs Tab

Button Name	Description
HotDocs	Either starts HotDocs and opens the HotDocs library window, or brings it to the front if it is already open. (This button is always in the word processor toolbar, regardless of whether HotDocs is running.)
Save Template and Component File 	Saves the template and component file you are currently editing.
Save and Close Template	Saves and closes the template and component file you are currently editing.
Test Assemble	Test assembles the document or a selected portion of the document.
Variable Field «»	Opens the Variable Field dialog box where you can select the type of variable field you want to create and insert. If the cursor is in a variable reference, HotDocs opens the Variable Field dialog box with that information already appearing. If you're using Word 2000 or higher, you can double-click in a variable field and the Variable Editor will appear
IF Field	Opens the IF Field dialog box where you can create a true/false test for conditional text. If you need a more complex test, click IF Expression at the IF Field dialog box.
REPEAT Field	Opens the REPEAT Field dialog box where you can specify a dialog to gather information for a list. At this dialog box, you can also specify a sort order, filter, and format for merging the list of answers into the document. You can also REPEAT a database component.
INSERT Field	Opens the INSERT Field dialog box where you can choose to insert a clause, clause library, or template. You can also convert a selected section of template text into an inserted template or clause component.

ASK Field	Opens the ASK Field dialog box where you can create and insert an ASK instruction in the template. An ASK instruction causes either a dialog or a database to be asked at a specific place in the template.
Other Fields «»	Opens the Other Field dialog box where you can choose to insert an ASSEMBLE , PLAY , LANGUAGE , DEBUG , or SPAN field. The options will change depending which of these four options you select. Clicking on the drop-down menu button next to the Other Fields button will display a list of options. Clicking on the drop-down menu you can select Other Field , or from a list of Dot Codes . See the table below for more information on Dot Codes .
Edit Component	Opens the Component Editor where you can modify the properties of an existing component. (If your cursor is outside a component field, HotDocs will allow you to create a new component. However, if you create a new component, it will only be saved to the component file—it will not be inserted into the template.)
Component Manager	Opens the Component Manager window where you can create, edit, delete, rename, and copy components.
Create Markup	Opens a Markup document in a new window. This document can then be saved and distributed to non-HotDocs users who needs access to the template. You will notice the HotDocs tab has changed to show the HotDocs Markup Tools only.
Clause Library	Opens a clause library where you can create new clauses or add existing clauses.
HotDocs Help 😧	Displays the HotDocs Help viewer, which contains information on automating templates and assembling documents.
Apply Colors	Assigns custom colors to the different field types in the template. This can help you more quickly identify sections or parts of your template as you are automating. (You can choose the colors at HotDocs Options.)
Label Fields	Opens the Label Fields dialog box where you can choose to label matching IF and END IF instructions, REPEAT and END REPEAT instructions, and SPAN and END SPAN instructions. Additionally, where you've nested these instructions, you can identify the level of nesting. Finally, you can assign the word processor's hidden text property to these comments and labels so that you can show and hide them during automation.
Match Fields ≣	Moves your cursor between opening IF, REPEAT, and SPAN instructions and closing END IF, END REPEAT, and END SPAN instructions. This can help you troubleshoot problems where one of the instructions is accidentally moved or deleted.
Go to Field ≣₊∕	Moves you cursor to a specific variable or instruction field in the template. This can be useful when you receive errors and need to quickly locate the field in the template containing the error.
Previous Field / Next Field 4/ >	Allows you to navigate through the template, field by field. This may be useful if you have several pages in a template without any variable fields—navigating by



fields allows you to skip these pages, rather than scrolling through them until you find the next field.

HotDocs buttons not on the Word Ribbon

HotDocs buttons not available on the Word Ribbon can be accessed from the drop-down menu below the **dHotDocs** button. They are:

Button Name	Description
Dot Codes	Under the Dot Codes heading, you can select from four sections: Character Format , Character Insertion , Sentence Punctuation , and List Punctuation . Clicking on any of the options within those sections will insert the corresponding dot code into the template.
Markup View / Developer View	Switches the look of the template between Markup View and Developer View . Markup View provides a simpler view of the template (which can then be saved and shared with a non-HotDocs user who needs to review the content of the template, for example). Developer View is the mode in which you automate the template.
	Switching between Markup and Developer Views is only supported for RTF templates.
HotDocs Outliner	Opens the HotDocs Outliner so you can review an outline of scripting (and variables) used in the template.

You should always use the HotDocs \square Save and \boxtimes Save and Close buttons to save and close a template rather than the word processor's buttons. The HotDocs buttons ensure the template and component files are properly saved and closed.

If Word opens to the default HOME tab, click on the HotDocs tab to start using HotDocs commands.

At a Glance: The New Template Dialog Box

ew Template	? ×
Type: Word DOCX Template (.docx)	•
File name:	
	<u> </u>
Target folder:	
C: \Users\HolliCooper \Documents \CDocs \Templates \	
Title:	
Description:	
•	
6	
3	
E Shared component file:	
E Shared component file:	
Shared component file:	
Shared component file: Initial contents	•
E Shared component file: Initial contents © Empty G	•
E Shared component file: Initial contents © Empty © Current open word processor document	•
E Shared component file: Initial contents Empty Current open word processor document Other file:	•
E Shared component file: Initial contents Empty Current open word processor document Other file:	• •
E Shared component file: Initial contents © Empty © Current open word processor document © Other file:	•
E Shared component file: Initial contents © Empty © Current open word processor document © Other file:	▼ Cancel

To automate document generation using HotDocs, you first need to create a template, the base file from which HotDocs generates all future documents.

You can create a new template by clicking the *** New Template** button on the HotDocs Library toolbar. This opens the **New Template** window.

Using the **Type** drop-down list A you can select the type of template you want to create, including text templates, form templates, interview templates, and clause libraries. When you select one of these options, HotDocs attaches the correct file extension to the file name.

In the **File name** text field **B** you can type a file name for the new template, if you decide to store the template somewhere other than the default template folder (as seen in the **Target folder** field), you can also type the full path for the new destination. Alternatively you can use the **Target folder** for the select a file name and destination for your new template. When you do this, the **Target folder C** shows the full path for the new template location.

The first time you type a file name for the template, when you click into the **Title** text field **D**, HotDocs copies the file name as the template title; however, you can edit the title if you want. After, the title appears in the HotDocs library.

In the **Description** field **3** you can enter an optional description of the template. Descriptions appear when you select the library item and view its properties.

When using several related templates that share multiple components, it is best practice to share component files to save time. Sharing component files minimizes the chance of user error when editing multiple templates. You can add an existing component file to the new template by selecting one from the **Shared component file** drop-down list **a**. The component file you choose gets shared between the new template and any other templates you previously shared it with.

Using the **Initial contents** option buttons **G** you can choose the initial contents of the new template:

- **Empty:** Creates a new, empty template. Once HotDocs has created the template, you can add text, variables, and other template components.
- **Current open file:** Copies the current, open word processor file into the new template. You must have a word processor file open before you select this option.
- Other file: Creates a new template based on the contents of an existing template or document. The text field shows the folder path and file name of the file whose contents you want to copy into the new template. Either type in the full path or click the Browse button to locate and select the template or document whose contents provide the basis of the template you are creating. Selecting an existing template here disables the Shared Component file option as the new template shares the same component file as the source template.

To learn more about creating new templates follow the links below:

- Creating a New Text Template File
- Creating a Text Template Based on an Existing Template
- Converting a Single Template to a New File Format
- Using One Component File for Multiple Templates

Creating a New Text Template File

A text template is a word processor file that contains text as well as HotDocs components, such as variables, dialogs, instructions, scripts, and formats. The first step in automating a HotDocs text template is to create a new template file that contains your document text.

Do not create your templates in the tutorial library that installs with HotDocs. If you ever reinstall HotDocs, the library will be overwritten and you will have to add your templates to the library again. Instead, create a new library for your templates.

To create a new text template

- 1. At the HotDocs template library, click the folder in which you want the template.
- 2. Click **New Template**.
- Select the type of text template you want to create from the **Type** drop-down list. Depending on which word processors are installed on your computer, your options include **Word DOCX Template (.docx), Word RTF Template (.rtf), Word Document Template (.dot)**, and WordPerfect Template (.wpt).
- 4. In the File name field type a file name for the new template. If you decide to store the template somewhere other than the default template folder (as seen in the Target folder field), you can also **Browse** to the new destination. HotDocs adds the correct extension to the file name based on the type of text template you selected. HotDocs displays the full path of the currently selected template location in the Target folder field.

Do not type a template name that includes a pound sign (#). This can cause errors when uploading.

You can check the default *Templates* location at the HotDocs Options window.

- 5. Type a title for the template in the **Title** field (or accept the suggestion HotDocs makes). The title is what identifies the file in the library.
- 6. Optionally, enter a description in the **Description** field. Descriptions appear in the **Properties** tab of the library when the user selects the template.
- 7. To add an existing component file to the new template, you can select a component from the **Shared component file** drop-down list. If you do so, the component file you choose is now shared between the new template and any other templates using that component file.
- 8. Select the contents for the new template by choosing from the following options:
 - **Empty:** HotDocs creates a new, empty template. Once HotDocs creates the template, you can add text, variables, and other template components.
 - **Current open file:** HotDocs copies the current, open word processor file into the new template. You must have a word processor file open before you select this option.
 - Other file: HotDocs creates a new template based on the contents of an existing template or document. The text field shows the folder path and file name of the file whose contents you want to copy into the new template. Either type in the full path or click the Browse button to locate and select the template or document whose contents provides the basis of the template you are creating. If you select an existing template here, HotDocs disables the Shared Component file option as the new template shares the same component file as the source template.

9. Click **OK** at the **New Template** window. HotDocs opens your selected word processor and displays the template. You can use the HotDocs editing toolbar to begin automating the template.

If you are using Microsoft Word to develop your templates, we recommend that you create **DOCX templates.** HotDocs doesn't have to convert DOCX templates so they assemble much faster than RTF templates and are less likely to encounter errors.

When you create a new, empty Word template, HotDocs applies whatever styles are in *HotDocs.dot* to the new template. When you create a new template based on another template or document, HotDocs applies the styles used in that template or document to the new template.

You can include **command-line options** (in the **File name** field) that allow you to specialize the assembly process for a given template.

If you have multiple word processors installed, you can have HotDocs always suggest a specific template type when you create a new template.

If you need to use your template with earlier versions of HotDocs, you must specify which version at **Component File Properties**.

HotDocs does not support templates with embedded mail merge data. You need to remove this data before assembling the template.

Creating a Text Template Based on an Existing Template

You can create a new text template based on the contents of a text template already in the library. When you create a new template based on another template or document, HotDocs copies the text of the existing template (including components) into the new template (and component) file and then applies the styles in that the existing template to the text of the new template.

Do not create your templates in the tutorial library that installs with HotDocs. If you ever reinstall HotDocs, the library will be overwritten and you will have to add your templates to the library again. Instead, create a new library for your templates.

To create a new template based on an existing one

- 1. At the HotDocs library, select the existing template you want to use from the template list.
- 2. Click **New Template**.

- 3. In the File name field type a file name for the new template. If you decide to store the template somewhere other than the default template folder (as seen in the Target folder field), you can also **Browse** to the new destination. HotDocs adds the correct extension to the file name based on the type of text template you selected. HotDocs displays the full path of the currently selected template location in the Target folder field.
- 4. Click in the **Title** field and either accept the suggested title or replace it with your own. You can also add an optional template **Description**.

Do not change the existing template's file path and name in the **Other file** field.

5. Click **OK**. HotDocs opens a new template file and copies the text and variables from the **Other file** into it. It also creates a new component file and copies the existing components into it.

You can also create a new template and copy that template to a new location by using the **Copy Templates** command. Once you have copied the template and component file, rename the files using Template Manager.

HotDocs does not support templates with embedded mail merge data. You need to remove this data before assembling the template.

Converting a Single Template to a New File Format

To convert multiple files at once or to convert all templates to .DOCX, use Template Manager. (See Convert Multiple Templates to Work with HotDocs 11 and Convert Templates and Clauses to Microsoft DOCX.)

HotDocs 11 uses a file format for component files that is different from earlier versions of HotDocs. To take full advantage of the features available in this latest version of HotDocs, you must convert your component file to the current format. However, if users of your templates are still using previous versions of HotDocs, you can keep the files in their existing formats.

Additionally, perhaps you've changed word processors and need to convert all your templates from one file type to another.

It is a good idea to back up templates, component files, or other HotDocs files before making major changes to them.

To convert an older HotDocs template to the latest version of HotDocs

- 1. At the template library, select the template you want to convert and click **Edit Template**. HotDocs displays a message box asking how you would like to convert the template.
- 2. Make your choice, based on the following options:
 - Click **Convert to Current Format** to convert the template to HotDocs 11 format. If you select this option, the format shown at **Component File Properties** will continue to be set to **Current**.
 - Click **Keep Existing Format** to leave the component file in the format it is already in. If you select this option, HotDocs will set that specific format for the component file at **Component File Properties**.

If you select **Keep Existing Format**, HotDocs will stop Template Manager from converting this template's component file to another format. If you want to convert it later you will need to use the option in **Component File Properties**.

• Click **Cancel** to close the message box and cancel the conversion process.

If you need to keep several templates in their existing pre-2009 format and you don't want to be prompted to make this selection on a template-by-template basis, you can use Template Manager to change the component file format for several files at once. See Modify Component File Properties Across Multiple Files for details.

If you are converting a HotDocs 5 template, you should verify the template works correctly in HotDocs. For details about this, see Verify HotDocs 5 Templates for Use with HotDocs 11.

After selecting **Keep existing format**, when prompted by the Component Manager, HotDocs will then only allow format changes to be made at the Component manager and will not enact changes made to the component file in Template Manager.

To convert a template to a new word processor format

- 1. Select the template at the HotDocs library and click **New Template**. The **New Template** dialog box appears.
- 2. Click the **Type** drop-down button and select the new word processor. The file name extension in the **File name** field is changed to the new word processor format.
- 3. Optionally, change the **Title** and **Description** of the template.
- 4. Do not change the information in the **Other file** field.
- 5. Click **OK**. HotDocs converts the template to the new format and displays it in the appropriate word processor. The quality of the template conversion depends on your word processor's conversion capabilities. Complex text formatting, unusual fonts, and so forth, may not be converted correctly. You should check new versions of templates for conversion errors.

Once you have updated your template, the old template may still be listed in the library. This happens when old and new versions of the template have different file name extensions. Delete the old version from the library. (See Remove Items from a Library.)

Since the release of HotDocs 11.1, HotDocs no longer supports converting component files to pre-2009/10 formats.

Edit a Template

Once you have automated a text or form template, you may need to make changes to your work. You can edit variables, dialogs, instructions, and other features.

To edit a template

- 1. At the HotDocs library, select the template and click **Edit**. The template opens and the **HotDocs Edit** toolbar appears.
- 2. Once the template is open, you can edit variables, custom dialogs, REPEAT instructions, INSERT instructions, and IF instructions.
- 3. When you finish, click the **Close Template** button to save your work and close the template.

If Word opens to the default HOME tab ribbon, click on the HotDocs tab to start using HotDocs commands.

HotDocs does not support templates with embedded mail merge data. You need to remove this data before assembling the template.

Add HotDocs Toolbar to WordPerfect Templates

Sometimes you may need to manually add the HotDocs navigation toolbar to WordPerfect.

To add the toolbar

- 1. Select the template in the library and click the **New Template** button. The **New Template** dialog box appears.
- 2. Don't make any changes to any of the template's properties and click OK.
- 3. Click **Yes** when asked to convert the file.

At a Glance: The Apply Colors dialog box

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	
— Color scheme ——— C <u>N</u> one	ОК
	Cancel
C Nested	
C Sequential	

After opening a template to edit, you can open the **Apply Colors** dialog box by clicking on the **PApply Colors** button on the HotDocs Ribbon.

Using the multiple choice list, you can choose the color scheme HotDocs will use for variables and Instructions. You can choose from **None**, **Regular**, **Nested**, and **Sequential**. You can set the colors HotDocs uses in each color scheme in the **HotDocs Options** window see Define Field Colors)

To learn more about assigning colors to fields follow the links below:

- Assign Colors to Fields and Instructions in Templates
- Define Field Colors

Assign Colors to Fields and Instructions in Templates

Click here if you are using WordPerfect and the HotDocs Navigation toolbar doesn't appear in your template.

You can assign custom colors to the different types of fields in a template. This can help you more quickly identify sections or parts of your template as you are automating. Additionally, you can assign colors to IF, REPEAT, and SPAN instructions in your templates. Assigning different combinations of these colors can help you differentiate levels of nesting as well as help you better view individual pairs of instructions.

By default, HotDocs assigns its own colors; however, you can customize the colors at the **HotDocs Options** dialog box. (See Define Field Colors.)

To apply custom colors to template fields and instructions

- 1. Open the template for editing. (See Edit a Template.)
- 2. Click the **PApply Colors** button. The **Apply Colors** dialog box appears.

- 3. Select one of the following options:
 - Choose None to make all variable and instruction fields black.
 - Choose **Regular** to mark all variable and instruction fields using a custom color specific to that field type. For example, all variables will be marked with one color, all IF instructions will be marked with a different color, all REPEAT instructions will use a different color, and so forth.
 - Choose **Nested** to mark each level of IF and REPEAT instructions using a custom color. (For example, all first-level IF instructions will use a specific color, while all second-level instructions will use a different color, and so on.)
 - Choose **Sequential** to mark each IF and REPEAT instruction field using a custom color. (For example, the first instruction in a template will be marked using one color, while the next instruction will be marked using a different color, and so on.)

If you change your default colors, or if you receive templates from other developers whose colors are different from yours, you can click the **PApply Colors** button to update existing templates with your custom colors.

If you're using Microsoft Word, you can also apply colors by clicking on the **HotDocs** dropdown menu in the HotDocs toolbar and choosing **Apply Colors**. (To access this menu button in Word 2007 and later, click the **HotDocs** drop-down button.)

Move Between Fields in a Template

Click here if you are using WordPerfect and the HotDocs Navigation toolbar doesn't appear in your template.

You can navigate through the template, field by field, using the **Next Field** and **Previous Field** buttons in the HotDocs Navigation toolbar. This may be useful if you have several pages in a template without any variable fields—navigating by fields allows you to skip these pages, rather than scrolling through them until you find the next field.

To move between fields in the template

- 1. Open the template for editing. (See Edit a Template.)
- Place your cursor somewhere in the template and click either the ▶Next Field or ◀Previous Field button. HotDocs moves your cursor to the next or previous field in the template and highlights it. (If you're using Word 2000 or later, you can quickly edit the variable field by doubleclicking in the field.)

If you're using Microsoft Word, you can also move between fields by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Next Field** or **Previous Field**.

At a Glance: The Label Fields dialog box



After opening a template to edit, you can open the **Label Fields** dialog box by clicking on the **Cabel Fields** button on the HotDocs Ribbon.

There are three check boxes in this dialog that you can select to set field labelling options:

- Label IF, REPEAT, and SPAN instructions: Causes HotDocs to label IF, REPEAT, and SPAN fields. A label can help you identify matching instructions within a series of instructions. The label that is assigned is a number. For example, HotDocs labels the first REPEAT instruction in the template with the number 1, the next REPEAT instruction with a number 2, and so forth.
- Identify level of nested IF, REPEAT, and SPAN instructions: Causes HotDocs to insert the level
 of the IF, REPEAT, or SPAN instruction in the field's label. If the field is nested, the level number is
 incremented. Levels are identified by the letter L. For example, if labeling the third level of a
 REPEAT instruction, the text L3 is merged in the label.
- Place comments (including field labels) in hidden text: Causes HotDocs to apply the word processor's hidden text property to field comments and labels. When you click OK, everything in the comment field is hidden. To view the comments again, you can either clear this option, or you can use the word processor's command for showing/hiding hidden text.

You can remove any of these options by deselecting the check boxes.

To learn more about labelling fields follow the link below:

• Use Labels to Identify Instructions

Use Labels to Identify Instructions

Click here if you are using WordPerfect and the HotDocs Navigation toolbar doesn't appear in your template.

You can assign labels to IF, REPEAT, and SPAN instructions to help you identify pairs or groups of instructions in a text template. When you label fields, you can choose to merge just a numeric name in a field or you can choose to merge a name as well as the level of nesting. This can help you match opening instructions with closing, as well as help you determine the level of nesting within an instruction.

Field labels are merged as comments in the field.

The following example shows a series of nested REPEAT instructions that have been labeled. The first instruction, *REPEAT Editor Information*, is labeled with the number 1, since it's the first REPEAT in the template. This same instruction is also assigned the level number of *L*1, since it's the first level of the REPEAT instruction. The *END REPEAT* is likewise labeled so that you can easily match them when examining the contents of the template.

For each subsequent instruction, HotDocs increments the field number. Additionally, if the next REPEAT instruction is nested (which, in this example, it is), HotDocs increments the nesting level number. Each END REPEAT instruction is likewise labeled.

«REPEAT Editor Information // [1:L1]» Editor: «Editor First Name» «Editor Last Name» «REPEAT Author Information // [2:L2]» Author: «Author First Name» «Author Last Name» «REPEAT Book Information // [3:L3]» Title Edited / Date Completed: «Book Title» / «Date Completed:03 JUN 90» «END REPEAT // [3:L3]» «END REPEAT // [2:L2]» «END REPEAT // [1:L1]»

When assigning labels to fields, you can choose whether to include both the field number and the nesting level. You can also choose to merge just one or the other.

Finally, you can have HotDocs assign the hidden text property to these labels (as well as any other comments you've assigned to fields in the template.) When this property is assigned, comments and labels will be hidden and shown, depending on whether you are viewing hidden text in the template. (For information on viewing hidden text, see your word processor's documentation.)

Repeats can not be nested more than four levels deep.

To label instructions

- 1. At the text template, click the *A* Label Fields button. The Label Fields dialog box appears.
- 2. Select an option, based on the following information:
 - To assign a numeric name to each instruction, select **Label IF, REPEAT, and SPAN** instructions.
 - To have HotDocs identify the level of nesting for each instruction, select **Identify level of nested IF, REPEAT, and SPAN instructions**.
 - To apply the hidden text property to comments and labels, select **Place comments** (including field labels) in hidden text. Once applied, you can quickly show and hide your comments and labels by choosing the appropriate command at the word processor.
- 3. Click **OK.** HotDocs applies the labels you have selected to any instructions in the template.

SPAN fields are supported in Microsoft Word only.

To clear labels from fields or show comments once again, click the *C***Label Fields** button, clear all of the options, and click **OK**.

If you're using Microsoft Word, you can also label fields by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Label Fields**.

At a Glance: The Go To Field dialog box

Go To Fie	ld	X
Field num	ber:	
		1

After opening a template to edit, you can open the **Go To Field** dialog box by clicking on the \blacksquare +Go To Field button on the HotDocs tab.

You can enter a number in the **Field number** field, click the **OK** button, and HotDocs will move the curser to the corresponding variable field on the template. For example, if you typed in the number 3 and clicked the **OK** button then HotDocs would place your curser within the third variable field from the top of the template.

To learn more about the Go To Fields button follow the link below:

• Go to a Specific Field in the Template

Go to a Specific Field in the Template

As you test assemble your templates, you may occasionally receive HotDocs error messages that instruct you to correct a problem in a specific field in the template. If you know the field number, you can have HotDocs move your cursor to the specific field in question.

To move your cursor to a specific variable field

- 2. Enter a number in the **Field number** field.
- 3. Click **OK**. Your cursor is taken to that field.

View the Template in Markup View

Moving between Markup View and Developer View is supported in Microsoft Word RTF templates only. In Microsoft Word DOCX templates you can click the Create Markup button to open a Markup document in a new Word window.

As you automate a Microsoft Word RTF template and create within it variable and instruction fields, the template can be complicated to read and understand, especially to someone unfamiliar with automation. At times, however, you may need for a subject matter expert to review the text of the template for accuracy or to make changes or edits. To make the template easier for a non-HotDocs user to review, you can change the formatting of the template to Markup View and then provide the reviewer a copy of it.

When displaying markup, HotDocs replaces variables and instructions in the template with markup fields, which include markers (such as brackets) and a name. How the resulting markup field looks depends on the type of field it is:

- **Variables:** By default, variables are marked using brackets. If a comment is assigned to a variable, it will be used as the field name. If no comment is assigned, HotDocs will use either the variable's title or name.
- **IF, REPEAT and SPAN instruction blocks:** By default, IF, REPEAT, and SPAN instruction blocks are marked using brackets. If a comment is assigned to these text blocks, the comment can be merged in the field label. If no comment is assigned, HotDocs will simply merge the field markers around the text block. The instruction text can appear as an annotated footnote or endnote (depending on your preferences). (By default, SPAN instructions are not included in the Markup View. You can select an option to display them, however.)
- **INSERT instructions:** HotDocs will merge the file name of the template you are inserting.

All other instructions (such as ASK, LANGUAGE, etc.) will be hidden when the template is changed to Markup View.

Once you finish automating a template and switch to Markup View, you can save the template as a document file and provide a copy of it to the reviewer. Please note, however, that when you (or the reviewer) are viewing a marked up template (in HotDocs), you cannot edit existing variable fields or instructions—you must switch back to Developer View to edit those fields.

To view a template in Markup View

- 1. Define the Markup View options you want to use in the template. (See Specify How Documents Should Be Marked Up.)
- 2. Edit the template you want to mark up. (See Edit a Template.)
- 3. Click the drop-down menu below the **dHotDocs** button.
- 4. Select the **Markup View** option in the drop-down menu. The template changes to show the marked up template.

To view a developer version of the template again, click the **Developer View** option.

You can also view a marked up version of your assembled document. See View an Assembled Document in Markup View for details.

Sometimes as you automate a template, you create Computation variables that perform some function, rather than merge values. To keep these types of Computation variables from appearing in a marked up document, enter **NONE** as the variable's title.

At a Glance: The HotDocs Outliner dialog box



In Word 2003 the **HotDocs Outliner** dialog box will appear as a separate window to Word but can be viewed side by side with the Word window by clicking the **Arrange** button.

After opening a template to edit, you can open the **HotDocs Outliner** dialog box by clicking on the **HotDocs Outliner** button on the from the drop-down list below the **HotDocs** button on the HotDocs tab of the Word ribbon. The Outliner opens within the Word window and this will display all HotDocs instructions in an outline form, without the main text or formatting of the document.

At the top of this dialog is the Outliner toolbar A where you can access the following options:

- **Expand All:** Expands the list of items in the HotDocs Outliner to show all levels of instructions and variables.
- **Collapse All:** Collapses the list of items in the HotDocs Outliner to show only first-level instructions and variables.
- **Refresh Outline:** Updates the HotDocs Outliner with any changes you've made to text in the template.
- **Synch Template:** Lets you view the template's cursor location in the outline. For example, when Synch Template is on, you can click in the template and have HotDocs show you where in the outline your cursor is positioned.
- Show Variables: Includes variables in the list of instructions. Viewing variables in the context of their instructions can help you better understand scripting in the template.
- **Provide a state of the apply Colors** dialog box, where you can choose which coloring scheme should be used to identify variables and instructions in the outline.
- **Whelp:** Opens the relevant page of the HotDocs Help File.

Underneath the tool bar is a check box and drop down list where you can choose to limit the outline shown in the instruction list by selecting to view single variables. This lets you view a specific variable to see which instructions or conditions are associated with that specific variable. If you select **<cursor position>** from this list, you can click in a field in the template and view the same information. This helps you view scripting on a variable by variable basis. If you don't tick the check box then the instruction list will show an outline of all the scripting and variables used in the template.

To learn more about the HotDocs Outliner follow the link below:

• View an Outline of Scripting in the Template

View an Outline of Scripting in the Template

The HotDocs Outliner is supported in Microsoft Word 2000 and later only.

Using the HotDocs Outliner, you can generate an outline of scripting in the template. This outline can include just a list of instructions used in the template, or it can include variables as well. Items in the outline appear in the same order as they are used in the template. Viewing this outline may help you better understand the logic used in the template.

You can expand the outline to view all nested instructions in the template. You can also click on entries in the outline and view where that particular variable or instruction is merged in the template (and vice-versa). You can also view a single variable or instruction and see just the conditions or instructions surrounding that particular instance.

To view a scripting outline

1. Edit a Word text template. (See Edit a Template.)

- 2. In the **HotDocs Edit** toolbar, click the drop-down menu below the **HotDocs** button, then select the **HotDocs Outliner** button. The **HotDocs Outliner** window appears.
- 3. Perform any tasks, as described in the following table:

То	Do This					
View all instructions in the template—including all nested instructions	Click the $\mathbf{\overline{\xi}}$ Expand All button. The outline expands to show all levels of instructions.					
	To collapse the interview outline, click the T Collapse All button.					
Update the Outliner after making changes in the template	Click the Refresh Outline button. The outline updates and displays any changes you've made to the template text.					
View the template's cursor location in the outline	Click the Synch Template button. Now, when you click in the template, entries in the outline are highlighted to show where in the outline your cursor position is					
View variables used in the template	Click the Show Variables button. The outline expands to include all variables merged in the template.					
	Viewing variables allows you to see relationships between variables and any surrounding instructions.					
Apply color coding to instructions in the outline	Click the PApply Colors button. The Apply Colors dialog box appears, where you can choose the color scheme you want to use in the outline:					
	Choose None to make all variable and instruction entries black.					
	 Choose Regular to mark all variable and instruction entries using a custom color specific to that field type. For example, all variables will be marked with one color, all IF instructions will be marked with a different color, all REPEAT instructions will use a different color, and so forth. 					
	 Choose Nested to mark all instructions using a custom color. Where pairs of REPEAT, IF, or SPAN instructions are nested, each level of nesting will be marked with its own color. 					
	 Choose Sequential to mark all instructions using a custom color. Each pair of REPEAT, IF, and SPAN instructions in the template will be marked by a different color. 					

Copy the contents of the script outline to the Windows Clipboard so you can paste it into another application	Click the Copy to Clipboard button. The entire outline is copied. Open another application and then paste the script using that application's Paste command.
View all conditions or other instructions surrounding the inclusion of a specific variable in the outline	Select Limit to and then click the drop-down button to select the variable you want to view. The outline changes to show any REPEAT or IF instructions that affect that particular variable.
View the conditions or other instructions surrounding the cursor position in the template	Select Limit to and then choose <cursor position=""></cursor> from the drop-down list. Click in the template where you want to view these instructions or conditions.
Edit or test an instruction or variable in the outline	Right-click on the instruction or variable in the outline and choose Edit or Test .
Have HotDocs show you where in the template a REPEAT, IF, or SPAN instruction's END instruction is inserted	Right-click on the instruction in the outline and choose Find Matching END . In the template, the instruction is highlighted.

In Word 2003 the **HotDocs Outliner** dialog box will appear as a separate window to Word but can be viewed side by side with the Word window by clicking the **Arrange** button.

If you wish to use the **HotDocs Outliner** in Word 2007 it is recommended that you install SP3 or higher.

Create a Foreign Language Template

You can automate templates in languages other than English. There are requirements for doing so, however. Specifically, you must create a LANGUAGE instruction in the template that, in turn, calls a special runtime DLL that lets you format dates and numbers in the template so they appear correctly in the assembled document.

When formatting date and number variables in your template, you must manually enter example formats in the language you have chosen for your template. If you attempt to use English-language formats, your date and number variables will not be processed correctly.

HotDocs supports formatting dates and numbers only in the languages listed below. If you are automating templates in a languages other than these and need specially formatted dates or numbers, you must use Text variables to represent these variables.

To create a template using a foreign language instruction

- 1. Create your template, using document text in the language you have chosen.
- 2. Create variables and dialogs, using variable names and prompts in the language you have chosen.
- 3. When inserting Date and Number variables, type the example format—in the foreign language in the Format field. For example, if you have a French Date variable, you would type 3 Juin 2000. (Where you assign this format—either at the Date Variable Editor or at the Variable Field dialog box—depends on your project. See Format the Variable for ideas.)
- 4. Position the cursor in the template where you want the LANGUAGE instruction to take effect.
- If you are using Microsoft Word, click the HotDocs drop-down menu in the HotDocs toolbar and choose Other Field from the list of options. The Other Field dialog box appears. (WordPerfect users, see step 10.)
- 6. Click the Field type drop-down button and choose LANGUAGE.
- 7. Click the **Language** drop-down button and choose the language you are using for the template. Your options include:

Language	Code
English	ENG
German	DEU
Swiss German	DES
Austrian German	DEA
French	FRA
Dutch	NLD
Spanish	ESN
Italian	ITA
Brazilian Portuguese	PTB

- 8. Optionally, enter the punctuation character you want to use for the **Thousands separator** and the **Decimal separator** in the appropriate fields.
- 9. Click **OK**. The instruction is merged in the template.
- 10. If you are using WordPerfect, copy and paste an existing variable and then replace the text between the chevrons (« ») with the language instruction. For example: «LANGUAGE FRA».

If you want to include a chevron in a HotDocs text template either as a part of your document text or a literal text string, you need to double the opening chevron (but *not* the closing chevron), so that HotDocs does not interpret the chevrons (and any text between them) as a field. The output is a single chevron, as follows: input: «some text», output: «some text».

Once the instruction is in the template and the document is assembled, all Date and Number variables after the LANGUAGE instruction are processed in that specific language.

Only the Spanish language DLL recognizes the gender-specific use of the word *one*. Options for that language include *uno*, *una*, and *un*.

The LANGUAGE instruction only affects date and number formats. You can use the default formats for the other types of variables or create your own. For example, if you want to use a Spanish True/False variable format, type **Sí/No** in the **Format** field.

Automating Plain Text Templates

You create or edit Plain Text Templates using your systems default text editor (the program which opens for .txt files). You can change what program you use by changing the file association for .ttx files in the Control Panel.

Because you can use any number of text editing programs to automate a Plain Text Template HotDocs does not integrate closely with your text editor in the same way it does for word processor templates. Instead HotDocs allows you to bring up the Component Manager for a Plain Text Template's component file and use this to create and edit components the same way you would with word processor templates.

Some text editors will allow you to drag and drop components directly from the Component Manager into the template. For other text editors, and for other types of fields (such as IF, INSERT, REPEAT etc) you must edit directly in the text editor.

When creating a Plain Text Template from an exiting RTF or DOCX template it is important to check all components have been transferred properly. Check all END instructions have been correctly placed in the new template before assembling.

Tips on Using Graphics in Word Templates

When automating templates, you often need to include graphics in the template. These graphics can represent company logos, illustrations, signatures, and so forth.

As you work with these files in the template, you may notice that their inclusion adversely affects template size and performance. A template without any graphics can easily jump in size from, say, 100K to 4 MB when you add a graphic file. This may happen because the size of the original graphic file is large even before you insert it, or it may happen because Word is copying the graphic and inserting it again as a metafile object. Additionally, sometimes graphics you insert in a header/footer may inadvertently be left in the header/footer, even if you think you have removed them.

Word stores any graphics in a .DOCX file in a compressed format that is close to the size of the original imported image. The RTF format, however, is a character-based document format, so

graphic images must be converted to character strings. When images are large, have high resolution, or are of certain difficult-to-represent formats (particularly bitmaps), they become very large in RTF files.

Proper use of graphics in templates can minimize and even eliminate many of these problems. The following sections describe some steps you can take to improve your experience.

Minimize the File Size of the Graphic Outside of Word

When adding graphics to your template, you should do all you can to minimize the size of the file before you insert it into the document. The following tips should help:

- Save the file in one of the following compressed formats: JPG, PNG, or GIF. Which format you choose depends on the type of graphic you are inserting. For example, if you are inserting a picture, you should use JPG. For simpler images, like illustrations or icons, you can use GIF. Do not insert uncompressed files such as Bitmaps (.BMP) in the template.
- Save graphic files in the lowest acceptable resolution. Resolution for an image printed on a laser printer should display well at 120 dpi (or dots per inch), but some may wish to go as high as 150 dpi.
- Crop and/or resize the file using a graphics-editing application. Even though Word includes tools for cropping or resizing the graphic directly in the document, these tools do not reduce the actual size of the file that is embedded in the template.

Keep Word from Storing Metafile Versions of Your Graphics in RTF

According to a Microsoft Knowledge Base article, "if an EMF, a PNG, a GIF, or a JPEG graphic is inserted into a Word document, when the document is saved, two copies of the graphic are saved in the document. Graphics are saved in the applicable EMF, PNG, GIF, or JPEG format and are also converted to WMF (Windows Metafile) format." Graphics stored in an RTF file in Windows Metafile format can be extremely large.

To resolve the problem, you can specify a Windows registry setting that keeps Word from saving two copies of the image.

You must be extremely careful when working in the Windows registry. Failure to follow the instructions below exactly could result in your making changes that negatively affect all of the programs on your computer. You may want to ask your system administrator for help if you are unsure of what you are doing.

To keep Word from saving two copies of graphics in your RTF templates

- 1. Close Word (if it's running.)
- 2. Click **Start > Run**. The **Run** dialog box appears.
- 3. In the **Open** field, type **regedit** and click **OK**. The **Registry Editor** opens.

- Navigate to HKEY_CURRENT_USER > Software > Microsoft > Office > Version > Word > Options. (Replace Version with the version of Word you are using.)
- 5. In the right pane of the window, right-click and choose **New > String Value** from the shortcut menu. A new string is created called **New Value #1**.
- 6. Rename the value to **ExportPictureWithMetafile**. (To do this, you can choose **Rename** from the shortcut menu.)
- 7. Once the name is changed, right-click on the string and choose **Modify**. The **Edit String** dialog box appears.
- 8. In the **Value data** field, enter **0**.
- 9. Click **OK**.

If templates already contain these extra metafile images, you can remove them using the Hidden Data Remover tool in HotDocs. See the HotDocs Help for more information on using this tool.

Use Graphics in Headers and Footers Correctly

Often, you design your templates using company letterhead. Frequently this letterhead includes a company logo or icon, which you choose to include on the first page of the document only. In this situation, you would typically select the **Different First Page** option, and then remove the graphic from any subsequent pages in the document. Problems come if you later decide you don't want a different first page header and you then clear the **Different First Page** option without first removing the graphic from the first page header. What happens is, Word assigns the headers from page 2+ to the first page so you can no longer see the graphic, but the graphic is still there. If you do not want graphics in your headers or footers, you must manually delete them before changing your header/footer options.

This same behavior can also happen when selecting the Different Odd & Even Pages option

Conclusion

In short, make sure you adhere to the following rules:

- 1. Use DOCX templates where possible or use an image file format that works well in RTF files. Generally this is JPG, although some types of images (e.g. line drawings) may be smaller in another format.
- 2. Size and crop images as required in the template before inserting them. Don't use Word's sizing/cropping or other picture manipulation features.
- 3. Use the lowest acceptable resolution for images (in order to decrease file size).
- 4. Insert images only where needed; avoid putting images in first page or odd/even headers that aren't used in the template.
- 5. Implement the registry setting that keeps Word from creating Windows Metafile Format copies of images in documents.

How Does HotDocs Assemble DOCX Templates?

In the simplest case, when assembling a DOCX file, HotDocs begins processing HotDocs fields at the top of the document and continues to the end in sequential order. However, deviations from this sequence may occur when HotDocs encounters certain items; such as file insertions, headers and footers, footnotes and endnotes, and comments.

Like any other field in a HotDocs template, HotDocs processes INSERT fields in the order encountered. This is also true if an INSERT field inserts a template, however HotDocs processes the inserted template before it is inserted, and before processing the rest of the fields in the receiving document. When the assembled sub-document is inserted at the point of the INSERT field, the headers and footers of its first and last sections are kept or discarded according to the KEEP instructions in the INSERT field.

The main body of a document is a series of one or more sections, and as expected, HotDocs processes the fields in each section in order. However, a section may include headers or footers. When HotDocs begins processing a section, if the section includes headers or footers, HotDocs processes the headers or footers before the remainder of the section. If a section includes multiple headers or footers (e.g. a different first page header), HotDocs processes the headers and footers in the order they are found. Since the author has no control over the order of the headers and footers in a DOCX file, the author should treat the order of one section's headers and footers as undefined.

Text in the document may contain references to comments, footnotes, and endnotes. When such a reference is found, HotDocs processes the content of the referenced item before continuing to process the rest of the document.

Inserting Variables in Text Templates

Introduction: Create and Customize Variables

Once your document text is in a template file, you must replace variable text (text that changes each time you assemble a document) with HotDocs variables. Examples of variable text include names, dates, pronouns, numbers, and calculations. A variable is what prompts HotDocs to ask for that information when the template is used to assemble a document.

You can create seven types of variables: Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information. The variable type determines what information users can enter when they complete an interview. For example, if you create a Date variable, users must enter a valid date—they cannot enter text, such as a name, for an answer.

The following graphic shows a paragraph that contains two items that change depending

on the user-the name (Aaron Jameson) and the date (12th day of October, 2002):

EMPLOYMENT AGREEMENT
This Employment Agreement, by and between Hobble Creek Publishing and Aaron Jameson, is entered into this 12th day of October, 2002.

You replace both of these items with variables:

EMPLOYMENT AGREEMENT

This Employment Agreement, by and between Hobble Creek Publishing and «Employee Name», is entered into this «Agreement Date:3rd day of June, 1990».

There are two parts to creating variables—creating the variable component, and creating the variable field:

- Variable Component: The variable component is the core part of a HotDocs variable. It contains all the information about a variable and how it should be processed, including the component name and prompt, any resource information, and any special patterns or other options that help control how the variable appears during the interview and how it will be processed when the final document is assembled. A variable component also creates an association between the components and the answers the user enters, so that the answer can be saved in an answer file. (Answer files can be reused when assembling other documents.)
- Variable Field: When you insert a variable into a HotDocs template, HotDocs creates a field where the user's answer is merged into the assembled document. In a text template, the field is denoted by HotDocs chevrons (« »), while in a form template, the field is denoted by a colored box that is overlaid on the form's static text. There are certain properties that are assigned to a variable's field that affect the way the variable is merged into the document—specifically, formats and merge text for Multiple Choice variables. You can also include fill characters that force the answer to be a certain number of characters, regardless of how long the answer is.

If you want to include a chevron in a HotDocs text template either as a part of your document text or a literal text string, you need to double the opening chevron (but *not* the closing chevron), so that HotDocs does not interpret the chevrons (and any text between them) as a field. The output is a single chevron, as follows: input: «some text», output: «some text».

When you insert a variable into the template, you automatically create a merge field for the variable. However, when you create a variable using Component Manager, there is not a merge field associated with the variable until the variable is inserted into the template.

At a Glance: The Variable Field dialog box

Variable type		
Text	True/ <u>F</u> alse	Personal Information
O <u>N</u> umber	Multiple Choice	
O <u>D</u> ate	O Computation	
ariable:	6	
Merge field prop	erties	
Format:		Non-breaking Vise default

After opening a template to edit, you can open the **Variable Field** dialog box by clicking on the **Variable** button on the HotDocs ribbon. You can also open this dialog box by double-clicking on an existing variable field to edit it.

From the multiple choice list A at the top of this dialog you can select the type of variable you would like to create:

- Text: Allows you to create a Text variable.
- Number: Allows you to create a Number variable.
- Date: Allows you to create a Date variable.
- **True/False:** Allows you to create a True/False variable.
- Multiple Choice: Allows you to create a Multiple Choice variable.
- **Computation:** Allows you to create a Computation variable. Click the Edit Component button to specify a computation script.
- **Personal Information:** Allows you to use or create a new Personal Information variable. Personal Information variables ask information about the user. HotDocs then stores the user's answer in the system registry so that information can be used in other documents. Once a question has been asked, it is not asked again.

The section below the text field can change depending on the type of variable that was selected from the multiple choice list A. If a Text, Number, Date, True/False, or Personal Information variable has been

selected you will see the **Merge field properties** as above. If you de-select the **Use Default** check box you can change the format of this specific use of the variable.

If a Multiple Choice variable has been selected then as well as the usual **Merge field properties** you will see a list of the Options, Prompts and Merge Text in this variable. If you de-select the **Use Default** check box here you will also be able to change the **Merge Text** used for this specific use of the variable.

Finally if a Computation variable has been selected then above the **Merge field properties** you will see the options for any Parameters associated with this variable and can edit them for use in this instance of the variable.

To learn more about inserting variables and variable fields follow the links below:

- Introduction: Create and Customize Variables
- Insert a Variable Field in a Text Template
- Create and Insert a Variable Using Component Manager
- Parameters

Insert a Variable Field in a Text Template

To insert a variable in your template text, you have two options: you can select a portion of the text that will change (such as a name, date, number, and so forth) and replace it with a variable field, or you can insert a variable field without selecting any text at all. If you select text, once HotDocs replaces the text with a HotDocs variable, it searches for other instances of the text and gives you the chance to replace those instances with the same variable. If you choose to simply insert a variable in the template without replacing template text, HotDocs inserts the variable only once.

To replace template text with a variable

- 1. Select the text you want to replace and click the ****Variable Field** button. The **Variable Field** dialog box appears.
- 2. Select the type of variable you want to insert. Your options include Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information.
- 3. Type a variable name in the **Variable** field. (If the variable already exists, select it from the dropdown list.) (See Tips on Naming Your Variables.)
- 4. Optionally, clear Use defaults and select an example format from the Format drop-down list. This format will determine how the answer appears in the assembled document. It is applied to this instance of the variable only. (See Format the Variable.)
- 5. Click the **Edit Component** button to modify the variable's component properties. (Click any of the following links for information on customizing a specific variable type: Text, Date, Number,

True/False, Multiple Choice, and Computation.) When finished making changes, click **OK**. The **Variable Field** dialog box appears again.

- 6. Click either **Replace Once** (replaces this instance of the variable only), or **Replace Multiple** (displays the **Find and Replace** dialog box where you can choose which instances you want to replace).
- 7. If you selected **Replace Multiple**, the **Find and Replace** dialog box appears.
- 8. Click the appropriate replacement option: **Replace**, **Replace All**, or **Find Next**. HotDocs replaces the text. (See Replace Other Instances of Selected Text with a Variable.)

WordPerfect users: When replacing multiple instances of text in a template, HotDocs will not replace those instances found in headers, footers, footnotes, and text boxes. You must manually replace those.

To insert a variable without replacing any text

- 1. At the HotDocs template, insert the cursor where you want the variable to be.
- 2. Click the ****Variable Field** button. The Variable Field dialog box appears.
- 3. Select the type of variable you want to insert. Your options include Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information.
- 4. Type a name in the **Variable** field. (If the variable already exists, select it from the drop-down list.) (See Tips on Naming Your Variables.)
- 5. Optionally, clear **Use default** and select an example format from the **Format** drop-down list. This format will determine how the answer appears in the assembled document. It is applied to this instance of the variable only. (See Format the Variable.)
- 6. Click the **Edit Component** button to modify the variable's component properties. (Click any of the following links for information on customizing a specific variable type: Text, Date, Number, True/False, Multiple Choice, and Computation..)
- 7. When finished making changes, click **OK** at both the **Variable Editor** and **Variable Field** dialog boxes. HotDocs inserts the variable.

You can assign text formatting—such as font, style, and size—to a specific occurrence of a variable. To do this, select the variable reference (including the chevrons) and use your word processor commands to assign the format. When the document is assembled, the text will be formatted the way you specify. If you must use a symbolic font, assign the font properties at the **Variable Field** dialog box. See **Control How Answers Appear in the Assembled Document**.

In Microsoft Word, you can also insert variables either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Variable Field**, or by right-clicking in the template and choosing **Variable Field** from the shortcut menu.

Be sure to use the relevant HotDocs field dialog (Variable Field, IF Field, etc.) when creating or editing fields in word processor templates. Editing field contents directly in Word must be done with care, since features like AutoFormat can substitute characters without your knowledge- for example replacing straight quotes with curly quotes, or hyphens with en-dashes. While HotDocs

tolerates some of these AutoFormat substitutions in RFT templates, it does not in DOCX templates. If you must edit the content of a HotDocs field directly, you should either turn off Word's AutoFormat feature or undo (using Ctrl+Z) the unwanted character substitutions it makes.

At a Glance: The Find and Replace dialog box

Find and Replace			? X
Fi <u>n</u> d what:			
	۵		
Replace with:		G	<u>M</u> odify
	B		
Match <u>c</u> ase Find	whole words only		
C_	Replace Replace A	I Eind Next	Close

After opening a template to edit, you can open the **Find and Replace** dialog box by clicking on the **Replace Multiple** button after creating a variable from template text.

In the top text field A you should see any text that was selected when you created your variable, but you can type in the word you would like HotDocs to find in the template if you need to change it. The name of the variable you have just created should appear in the second text field **B**. To edit the variable before replacing click the **Modify** button **C**.

Below the text fields are two check box options **D**. Tick the first one to have HotDocs make the search case sensitive and the tick the second field to have HotDocs only look for whole words only and ignore partial matches.

At the bottom of the dialog is a row of buttons where you can choose to **Replace** the current word and move on to the next match, **Replace All** matches without looking at each one, **Find Next** match without replacing or **Close** the dialog without replacing.

To learn more about using replacing variables follow the links below:

• Replace Other Instances of Selected Text with a Variable

Replace Other Instances of Selected Text with a Variable

When you replace text with a variable, HotDocs allows you to search forward through the template and replace other instances of the text with the variable. You can control this process.

To replace selected text with the variable

- Select the template text and create the variable. (See Insert a Variable Field in a Text Template.) When finished, click either **Replace Once** (replaces this instance of the text only), or **Replace Multiple** (displays the **Find and Replace** dialog box where you can control the replacement process).
- If you clicked **Replace Multiple**, the **Find and Replace** dialog box appears. The **Find what** field shows what text HotDocs is searching for. The **Replace with** field shows the name of the variable HotDocs is inserting.
- 3. Make a selection, based on the following information:

То	Do This
Replace this instance of text and move to the next instance	Click Replace .
Have HotDocs replace all instances of text from the selected instance to the end of the template (without confirmation)	Click Replace All .
Have HotDocs skip this instance and search for the next	Click Find Next.
Replace only those instances of text that are complete words (for example, prevent HotDocs from replacing the <i>his</i> in <i>this</i> with a variable for a pronoun)	Select Find whole words only.
Replace only those instances of text that have the same capitalization as the selected instance	Select Match case.
Change the properties of the variable	Click Modify and make the changes.

Create and Insert a Variable Using Component Manager

At times you might want to create a variable that might not appear in the template at all (such as a temporary variable that helps process a computation.) Or, perhaps you want to create several variables consecutively but you might not be sure where to use them in the template. You can use Component Manager to do this.

To create a variable using Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Select a variable type by clicking the **Components** drop-down button.
- 3. Click the **New Component** button. The **Variable Editor** appears to gather information about the new variable.
- 4. Assign a name and any other properties to the variable and click **OK**.

If you use Component Manager to create variables, you can likewise use Component Manager to insert the variables directly in the template.

To insert a variable into the template using Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Select the variable you want to insert from the component list. (To show only that certain type of variable in the list, click the **Components** drop-down button and select the variable type.)
- Either place your cursor in the template where you want the variable and click the ***Insert* Variable button (in Component Manager), or drag the variable from the Components list into the template.

Inserting Instructions in Text Templates

Introduction: Types of Insert Instructions

Using Insert instructions in your templates enables you control a wider range of features and automate much more complex documents.

There are four main types of instructions you can have in a HotDocs template.

IF Instructions: Makes part of a template conditional.

REPEAT Instructions: Creates lists within the document.

INSERT Instructions: Inserts one template into another.

SPAN Instructions: Allows the user to edit document text during the interview.

Understand How Smart Returns are Inserted After Instructions

By default, each time you insert any instruction (IF/END IF, REPEAT/END REPEAT, SPAN/END SPAN, INSERT, or ASK) in a template, HotDocs adds a return following the instruction. Then, when HotDocs assembles the template and processes the instruction, it removes the return it inserted. To some, this insertion of returns creates problems or confusion in automating the template because it sometimes makes the template text difficult to read and understand, especially when instructions (and their subsequent returns) within a paragraph sometimes break the paragraph into separate lines of text.

To accommodate this, HotDocs allows you to choose whether to insert returns following these instructions. (See Control When Returns Are Inserted After Instructions in Text Templates.). Specifically, you can:

- Always insert a return after an instruction.
- Never insert a return after an instruction.
- Use "smart" return insertion.

The first two options behave exactly as expected—HotDocs will either always insert a return (and remove it during assembly) or never insert a return, depending on which of the two options you choose.

The behavior of third option, using smart returns, depends on how and where you're inserting the instruction.

If text you select represents a complete paragraph (meaning there's a paragraph mark at the end of the selected text), HotDocs will insert a return character after the instruction. In the following example, brackets ([]) indicate where text selection begins and ends. Note that the closing bracket ends on the line after the paragraph mark (1):

Here is a paragraph.
[Here is another paragraph.9
] Here is a third paragraph.

Would produce a section of conditional text that looks like this:

Here is a paragraph. «IF Instruction» Here is another paragraph. «END IF» Here is a third paragraph.

However, if the text you select is within a paragraph, HotDocs will *not* insert a return character. So, in the following example, when you insert the REPEAT instruction, HotDocs does *not* insert a return because the

insertion is happening in the middle of a paragraph. There is no paragraph break at the end of the insertion:

The client, «Client Name», has the following children: «REPEAT Children:a, b, and c»«Child Name»«END REPEAT». These children live with the client and are covered under the provision.

At times, you may simply insert an instruction without first selecting text, or perhaps you may select some text but *not* include the paragraph mark in the selection. If HotDocs detects an existing paragraph break immediately after the instruction, it includes a Keep Return code (or vertical bar (|)) at the end of the instruction. The code tells HotDocs to *not* delete the return when it assembles the document. So, the following example, where brackets indicate text selection:

Here is a paragraph. [Here is another paragraph.] ¶ Here is a third paragraph.

would produce the following automation:

Here is a paragraph. «IF Instruction»Here is another paragraph.«END IF|» Here is a third paragraph.

Note that the Keep Return code () in the END IF instruction indicates to HotDocs to keep the return character when it assembles the document.

HotDocs' capacity to delete both an IF instruction and the return that follows it is very important in some situations. If an optional paragraph includes numbering, it is difficult to condition out the entire paragraph and its numbering without putting the IF on a separate line before the paragraph. It can be done, but the template looks odd, since the IF is inside the numbering of the paragraph.

Use Labels to Identify Instructions

Click here if you are using WordPerfect and the HotDocs Navigation toolbar doesn't appear in your template.

You can assign labels to IF, REPEAT, and SPAN instructions to help you identify pairs or groups of instructions in a text template. When you label fields, you can choose to merge just a numeric name in a field or you can choose to merge a name as well as the level of nesting. This can help you match opening instructions with closing, as well as help you determine the level of nesting within an instruction.

Field labels are merged as comments in the field.

The following example shows a series of nested REPEAT instructions that have been labeled. The first instruction, *REPEAT Editor Information*, is labeled with the number 1, since it's the first REPEAT in the template. This same instruction is also assigned the level number of *L*1, since it's the first level of the REPEAT instruction. The *END REPEAT* is likewise labeled so that you can easily match them when examining the contents of the template.

For each subsequent instruction, HotDocs increments the field number. Additionally, if the next REPEAT instruction is nested (which, in this example, it is), HotDocs increments the nesting level number. Each END REPEAT instruction is likewise labeled.

«REPEAT Editor Information // [1:L1]» Editor: «Editor First Name» «Editor Last Name» «REPEAT Author Information // [2:L2]» Author: «Author First Name» «Author Last Name» «REPEAT Book Information // [3:L3]» Title Edited / Date Completed: «Book Title» / «Date Completed:03 JUN 90» «END REPEAT // [3:L3]» «END REPEAT // [2:L2]» «END REPEAT // [1:L1]»

When assigning labels to fields, you can choose whether to include both the field number and the nesting level. You can also choose to merge just one or the other.

Finally, you can have HotDocs assign the hidden text property to these labels (as well as any other comments you've assigned to fields in the template.) When this property is assigned, comments and labels will be hidden and shown, depending on whether you are viewing hidden text in the template. (For information on viewing hidden text, see your word processor's documentation.)

Repeats can not be nested more than four levels deep.

To label instructions

- 1. At the text template, click the *C* Label Fields button. The Label Fields dialog box appears.
- 2. Select an option, based on the following information:
 - To assign a numeric name to each instruction, select Label IF, REPEAT, and SPAN instructions.

- To have HotDocs identify the level of nesting for each instruction, select **Identify level of nested IF, REPEAT, and SPAN instructions**.
- To apply the hidden text property to comments and labels, select **Place comments** (including field labels) in hidden text. Once applied, you can quickly show and hide your comments and labels by choosing the appropriate command at the word processor.
- 3. Click **OK.** HotDocs applies the labels you have selected to any instructions in the template.

SPAN fields are supported in Microsoft Word only.

To clear labels from fields or show comments once again, click the *CLabel Fields* button, clear all of the options, and click **OK**.

If you're using Microsoft Word, you can also label fields by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Label Fields**.

Match Opening Instructions with Closing Instructions

Click here if you are using WordPerfect and the HotDocs Navigation toolbar doesn't appear in your template.

Every IF, REPEAT, and SPAN instruction inserted in a template must include an END IF, END REPEAT, or END SPAN instruction. Without both of these instructions, you will receive an error when you attempt to assemble a document. (The one exception is a repeated row in a table. See Use a Word Processor Table to Display a List.) However, sometimes during the course of editing a template, you may inadvertently delete or move part of the instruction. Doing so will cause errors when you attempt to assemble the document. To troubleshoot errors like this, you can have HotDocs match the different parts of the instruction. Likewise, if you have an intricate section of nested instructions, you can pair these different parts of the instruction.

To match opening instructions with their closing instructions

- 1. Open the template for editing. (See Edit a Template.)
- 2. Place the cursor inside either the opening instruction or the closing instruction. For example, if you're trying to find the opening instruction of a REPEAT, place your cursor in the END REPEAT instruction.
- 3. Click the **Match Fields** button. HotDocs moves your cursor to the corresponding instruction.

When editing instructions in a computation script, you can place your cursor in either the opening or closing instruction and press **Ctrl+M**. This will move your cursor to the corresponding instruction.

If you're using Microsoft Word, you can also match fields by clicking on the **HotDocs** dropdown menu in the HotDocs toolbar and choosing **Match Fields**.

Using IF Instructions (Make Parts of the Template Conditional)

Introduction: Make Parts of Templates Conditional

Sometimes your templates contain optional text, or text that should be included only when certain conditions are met. To include or exclude text based on these conditions, either you create an IF instruction, which is based on a True/False variable, or you create an IF expression, which is based on other conditional tests. In either situation, the IF instruction or expression must result in a *true/false* (or *yes/no*) result.

For example, in an estate planning document, there may be a paragraph that lists children who stand to inherit part of the client's estate. However, if the client doesn't have children, this paragraph should not be included. To keep it from being merged in cases where it's not relevant to the client, you can condition the paragraph on a True/False variable that determines whether the client has children. If a simple True/False variable doesn't accomplish what you need, you can create a more complicated expression. Types of expressions include testing against which option of a Multiple Choice variable was selected, whether the value of an answer is equal to (or greater than or less than) another value, and so forth.

In this example, the sentence that begins "Premium rate adjustments due to change..." will be merged into the document only if the true/false variable *Step Rate Tables* is *TRUE*.

Also, the sentence beginning "Rates will be guaranteed..." and containing the computation variable *Rate Guarantee Calculation* will be merged into the document only when the answer to the variable *Rate Guarantee* is either *2 years* or *3 years*. If this condition is not met, *Rate Guarantee Calculation* will not be run.



In addition to using a single instruction or expression to merge or remove a single block of text, there may be times when you have multiple versions of a paragraph or other text, only one version of which should be merged into the assembled document. To accomplish this, you can use a series of ELSE IF instructions or expressions. If you have a paragraph that should be merged when no conditions are met, you can use an ELSE instruction.

In past versions of HotDocs, using END by itself instead of the complete END IF instruction was sufficient. Beginning with 11.1, HotDocs no longer processes the shortened version, and now requires the complete instruction.

At a Glance: The IF Field dialog box (IF True/False Variable)

🚱 IF Field	? 🔀
Field type	
True/False <u>v</u> ariable:	G •
Comment:	
Test Hide <u>A</u> dvanced	<u>O</u> K Cancel

After opening a template to edit, you can open the **IF Field** dialog box by clicking on the rightarrow **IF Field** button on the HotDocs ribbon. If you select **IF True/False Variable** from the multiple choice rightarrow options at the top of the dialog box then you will be presented with the following options below:

In the first text field **B** you can enter a name to create a new true/false variable and click the **dit** button **C** to set the option, or you can select an existing true/false variable from the drop-down menu. In the second text field **D** (visible if you click the **Show Advanced** button) you can enter a comment about the IF Field.

To learn more about using IF fields follow the links below:

- Introduction: Make Parts of Templates Conditional
- Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression
- Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions
- Edit an IF Instruction or Expression
- Nest IF Instructions
- Use Conditional Instructions in a Computation Script
- Tips on Using IF Instructions

At a Glance: The IF Field dialog box (IF Expression)

Field type			
IF True/ <u>F</u> alse Variable 🛛 💿 IF Exp	ession		
Expressio <u>n</u> : 🚿 🔊 o	4 B 🖪 🕅	∰ ås 4≣ ∰ ∰ ∰ 2 #	≡ G
Components: Available Component Agreement Date Agreement Information Annual Salary Company Representative Employee Gender Employee Gender Employee Information A Employee Name Employee to Complete Trial Period Employee to Receive Paid Seminar		ision models: r a Date r a Number r some Text r True or False DLUTE VALUE(NUM) (DATE) WERED(VAR) WERED(DIALOG) ING(NUM) WT(DIALOG) NT(MULT_CHOICE_VAR)	

After opening a template to edit, you can open the IF Field dialog box by clicking on the $rac{1}{2}$ IF Field button on the HotDocs ribbon. If you select **IF Expression** from the multiple choice options **A** at the top of the dialog box then you will be presented with the following options below:

Below the multiple choice options \underline{A} is the **Script Toolbar** where you can choose from the following buttons to search and edit the **Script** field \underline{B} :

- **Auto Format:** Indents matching pairs of IF and REPEAT instructions, based on the level of their insertion.
- **Undo:** Removes any changes you have made to the script.
- **Redo:** Reapplies any changes you have made to the script.
- **Cut:** Removes the selected text and copies it to the Clipboard.
- **Copy:** Copies the selected text to the Clipboard.
- **Paste:** Pastes the selected text on the Clipboard at the cursor position in the script.
- A Find: Displays the Find dialog box where you can specify the word or text string for which you are searching.

- **# Find Next:** Finds the next instance of the word or text string for which you are searching.
- ^{ab}_{ac} Find and Replace: Searches for a specific word or text string and replaces it with word or text string you specify.
- **L=Go To:** Displays the Go To dialog box where you can specify the location in the script to which you want to move your cursor. You can go to either a specific line or character in the script.
- **Findent Block:** Indents the selected text.
- **©Outdent Block:** Realigns the indented text with the left margin of the scripting box. (If the text has been indented more than once, realigns the text with the previous tab stop.)
- **Comment Block:** Causes the selected text to become inactive, meaning HotDocs won't process it when it processes the computation. Often, template developers "comment out" scripting if they need to close the scripting dialog box but HotDocs won't let them because the script is invalid. Additionally, developers often add personal explanations to the scripts they are writing, and commenting them out keeps the computation operational.
- **Uncomment Block:** Removes any commenting that has been applied to the selected script. Once the script is uncommented, HotDocs will attempt to process it.
- **Options:** Displays the HotDocs Options dialog box where you can customize the way script editing features work.
- **Whelp:** Opens the relevant page of the HotDocs Help File.

Below this is the **Script** field in which you can type in the expression. You can also use your mouse to drag variables, operators, and models from the lists below to this box, or press Ctrl+Space to have HotDocs display these lists directly in the script so you can use the keyboard to choose the syntax item you need.

On the left is the Component list \subseteq showing a list of all the components in this template. You can search it using the find field at the bottom, filter it by selecting a component type from the drop-down list at the top or create a new component by clicking on the 2 **New Component** button. In the center D is a list of useful scripting operators you can drag into the Expression box. Operators help you compare relationships between items in your script. They also allow you to perform mathematical calculations on items in your script. At the right is a list of Expression models E you can use to create a custom expression. Expressions must result in a true/false value.

For further information on how to use the operators, see Use Operators when Scripting, and for further information what the expression models do, see Full List of Expression Models.

If you click the **Show Advanced** button **F** you will see a text field at the bottom on the dialog where you can enter a comment about the IF Field.

To learn more about using IF fields follow the links below:

- Introduction: Make Parts of Templates Conditional
- Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression
- Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions

- Edit an IF Instruction or Expression
- Nest IF Instructions
- Use Conditional Instructions in a Computation Script
- Tips on Using IF Instructions

Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression

Often you have text in your template that you want to include or exclude based either on users' answers or on other conditions being met during the interview. For example, perhaps you need to include a client's shipping address. You can use a True/False variable to determine whether the shipping address is different from the billing address. If so, the *Shipping* variables will be asked and their answers will be merged into the assembled document. If not, only the *Billing* variables will be asked and merged.

Sometimes the situation in which you want to merge the text requires a little more information. In this case, you can use a True/False expression to condition the text. For example, perhaps you need to include a list of minor children in the document; however, you only want this list to appear if the client is married and if the children are under the age of 18. To accomplish this, you can create a True/False expression. Like a True/False variable, an expression must result in either *true* or *false* for the text to be merged or excluded correctly. Expressions are often used when a simple True/False variable doesn't convey the condition you need.

In some templates, you may have multiple versions of text, only one of which you want to include in the document. You can condition each section using a series of ELSE IF and ELSE expressions. The process of creating these expressions is similar to creating them for True/False variables. To understand how to do this, see Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions.

A true expression produces a true/false value. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a *true* or *false* value—it is not a True/False expression. But the expression YEARS FROM(Child's Birth Date, TODAY) <= 17 can only result in *true* or *false*. It is a True/False expression.

By default, when HotDocs inserts an IF expression in the template, it includes a return character after the instruction. During assembly, as the instruction is processed and removed from the assembled document, this return character is likewise removed. If you would prefer to keep this character from being merged entirely, or if you want to merge a return character only when working with complete paragraphs of text, you can specify an option that controls this. See Control When Returns Are Inserted After Instructions in Text Templates and Understand How Smart Returns are Inserted After Instructions.

To condition text using a True/False variable

- 1. Highlight the template text you want to make conditional—a word, a sentence, a paragraph, an instruction, and so forth.
- 2. Click the AIF Field button. The IF Field dialog box appears.
- 3. Select IF True/False Variable.
- 4. Type a name in the **True/False variable** field, or select an existing variable from the drop-down list.
- 5. Optionally, click the **dit Component** button to make changes to the variable's component properties. (See Customize a True/False Variable.)
- 6. Click **OK**. HotDocs inserts an IF instruction around the text you selected.

To add or exclude text based on a True/False expression

- 1. Select the text in the template you want to make conditional—a word, a sentence, a paragraph, an instruction, and so forth.
- 2. Click the A **IF Field** button. The **IF Field** dialog box appears.
- 3. Select **IF Expression**. The window changes to let you create a script for the expression. (See Understand the HotDocs Scripting Language.)
- 4. Enter the script, using models from the Expression models list. (For example, the following is an expression that tests three different conditions: 1) the client must be married, 2) the client must be over age 25, and 3) the client must live in Pennsylvania: Client is Married AND AGE (Client Birth Date) > 25 AND Client State= "Pennsylvania".)
- 5. Click **OK**. HotDocs inserts the expression around the part of the template you selected.

True/false expressions can contain only 512 characters. If longer, you must use a Computation variable.

When the True/False variable appears during the interview, the variable name is used as the prompt. You can enter a **Prompt** at the **True/False Variable Editor** to make it more specific. (See **Tips on Naming Your Variables** and **Create a Prompt for a Variable**.) Additionally, you can include a resource (see Add Resource Information to a Variable or Dialog.)

To help you create the expression, you can use predefined expression models from a list of models. Two expression models you will use frequently are **MULT_CHOICE = TEXT** and **MULT_CHOICE != TEXT**—both of which test what the answer is for a Multiple Choice variable.

You cannot use an IF instruction to condition a table that appears within another table. For example, if you have a table which contains another table within one of its cells, you cannot use a condition to remove the embedded table during assembly. (If you do, HotDocs only removes the text from the table and leaves the empty table in the assembled document.)

Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions

Often you have a simple paragraph (or other section) of text in your template that you want to include or exclude based either on users' answers or on other conditions being met during the interview. (See Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression.)

In other cases, however, you may have multiple versions of text, only one version of which you want to include in the document. You can condition each section using a series of ELSE IF and ELSE instructions. For example, if you have different versions of a paragraph you use for different field offices, you can use the ELSE IF and ELSE expressions to insert the correct paragraph, based on a user's answers. You can use both ELSE IF and ELSE in the same IF instruction, but ELSE must always be the last item in the instruction.

In the following example, the first part of the condition tests whether the client lives in the state of Idaho. If so, the paragraph for Idaho sales tax is merged. If the client doesn't live in Idaho but does live in Utah, the paragraph about Utah sales tax is merged. Finally, if the client lives in a state other than Idaho or Utah, the paragraph keeping them from paying sales tax is merged.

«IF Client Lives in Idaho»

Sales tax for the State of Idaho is «Sales Tax Amount»% and must be paid at the time of purchase.

«ELSE IF Client Lives in Utah» Sales tax for the State of Utah is «Sales Tax Amount»% and must be paid at the time of purchase.

«ELSE» No sales tax must be paid.

«END IF»

To create these ELSE IF sections, you can use either True/False variables or you can use expressions.

HotDocs processes IF instructions by starting at the top of the IF instruction and looking for the first condition that is true. Once it finds a true condition, it inserts the correct text and continues processing any additional instructions. But if all the conditions are false, HotDocs doesn't insert anything at all (unless there is an ELSE instruction.) When the True/False variable that controls the IF instruction is not answered, none of the instructions are processed and no text is merged in the document.

To add or exclude multiple versions of text using two or more True/False variables or expressions

- 1. Highlight the first conditional part of the template and create an IF instruction. (See Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression.)
- 2. Select the IF instruction's **END IF** keywords together with the next section of conditional text and click the **PIF Field** button. The **ELSE Field** dialog box appears.

- 3. Select **ELSE IF True/False Variable** or **ELSE IF Expression.** The dialog box changes to show the options for defining the condition.
- 4. Enter a name in the **True/False variable** field, or enter an expression in the **Expression** field.
- 5. Click **OK** to complete the instruction.
- 6. Repeat this procedure for each section of conditional text, making sure you select the preceding section's END IF instruction, along with the new section of text.

Optionally, if you have a paragraph or block of text you want to include if all other conditions are false, you can use an ELSE instruction.

- 7. Highlight the last **END IF** instruction together with the part of the template you want to be used if none of the preceding conditions in the instruction are true.
- 8. Click the 😪 IF Field button. The ELSE Field dialog box opens.
- 9. Select **ELSE** and click **OK**. The selected section of the text is added to the IF instruction. (You don't have to specify a condition for this section—it will simply be used whenever all the earlier conditions are false.)

True/false expressions can contain only 512 characters. If longer, you must use a Computation variable.

When the conditions in an ELSE IF instruction are all True/False variables, you can put them together in a single-selection group in a dialog. (See Change a Dialog's Options.)

You cannot use an IF instruction to condition a table that appears within another table. For example, if you have a table which contains another table within one of its cells, you cannot use a condition to remove the embedded table during assembly. (If you do, HotDocs only removes the text from the table and leaves the empty table in the assembled document.)

Edit an IF Instruction or Expression

You can edit an IF instruction or expression.

To change the condition for an IF instruction or expression

- 1. At the template, place the cursor in the IF instruction or expression.
- 2. Click the AIF Field button. The IF Field dialog box appears.
- 3. Edit the condition as necessary. Optionally, you can:
 - Create or select a different True/False variable or create a true/false expression. (See Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression or

Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions.)

- Click the Edit Component button and type a prompt in the Prompt field to help the user make a selection. (See Create a Prompt for a Variable.)
- Click the **Edit Component** button and then click the **Resource** tab to add a resource to the instruction or expression. (See Add Resource Information to a Variable or Dialog.)
- Click the Edit Component button and then click the Advanced tab to specify several advanced options that control how the variable used in the instruction is processed. (See Control How HotDocs Processes a Variable, Control How Answers Appear in the Assembled Document, and Specify the Width of Answer Fields in the Interview.)

In past versions of HotDocs, using END by itself instead of the complete END IF instruction was sufficient. Beginning with 11.1, HotDocs no longer processes the shortened version, and now requires the complete instruction.

Nest IF Instructions

You can nest an IF instruction inside another IF instruction. HotDocs won't process the nested instruction unless the first (or top) level of the instruction is true.

In this example, if HotDocs finds that *Client's Marital Status* was answered *Married*, HotDocs inserts the text with the spouse's name and then checks to see if *Client Has Children by This Marriage* was answered *true*. If it is, HotDocs inserts the names of the children.

This ruling affects the client, «Full Name of Client»«IF Client's Marital Status = "Married"»; and spouse, «Full Name of Spouse»«IF Client Has Children by This Marriage»; and the following children: «REPEAT Children:a, b, and c»«Full Name of Child»«END REPEAT»«END IF»«END IF».

But if HotDocs finds that *Client's Marital Status* was not answered *Married*, HotDocs skips the rest of the instruction, including the question about children in the marriage.

If you ask variables later in the template based on a nested IF instruction being true, you must make sure you use the full conditional reference before asking the variable. See **Tips on Using IF Instructions**.

You can have HotDocs identify the level of nesting in your IF instructions using both labels and field coloring. See Use Labels to Identify Instructions and Assign Colors to Fields and Instructions in Templates.

Use Conditional Instructions in a Computation Script

Just as you can insert IF instructions and expressions directly in a template, you can use a Computation variable to control conditional text. The computation can be as complicated as it needs to be, as long as it follows the same logic applied when you insert an instruction directly in the template. (See Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression and Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions for a reminder.)

The IF instruction keywords (IF EXPRESSION, ELSE IF EXPRESSION, and so forth) are all available at the **Instruction models** list in the **Computation Editor**. You can either type the instructions directly in the **Script** field, or you can drag the models to the **Script** field and replace the EXPRESSION placeholders with either True/False variables or true/false expressions. (See Use the Script Editor.)

The following is an example of a script that uses conditional logic:

```
IF State = "Idaho"
    Total Price + (Total Price * 0.05) + Shipping
ELSE IF State = "Utah"
    Total Price + (Total Price * 0.0625) + Shipping
ELSE
    Total Price + Shipping
END IF
```

In past versions of HotDocs, using END by itself instead of the complete END IF instruction was sufficient. Beginning with 11.1, HotDocs no longer processes the shortened version, and now requires the complete instruction.

Tips on Using IF Instructions

How HotDocs Processes IF Instructions and Expressions

HotDocs processes IF instructions by starting at the top of the IF instruction and looking for the first condition that is true. Once it finds a true condition, it inserts the correct text and continues processing any additional instructions. But if all the conditions are false, HotDocs doesn't insert anything at all (unless there is an ELSE instruction). When the True/False variable or expression that controls the IF instruction is not answered or satisfied, none of the instructions are processed and no text is merged in the document.

In past versions of HotDocs, using END by itself instead of the complete END IF instruction was sufficient. Beginning with 11.1, HotDocs no longer processes the shortened version, and now requires the complete instruction.

Line Spacing

The key to correct line spacing between paragraphs when one or more paragraphs is conditional is understanding the process by which HotDocs inserts returns. By default, HotDocs inserts a return after each IF instruction. Then, during assembly, it removes any returns it has inserted.

You can select an option that inserts returns after every instruction, an option that inserts returns only if a complete paragraph is selected, or an option that doesn't insert any returns at all. (See Control When Returns Are Inserted After Instructions in Text Templates.) If you opt to include returns after complete paragraphs, if there are blank lines between paragraphs, be sure to consistently include either the preceding or following blank line.

Qualifying Instructions

When automating a template, you can cause certain things to happen in your interview based on answers a user gives. You accomplish this by using IF instructions. When using IF instructions, however, you should never reference a conditional variable outside the context of its full condition.

For example, the following script only asks *Text Variable B* if *True/False Variable A* has been answered true. Then, later in the template, another dialog is asked based on whether Text *Variable B* has been answered.

```
«IF True/False Variable A»
«Text Variable B»
«END IF»
…
«IF ANSWERED( Text Variable B)»
«ASK Dialog 1»
«END IF»
```

However, this script is incorrect since a user can reassemble the document (using the same answer file) and change the answer for *True/False Variable A*. This would keep *Text Variable B* from being asked again. However, a value still exists for *Text Variable B* in the answer file, so when HotDocs reaches the IF expression later in the template, *Dialog 1* is asked.

To fully qualify this condition, create an IF expression that tests both *True/False Variable A* and *Text Variable B* before asking *Dialog 1*. For example:

«IF True/False Variable A AND ANSWERED (Text Variable B)»

«ASK Dialog 1»

«END IF»

Using REPEAT Instructions (Create a List of Answers)

Repeats and Lists Overview

This document covers the following topics:

Repeats and Lists Defined

You can include lists of answers in your documents by repeating variables and enabling your end-users to supply as many answers as necessary. You repeat variables by surrounding the variables with a REPEAT instruction. Instead of collecting and merging a single answer, as variables normally do, the repeated variable can merge an entire list of answers into the assembled document.

For example, the following shows a **REPEAT** instruction inserted into a template:

👿 🖬 🤊 -	× 🔮 🖛 Demo Personal Data - Microsoft Word 👝 💿								23					
File	lome	Insert	Page	ge Layout References Mailings Review View Add-Ins HotDocs						cs	۵ 🕜			
G HotDocs	r 🔒 Sav	ve «» Variable Field 🛃 INSERT Field 🖌 Edit Component 🖙 Clause Library						2	/ ≣					
	🖾 Sav	e and Clo	se g	🛧 IF Field 🔲 ASK Field 🔒 Component Manager 🔞 HotDocs Help				ocs Help	Ø	r 🖣 🛛				
	🔖 Tes	t Assembl	le 🦷	REPEAT	Field	«» Ot	her Fields 🔻	🚹 Create M	lodel				ן א נ	
	HotDocs				Fie	elds			Too	ols		Appeara		
													22	
CE CE LO												-	-	
SECHO	SECTION 3: EDUCATION DATA													
College	Colleges or Universities Attended:													
DEDEA	DEDEAT Educational la Connectional Alternation													
«REPEAT Educational information::< Years Attended» Attended «University or College Name», from «Years Attended»»/E Degree Earned I= "No Degree"»														
and received «.an» «Degree Earned» in «Curriculum Studied»«END IF».														
«END R	EPEAT»													
Degrees	Earned:												-	
«Employee Full Name» has earned the following degrees: «REPEAT Educational Information:a, b,							*							
and	ind c:::Graduated»«Degree Earned» («University or College Name»)«END_REPEAT».								0					
4													•	
Page: 1 of 2	Words: 2	204 🕉) 🚈 🔳 1	00% 😑 🗕			÷ .;	

This shows the list of answers the REPEAT instruction merged into the template:

👔 Employee Personal Data - HotDocs Developer 📃		3
File Edit View Navigate Tools Help		
🗐 🛛 New Answer File 🗾 🔞 📴 🚱 🗈 🕞 🛃 🛜 🖓 🐖 🐼 🕢		
Interview Document Preview Question Summary Answer Summary		
	^	
SECTION 3: EDUCATION DATA		
Colleges or Universities Attended:		
Attended University of Aberdeen from 1991-1992 and received a Master of Business Administration in Business Management. Attended University of Glasgow from 1979-1980 and received a Master of Sciences in Computer Systems. Attended University of Edinburgh from 1974-1978 and received a Bachelor of Sciences in Engineering	n 9	
Degrees Earned: Ivan J Reilly has earned the following degrees: Master of Business Administration (University of Aberdeen), Master of Sciences (University of Glasgow), and Bachelor of Sciences (University of Edinburgh).	Е	
4	- F	
🖌 Beginning 🥖 📢 📢 Previous Next 🕨 🕨 🕅	End 🕨	
	NUM	

Create and edit a REPEAT instruction

You can create a list of repeated answers by inserting a REPEAT instruction around one or more variables. For example, perhaps you want to insert a list of personal property items with their associated values. You can create a variable for each of the two data types (items and values), and then repeat them.

When you create the REPEAT instruction, you are asked to create or select a custom dialog to control the instruction. During the interview, the user can answer the dialog over and over again, until they have listed all their pertinent property items.

You also have the option to **limit the number of answers** a user can supply for a repeated variable. HotDocs merges these answers into the finished document.

Additionally, you can use the REPEAT instruction to repeat any text, instructions, or other parts of a template. Everything inside a REPEAT instruction becomes repeatable together with the variables.

Repeats in a Table

Often, you may need to repeat information within a word processor table. This is best done by repeating a single cell or an entire row.

Repeats with Computation Variables

Instead of inserting a REPEAT instruction directly in a template to generate a list of answers, you can also use a REPEAT instruction in a Computation variable to create your list of answers. Using a computation enables you to quickly insert a list in more than one location in the template.

Repeat Return Options

By default, when HotDocs inserts a REPEAT instruction in the template, it includes a return character (similar to a paragraph mark in a word processor) after the instruction. During assembly, as the instruction is processed and removed from the assembled document, this return character is likewise removed. Depending on several factors, you may prefer to keep HotDocs from adding this character, or you may want to add a return character only under certain situations. You can **specify an option** that controls this. See also, **Understand How Smart Returns are Inserted After Instructions**.

Edit Repeats

You can make changes to REPEAT instructions after they have been created, including editing the repeated dialog's contents, adding a filter, choosing an answer-sorting order, and adding punctuation to the list of answers.

Punctuate, Sort and Filter Your Lists

If you're inserting a list in sentence style, you must choose the punctuation you want for the list. If you don't choose a punctuation style, the list runs together without any punctuation or spaces—for example, *DanielNathanEmilyKate*. Once you select a punctuation style, HotDocs punctuates the list automatically—for example, *Daniel, Nathan, Emily, and Kate*.

When assigning a format, note that formats with all capital letters only capitalize the conjunction in the sentence, for example, the word *AND*. Individual answers in the list are either merged as the user types them or merged as you have formatted them using a variable-level example format.

You can also have HotDocs alphabetize your list automatically, in either ascending or descending alphanumeric order (for example, *A to Z* and *1 to 9*). You can also sort based on any repeated variable. If there's more than one repeated variable in the REPEAT instruction, you can sort on two levels—for example, first on *State*, then on *City* (or cities within the state).

You can also create a computation script that filters a list so only the entries that meet some condition are merged into the assembled document. For example, you could create a list of all the children in a family and then limit the list to include only those who are under the age of 18.

Choose a Presentation Style

You can choose how a repeated dialog appears during an interview, either by presenting the list as a series of dialogs, or as a single spreadsheet with multiple entry points.

Your options include:

- Repeated Series—causes a regular dialog to appear with an answer field for each variable. When the user answers the questions in the dialog, they can either click the Add Another button to add another repetition to the dialog, or they can click Next to move to the next dialog in the interview.
- **Spreadsheet**—causes a single dialog to appear with a column for each variable and an unlimited number of rows. The user fills in as many rows as they need. (The spreadsheet format lets the user see all the answers at one time and makes it easier to change answers. However, a dialog with several variables can make a spreadsheet too large to view easily.)

Customize REPEAT Titles

You can customize the titles used for each repetition in a repeated series of dialogs. For example, perhaps you need to ask the user for a list of beneficiaries. By default, HotDocs displays this list in the interview outline using an incremented number, followed by the dialog title:

🖃 🔚 Beneficiary Information
1: Beneficiary Information
2: Beneficiary Information
3: Beneficiary Information
- 😑 4: Beneficiary Information
5: Beneficiary Information
New: Beneficiary Information
End of Interview

This approach may work in some situations, but in others, it may be better to customize the title for each repetition so the user knows what's in that particular dialog. For example, maybe you want the topmost icon in the list to use a generic title (like *List of Beneficiaries*), but you want the name of the beneficiary to appear for each individual repetition in the interview. This way, the user can easily review what's in each dialog, like this:

🖃 🔚 List of Beneficiaries
📃 1: Jeffrey Shields
5: Ilena Shields
New: Next Beneficiary
End of Interview

To accomplish this, you must do the following:

• Assign a generic label (such as *List of Beneficiaries*) to the repeated series.

• Create a computation script that merges the user's answer to *Beneficiary Name* in the title. If you don't give an answer, the script merges other text that shows the user that the dialog is unanswered. Otherwise, HotDocs merges an unanswered variable placeholder in the title.

Additional Options

You can further customise REPEAT instructions in the following ways:

- Control How Many Rows are Visible in a Spreadsheet Dialog—By default, when displaying a spreadsheet dialog, HotDocs always shows 10 rows of the spreadsheet. If you need to show more or less than this number of rows, you can enter a specific number at the Dialog Editor. (This option doesn't limit the number of answers a user can give—it only controls the number of answers that can be viewed at any given time.)
- Limit the Number of Answers Allowed for a Repeated Dialog—At times, you may want the user to enter only a specific number of answers in a list. You can use the LIMIT instruction to limit the number of times a dialog repeats. Once the user reaches the limit, they can not enter anymore answers.
- Get the Sum Totals for Repeated Number Variables—You can use a Computation variable to find the sum total of a repeated Number or Computation variable. For example, you may need to find the grand total of a series of dollar amounts.
- Make a List Appear as one or More Columns—Sometimes it makes more sense to have your list of answers appear in columns, rather than in a punctuated sentence. If you want to make the list appear in a column, include a hard return after the text in the REPEAT instruction.
- **Count the Number of Entries in a List**—You can use a computation script to determine how many times a user answers a repeated dialog. You can also use a built-in Number variable that numbers a list of answers automatically.
- **Retrieve a Specific Answer from a List**—At times, you may need to retrieve a specific answer from a list of answers. For example, maybe you have created a list of employees, but later in the document you need to merge the name of just the second employee in the list. You can do this by using explicit indexing. To do this, you type the index number of the answer you need between brackets, just after the variable name.
- **Retrieve Information Other Than Answers from a REPEAT Instruction**—You can use a Computation variable to retrieve information (other than a list of answers) from a REPEAT instruction.

At a Glance: The REPEAT Field dialog box
REPEAT Field		? <mark>×</mark>
Field type	🔘 REPEAT Data <u>b</u> ase	
<u>D</u> ialog:	B	•
Format:	O	•
Show <u>A</u> dvanced		OK Cancel

After opening a template to edit, you can open the **REPEAT Field** dialog box by clicking **GREPEAT Field** on the HotDocs ribbon.

To choose the type of REPEAT field, you can choose either **REPEAT Dialog** or a **REPEAT Database** by selecting the pertinent option \mathbf{A} .

In the **Dialog** field \mathbf{B} you can type a name for the Dialog or Database, or select from a list of existing components using the drop-down list. You can edit your component using the **drop-down d**.

You can also select the format, allowing you to punctuate a list of answers in sentence-style . For example, your list could appear as A, B, and C—with the comma preceding the final item in the list. Click the drop-down button to see a list of formats or type your own format in the text box.

To learn more about using REPEAT fields follow the links below:

- Introduction: Include Lists in Your Documents
- Creating a REPEAT Instruction to Gather a List of Answers
- Editing a REPEAT Instruction
- Choosing a Presentation Style for the Repeated Dialog
- Customizing REPEAT Titles
- Punctuating a Sentence-Style List
- Sorting a List of Answers
- Filtering a List of Answers
- Controlling How Many Rows are Visible in a Spreadsheet Dialog
- Limiting the Number of Answers Allowed for a Repeated Dialog
- Getting Sum Totals for Repeated Number Variables
- Using a Word Processor Table to Display a List
- Making a List Appear as one or More Columns
- Counting the Number of Entries in a List

• Retrieving a Specific Answer from a List

Creating a REPEAT Instruction to Gather a List of Answers

Sometimes you may need to collect and merge a list of answers (instead of just a single answer) into a document. You can create such a list by inserting a REPEAT instruction around one or more variables.

Often, you must repeat information within a word processor table This is best done by repeating a single cell or an entire row.

To create a REPEAT instruction to gather a list of answers:

- 1. Select the template text (including any pertinent variables) you want to repeat, and click **REPEAT Field**.
- 2. Either select an existing dialog from the **Dialog** drop-down list, or type the name of a new dialog in the box.
- 3. Click the *i* Edit Component button.
- 4. Click the **Style** drop-down button and select the format for the repeated dialog.
- Optionally, in the Add button text box, type the custom label text you want to use on the Add Another button. (This button appears on a repeated series dialog. Users click it to add a new repetition to the series of answers.)
- 6. If you are creating a new dialog, add to the dialog the variables you want repeated.
- 7. Click **OK** at the **Dialog Editor**.
- 8. In the **REPEAT Field** dialog box, complete either of the following optional tasks:
 - Click the **Format** drop-down button to choose a punctuation style for a sentence-style list (for example, *A*, *B*, and *C*.)
 - Click **Show Advanced** to select sorting and filtering options for the repeated dialog.
- 9. Click **OK.** HotDocs inserts the REPEAT instruction in the template.

You can always change punctuation, sorting, or other options by editing the instruction or the contents of the repeated dialog

You can use repeats to find the total sum of a series of repeated Number variables, such as a list of dollar amounts.

In Microsoft Word, you can also insert a REPEAT instruction either by clicking the **HotDocs** drop-down list in the HotDocs toolbar and choosing **REPEAT Field**, or by right-clicking in the template and choosing **REPEAT Field** from the shortcut menu.

If you are using both DOCX and RTF template you need to be aware that the format of repeated paragraphs may appear differently in RTF documents than in DOCX documents. When converting RTF templates to DOCX, you should inspect paragraph formatting of repeated paragraphs in assembled documents to ensure that they appear as you intend.

Editing a REPEAT Instruction

You can make changes to REPEAT instructions, including editing the repeated dialog's contents, adding punctuation to the list of answers, choosing an answer-sorting order, and adding a filter.

To edit a REPEAT instruction

- 1. Position the cursor in the opening REPEAT instruction.
- 2. Click the **EXECUTE** REPEAT Field button. The **REPEAT Field** dialog box appears with the repeated dialog already selected.
- 3. Make any of the following optional changes:
 - Click the **dialog** (including adding or removing variables from the dialog and selecting presentation options for the dialog.)
 - Click the **Format** drop-down button and select a punctuation format.
 - Click **Show Advanced** to specify a sorting order and add a filter.

If you're using a complicated series of REPEAT instructions, you can match REPEAT instructions with END REPEAT instructions. You can also assign color coding to REPEAT instructions to help you view relationships between the REPEATS and other instructions in the template.

Choosing a Presentation Style for the Repeated Dialog

You can choose how a repeated dialog appears during the interview, either as a series of dialogs or as a single spreadsheet with multiple entry points.

To make a repeated dialog appear as a series

- 1. Edit the REPEAT instruction.
- 2. Click **Edit Component** to open the **Dialog Editor**.
- 3. Click the **Style** drop-down button and choose **Repeated Series**.

- 4. (Optional) Assign a custom title to each repetition in the dialog.
- (Optional) Enter your custom label text for the Add Another button in the Add button text field. (This button appears on a repeated series dialog. Users click it to add a new repetition to the series of answers.)

To make a repeated dialog appear as a spreadsheet table

- 1. Edit the REPEAT instruction.
- 2. Click **Edit Component** to open the **Dialog Editor**.
- 3. Click the **Style** drop-down button and choose **Spreadsheet**.
- 4. **(Optional)** Limit the number of visible rows in the spreadsheet by entering a number in the **Rows to display** field.
- 5. **(Optional)** Select **Hide buttons** to keep the **Edit Row**, **Insert Row**, and **Delete Row** buttons from appearing just below the spreadsheet. Users can still access these commands either by right-clicking the spreadsheet row, or by choosing the commands from the **Edit** menu.

If the repeated dialog contains a multiple-selection Multiple Choice variable, you must use the **Repeated Series** style—you cannot use the **Spreadsheet** style.

If you use the **Spreadsheet** style for a repeated dialog containing a True/False variable, the unanswered variable defaults to false.

HotDocs takes the format for repeated paragraphs in DOCX templates from the first paragraph. This may appear different to previous RTF template formatting.

You can use the **Spreadsheet on Parent** option from the Style menu to **display a child dialog** directly on the parent dialog.

You can control the width of the columns in a spreadsheet the same way you can control the width of answer fields in the interview.

Customizing REPEAT Titles

You can customize the dialog titles used for each repetition in a repeated series of dialogs.

To customize the title of the repeated dialog

- 1. Edit the repeated dialog.
- 2. At the Style drop-down list, select Repeated Series.
- 3. Type a generic title in the **Series Label** text box.

- 4. Right-click the **Title** text box and choose **Variable Field**.
- 5. Choose **Computation** as the **Variable type** and type a name in the **Name** field.
- 6. Click *i* **Edit Component**.
- 7. In the **Script** field, enter the script that tests whether the user has answered the variable you want to use in the title. The script should also merge the text you want HotDocs to use in the title. An example script would be:

VALUE(Beneficiary Name, "Next Beneficiary")

- 7. Click **OK** at both the **Computation Editor** and at the **Variable Field** dialog box. HotDocs merges the variable in the title field.
- 8. Click OK to close the Dialog Editor

Punctuating a Sentence-Style List

If you want to insert a list in sentence style, you must choose the punctuation you want to use. If you don't choose a punctuation style, the list runs together without any punctuation or spaces—for example, *DanielNathanEmilyKate*. Once you select a punctuation style, HotDocs inserts that style of punctuation in the list—for example, *Daniel, Nathan, Emily, and Kate*.

To punctuate a list

- 1. At the template, edit the REPEAT instruction you want to punctuate.
- 2. Click the Format drop-down button.
- 3. Select a punctuation style, as explained in the following table:

То	Do This
Insert a comma even when there are only two items	Choose a, and b .
Punctuate a list, but keep <i>and</i> from being used	Choose a, b .
Keep the last comma before and from being inserted	Choose a, b and c .
Use or instead of and	Choose a, b, or c .
Use semicolons instead of commas between list items	Choose a; b; and c .

Enter your own punctuation style

Type your style in the **Format** field. Base it on the other formats in the list. For example, you could use *or* or some other text instead of *and*, or you could use a semicolon instead of a comma.

You can also create a list in one or more columns.

You can punctuate the same list differently at another place in the document. Choose a punctuation style separately at each REPEAT instruction field.

Sorting a List of Answers

You can have HotDocs alphabetize your list in either ascending or descending alphanumeric order (for example, *A to Z* and *1 to 9*). You can sort based on any repeated variable. If there's more than one repeated variable in the REPEAT instruction, you can sort on two levels—for example, first on *State*, then on *City* (or cities within the state).

To sort a repeated list of answers

- 1. At the template, edit the REPEAT instruction you want to sort.
- 2. Click **Show Advanced**. The dialog box changes to show sorting and filtering options.
- 3. Click the **Sort by** drop-down button and choose the variable on which you want to sort.
- 4. Select **Ascending** or **Descending**.
- 5. Optionally, to sort on a second level, select a variable at the **Then By** field and choose either **Ascending** or **Descending**

When you close the **REPEAT Field** dialog box, you can see the names of the variables on which you are sorting added to the REPEAT instruction.

You can sort the same list differently at another place in the document. Edit that specific REPEAT instruction and specify the sort order you want, following the instructions given above.

Filtering a List of Answers

You can create a computation script that filters a list so HotDocs merges into the assembled document only those entries that meet a particular condition. For example, you can create a list of all the children in a family and then limit the list to include only those who are under the age of 18.

To filter a list

- 1. At the template, edit the REPEAT instruction.
- 2. Click Show Advanced. The dialog box changes to show sorting and filtering options.
- 3. Click the **Edit Component** button next to the **Filter by** drop-down list. The **Computation Editor** appears.
- 4. Type a name for the filter in the **Variable name** field.
- 5. In the **Script** field, type the condition that has to be true for an entry to be included in the list. (The variable in the condition must also be in the repeated dialog.) For example:

Dependent Age <= 17

- 6. Click **OK** to return to the **REPEAT Field** dialog box.
- 7. Optionally, if you've assigned a sorting option and you want the sorting to happen before the filter is applied, select **Sort first**.
- 8. Click **OK** again to return to the template.

You can make a filter as complicated as you need it to be, but it must result in either true or false. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a true or false value—it is not a filter. But the expression YEARS FROM(Child's Birth Date, TODAY) <= 17 can only result in true or false. It can correctly filter all children under the age of 18 from a list.

Controlling How Many Rows are Visible in a Spreadsheet Dialog

By default, for a spreadsheet dialog, HotDocs displays 10 rows of the spreadsheet. If you need to display more or less than this number of rows, you can enter a specific number at the Dialog Editor. (This option doesn't limit the number of answers a user can enter—it only controls the number of answers that a user can view at any given time.)

To change the number of viewable rows in a spreadsheet

- 1. Edit the REPEAT instruction.
- 2. To open the **Dialog Editor**, click the **Edit Component** button.
- 3. Click the Style drop-down button; then choose Spreadsheet.

4. In the **Rows to display** field, either type a number, or click the up or down arrows to select a number.

This feature only alters the amount of spreadsheet rows displayed in the interview and will not stop the user from entering information in further rows. The user can view the additional rows by using the scroll bar on the spreadsheet. To limit the amount of rows that the user can create see **LIMIT NUM**.

The **Rows to display** feature is not compatible with templates published for **HotDocs Server**. You can not see the effect of any alteration you make to this field in the **Server** interview.

Limiting the Number of Answers Allowed for a Repeated Dialog

At times, you may want users to enter only a specific number of answers in a list. You can use the LIMIT instruction to limit the number of times HotDocs repeats a dialog.

To limit the number of answers that a user can enter

- 1. Open the repeated dialog for editing.
- 2. Click the **Script** tab. The window changes to show scripting options.
- 3. Drag **LIMIT NUM** from the **Instruction models** list into the **Script** field.
- 4. Replace the **NUM** placeholder with the maximum number of answers you want the user to enter.

You can let the user change the limit each time HotDocs assembles the document. To do this, replace **NUM** with a Number variable. Make sure the user answers the Number variable before HotDocs processes the REPEAT instruction.

To control the number of viewable rows in a spreadsheet (but still allow users to enter as many answers as they need), enter the number in the **Rows to display** field. (Make sure you select **Spreadsheet** as the **Style**.)

Getting Sum Totals for Repeated Number Variables

You can use a computation variable to find the sum total of a repeated number or computation variable. For example, you may need to find the grand total of a series of dollar amounts.

To total a column of number variables

- 1. If the repeated variable you are totaling is a list of Number variables, create a computation that contains the expression **SUM(NUM_VAR)**, or if you are totaling a list of Computation variables, create a computation that contains the expression **SUM(COMPUTATION VAR)**.
- 2. Replace the **NUM_VAR** or **COMPUTATION_VAR** placeholder with the name of the repeated Number or Computation variable, respectively.
- 3. Insert the computation variable containing the SUM instruction where you want the total to appear—for example, at the bottom of a column of numbers.

Repeats in a Word Processor Table Overview

You can insert a list of answers into a word processor table using a **REPEAT instruction**. Then, as the user provides answers during the interview, HotDocs adds that information to the table.

There are two main approaches to inserting a REPEAT instruction into a table:

- **Regular REPEAT**—If you want to repeat the contents of a single cell in a table you select only the text within that cell, taking care to not select the cell itself. This causes the repeat to function as a regular REPEAT instruction does when not in a table.
- **Row REPEAT**—If you want to repeat an entire row of a table, you need to select an entire cell or the entire row of the table. When you insert a row REPEAT instruction in a table, HotDocs does not include an END REPEAT instruction—the end of the row signifies the end of the repeat.

When you place row REPEAT instructions in a table you should be aware that:

- Each row in the table must have the same number of columns
- The REPEAT instruction must be the first HotDocs field in the table row; no other HotDocs field can precede the REPEAT
- You can repeat only a single row of the table; if you use a row REPEAT field in a table it cannot be followed by another row REPEAT field in the same table

It is possible to use multiple REPEAT instructions of both types within a single table. To do so, ensure that the REPEAT instructions are correctly nested in dialogs, and that you place any regular REPEAT instructions completely within a single cell.

HotDocs supports the following different combinations of REPEAT instructions in a table cell:

• A regular REPEAT completely contained within a single table cell. In the case of nested regular REPEAT instructions, you must place the outermost regular REPEAT completely within a single table cell.

- A row REPEAT that is the first field in the table row.
- A row REPEAT that is the first field in a table row followed by any number of Regular REPEAT instructions, either in the same cell (after the single Row REPEAT) or in other cells of the same row. You must place each Regular REPEAT (and any Regular REPEATS nested inside them) completely within the table cell in which they appear. Additionally, you must make each outermost regular REPEAT a child dialog of the row REPEAT field at the beginning of the row. Equally you must make nested regular REPEAT instructions child dialogs of their respective containing REPEAT instructions.

Repeats can not be nested more than four levels deep.

In addition to the two main methods for inserting a REPEAT instruction in a table covered above, you can also create REPEAT instructions using a computation variable.

Using REPEAT instructions in a Table in Different Template Types

When you are using REPEAT instructions within word processor tables, it is important that you are aware of the differences which you can expect when using different types of HotDocs templates.

REPEAT instructions in DOCX templates

When you put a repeat in a table, and the interviewee does not provide any answers for the repeated element, the default behavior in RTF or WPT templates is for the assembled document to display the empty row. This differs from a DOCX template, which doesn't display the empty row (if the empty row was the only row in the table, HotDocs removes the entire table from the assembled DOCX document). If you want to use the same template across multiple formats, and you want the template to behave in a consistent, predictable way regardless of the format, you can surround the entire table in an IF block (this is easiest if you select around the table before you click **IF Field**) in the following format:

IF COUNT(RptDlg)>0/END

This makes the table conditional in the RTF and WPT formats in the same way that the DOCX table already is on its own. The image below is of an actual RTF template using this format to hide the row in the assembled document.

REPEAT Repeat»x«t1»y	
----------------------	--

Due to the differences in how HotDocs assembles a DOCX template or an RTF template, there are often some differences in table formats. For example, if you choose to enclose the whole table in a repeat instruction rather than using REPEATs within the table, HotDocs does not recommend having any text on the same line as the REPEAT instruction preceding a table. In an RTF template HotDocs merges this text into the first cell of the table after assembly. In a DOCX template that text remains visible above the table; however, the table itself has a gap between each row. These issues can be avoided by placing the REPEAT instruction on a line by itself.

REPEAT instructions in WordPerfect templates

The option for inserting a regular REPEAT in a table cell when editing a WordPerfect template is not currently available from the REPEAT field dialog. You must type the code into the template manually.

REPEAT instructions in form templates

HotDocs supports a regular REPEAT within a table cell in text templates (RTF, DOCX, and WPT) only. There is no support for a regular REPEAT in a table cell of a graphical form. Using the REPEAT instruction in a form template only produces row REPEAT instructions. regardless of where you place the REPEAT within the cell.

Using a Word Processor Table to Display a List

You can insert a list of answers into a word processor table. As the user provides answers during the interview, HotDocs adds information to the table.

To insert a regular REPEAT into a word processor table

- 1. Open HotDocs; then create a new text template.
- 2. Create a table in your word processor. (See your word processor's documentation for information on creating tables.)
- 3. Insert the variable you want to repeat in a cell within the table.

4. Once you have created your variable, position the cursor before the first variable and select only the text within the cell, not the cell itself. You can tell when you have selected just the text within the cell when the grey highlight looks like the left-hand image below:

Select the text within the cell	Select the whole cell
Items	Items
«Items»	«Items»

- 4. Click the 🗟 **REPEAT Field** button.
- 5. Then click the **Dialog** drop-down button and select the dialog you want to repeat. (To create a new dialog, click the **dialog** drop-down button.)
- 6. Select a repeat style for the dialog.
- 7. When you have finished, click **OK** at the **REPEAT Field** dialog box. HotDocs inserts the REPEAT instruction in the table, like this:

CLAIM INFORMATION FOR MULTIPLE PARTIES

Claimant Name	Items	Total Value
«Claimant Name»	«REPEAT Item list»	\$«Total Value»
	«Items»«END REPEAT»	

To insert a row REPEAT into a word processor table

- 1. Open HotDocs; then create a new text template.
- Create a table in your word processor with a single row and a column for each repeating variable. (If you want to use a row for column headings, create a two-row table and put the headings in the first row. See your word processor's documentation for information on creating tables.)
- 3. Insert the variables in the table, with one variable per column.
- 4. Once you have created your variables, position the cursor in the first table cell—before the first variable, and select the entire row.
- 5. Click the 🗟 **REPEAT Field** button. The **REPEAT Field** dialog box appears.
- 6. Click the **Dialog** drop-down button and select the dialog you want to repeat. (To create a new dialog, click the **dialog**, click the **dialog**.)
- 7. Select a repeat style for the dialog.
- 8. When you have finished, click **OK** at the **REPEAT Field** dialog box. HotDocs inserts the REPEAT instruction in the table, like this:

CLAIM INFORMATION FOR MULTIPLE PARTIES

Claimant Name	Items	Total Value
«REPEAT Claimant	«Items»	\$«Total Value»
information»«Claimant_Name»		

Due to the differences in how HotDocs assembles a DOCX template or an RTF template, there are often some differences in table formats. For example, if you choose to enclose the whole table in a repeat instruction rather than using REPEATs within the table, HotDocs does not recommend having any text on the same line as the REPEAT instruction preceding a table. In an RTF template HotDocs merges this text into the first cell of the table after assembly. In a DOCX template that text remains visible above the table; however, the table itself has a gap between each row. These issues can be avoided by placing the REPEAT instruction on a line by itself.

Making a List Appear as one or More Columns

Sometimes it makes more sense to have your list of answers appear in columns, rather than in a punctuated sentence. If you want to make the list appear in a column, include a hard return after the text in the REPEAT instruction.

For example, the following script, inserted directly in the template:

«REPEAT Children's Names» «Name of Child» «END REPEAT»

would result in a list like this:

Sarah

Jackson

Madi

Adam

If there are two or more variables in the REPEAT instruction, you can separate them with tabs so they appear in multiple columns. The following script:

```
«REPEAT Children's Information»
«Name of Child» «Age of Child»
```

«END REPEAT»

would create a list like this:

Sarah	14
Jackson	11
Madi	7
Adam	2

By default, when HotDocs inserts a REPEAT instruction in the template, it includes a return character (similar to a paragraph mark in a word processor) after the instruction. During assembly, as the instruction is processed and removed from the assembled document, this return character is likewise removed. Depending on several factors, you may prefer to keep HotDocs from adding this character, or you may want to add a return character only under certain situations. You can **specify an option** that controls this. See also, **Understand How Smart Returns are Inserted After Instructions**.

Counting the Number of Entries in a List

You can use a computation script to determine how many times a user answers a repeated dialog. You can also use a built-in Number variable that numbers a list of answers automatically.

To determine the number of entries in a list

- 1. Create a Computation variable containing the expression COUNT(DIALOG).
- 2. Replace the **DIALOG** placeholder with the name of the repeated dialog.
- 3. Optionally, insert this Computation variable wherever you want the number of entries to appear.

You don't have to insert the Computation variable in the template. Often, template developers use the **COUNT(DIALOG)** expression as a way to keep track of the number of answers, without ever inserting the actual number in the template, by referencing it in other computation variables.

To number a list automatically

- 1. At the template, position your cursor before the variable in the list you want to number.
- 2. Insert a Number variable. Instead of typing a variable name, however, click the **Variable** dropdown button and select **COUNTER**.
- 3. Click **OK.**
- 4. Add any punctuation or spaces you want to go with the number. For example:

The children are listed, as follows: «REPEAT Children's Names» «COUNTER». «Name of Child», «Age of Child» «END REPEAT»

Retrieving a Specific Answer from a List

At times, you may need to retrieve a specific answer from a list of answers. For example, maybe you have created a list of employees, but later in the document you need to merge the name of just the second employee in the list. You can do this by using explicit indexing. Type the index number of the answer you need between brackets, just after the variable name, to merge only that list item.

To use explicit indexing

- 1. Insert the variable whose value you want to retrieve into the template text. (Make sure you have repeated the variable someplace earlier in the template.)
- 2. Once you have inserted the variable, place your cursor immediately after the variable name and type—between brackets—the index number of the repetition you want to retrieve. Make sure this index is before any additional field formatting notations, for example:

The second alternate will be «Alternate Board Member[2]:LIKE THIS».

To set an explicit index for a variable in a form field, first create the variable. Then select the field and open the **Field Properties** dialog box. Enter the explicit index in the **Variable** field, immediately after the variable name (like in the example above).

Normally, you must position a repeated variable inside a REPEAT instruction, but when you a specify a particular list entry this way, the REPEAT instruction should not be included.

The **Maximum WHILE iterations** limit controls the maximum-allowed explicit index. (You set this value at the **Component File Properties** dialog box.)

Retrieving Information Other Than Answers from a REPEAT Instruction

Retrieving Information Other Than Answers from a REPEAT Instruction

You can also use a Computation variable to retrieve other information than a list of answers from a REPEAT instruction.

For example, perhaps you want to gather a list of children, and then find and return the number of children who are minors. You can use the REPEAT instruction in a computation to get the result you want. The following computation script shows an example of how to do this:

0 REPEAT Child Information IF AGE(Birthdate) < 18 RESULT + 1 END IF END REPEAT

Using INSERT Instructions

Overview: Insert the Contents of One Template into Another's Assembly Process

You can use the INSERT instruction to tell HotDocs to insert an existing template's contents into the document assembly process of another template (the "parent" template where you add the INSERT instruction).

When (during the parent template's document assembly process), HotDocs encounters the INSERT instruction, HotDocs interrupts the parent assembly to assemble a document from the "child" template. When HotDocs finishes assembling the child document, HotDocs inserts the text of the child document at the location of the INSERT instruction, and then finishes assembling the parent document so that the final output is one seamless document.

This is different from how HotDocs handles the ASSEMBLE instruction.

The INSERT instruction can be particularly useful for inserting content you use over and over again—such as signature blocks, legal headers, and other boilerplate language.

Reasons for inserting contents from one or more templates into another template's document assembly process vary with your needs. For instance, you might want to:

• Reuse your firm's boilerplate language in multiple documents

• Reference a set of related documents in one main template and enable users to choose the document that is pertinent to them

You cannot place INSERT instructions in a table.

Converting a Part of an Existing Template into a Second "Inserted" Template

In addition to using existing templates, you can select text in your current template and convert that text into a new template. At the same time, you use the INSERT instruction to add that content back into the parent template's document assembly process. You can also use the INSERT instruction to add that content into other templates document processes, as needed. For example, perhaps you realize that a section of your template could be reused in other templates. You can select this section of text, create an template from it, and then reference your new template in other related templates, using the INSERT instruction.

When you create templates by selecting existing template text, the new inserted (child) template uses the same component file as the parent template, as long as you save the child template to the same folder as the parent template.

If you save the child template to a different folder, a new, empty component file is created for the child template. If the child template uses variables from the parent template, you need to copy those variables into the new component file.

INSERT instructions inside a SPAN field will only work if the parent template and the inserted template use the same shared component file.

Use Insert to Select from Multiple Templates

You can let your users decide which child templates they want to reference if you surround your INSERT instruction with IF instructions. (See Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression and Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions.) If you have multiple instructions, you can place the True/False variables in a custom dialog and specify whether the dialog should use a multiple-select or single-select grouping.

If you want to enable your users to select several documents for assembly at the same time, be sure there's a section break or page break at the bottom of each template so each assembled document starts on a new page.

Before working with section breaks in templates, you should become very familiar with the HotDocs Header and Footer Override Scenarios.

Specify File Names For Insertion During Assembly

When inserting template contents (likewise, when inserting images) in another template, you may not always know, during template development, which files you need to reference with an INSERT instruction. When this situation arises, you can instead use a variable INSERT in the template. A variable INSERT instruction enables you to specify the template's file name during document assembly, rather than during

template development. The type of variable you insert must accept text as an answer, which means you must use a Text variable, a Multiple Choice variable, or a Computation variable that produces a text result.

For example, say there are hundreds of sub-templates to choose from. It wouldn't make sense to try to automate the template to account for each of these sub-templates—especially when template developers are creating new sub-templates on a regular basis. Using a variable INSERT instruction, however, you can script your template to specify the name of the inserted file at document assembly.

If you reference any specific templates in a variable INSERT instruction, make sure you include the template in the library when you run Template Manager. If you do not, Template Manager may not accurately report component usage information.

When publishing templates, you must manually add any files you reference in variable INSERT instructions to the published file by clicking the **Additional Templates** button on the Component File Properties General tab.

At a Glance: The INSERT Field dialog box

After opening a template to edit, you can open the **INSERT Field** dialog box by clicking on the **#INSERT** Field button on the HotDocs ribbon.

At the top of the dialog box, you can select what you would like to insert into the template from a multiple choice list. The available options change depending on what you select; below is an image of each possibility with a short description of the option:

INSERT Template

INSERT Field			? <mark>×</mark>
Field type Field	⊘ INSERT <u>I</u> mage	O INSERT Clause	INSERT Clause Library
Template <u>t</u> o insert:			
Edit Template S	now <u>A</u> dvanced		OK Cancel

In the text field you can type the file path to the template you want to insert, or you can click the **Open** button and navigate to the template.

INSERT Image

INSERT Field	? 💌
Field type Field type INSERT Template INSERT Image	ERT Clause 💮 INSERT Clause Library
Image file <u>t</u> o insert:	
Edit Template Show Advanced	OK Cancel

In the text field you can type the file path to the image file you wish to insert or you can click the **Open** button and navigate to the template.

INSERT Clause

INSERT Field			2	x
Field type	O INSERT Image	INSERT Clause	INSERT Clause Library	
Clause <u>t</u> o insert:			•	2
INSERT dause only I	IF selected		<u>OK</u> Cancel	

If you have a portion of the template text selected when you click **INSERT Clause**, you can type a name for a new clause in the text field and click the **OK** button to open the Edit Clause dialog box. If you want to insert the clause as part of an IF statement, then select **INSERT clause only IF selected** before clicking the **OK** button.

INSERT Clause Library

INSERT Field			? <mark>*</mark> *
Field type	INSERT Image	INSERT Clause	INSERT Clause Library
Clause library <u>t</u> o insert:			
s	how <u>A</u> dvanced		<u>O</u> K Cancel

In the text field you can type the file path to the clause library you want to insert or you can click the **Den** button and navigate to the template.

To learn more about using INSERT fields follow the links below:

- Introduction: Insert Templates into Templates
- Making Part of a Template into an Inserted Template
- Inserting an Existing Template
- Editing an INSERT Template Instruction
- Inserted Templates and Shared Component Files
- Overriding Headers and Footers in Inserted Word Templates
- Inserting Image Files into a Template
- Specifying File Names for Inserting Files During Assembly
- Inserted Template File Locations
- Differences Between ASSEMBLE and INSERT Instructions

Inserting an Existing Template's Contents into Another Template's Document Assembly Process

You can use the INSERT instruction to insert the assembled contents of another template into the document assembly process of the template containing the INSERT instruction.

Although you can insert templates or graphics from any location, file names for inserted templates or graphics should be unique if you plan to publish templates for use on HotDocs Server or Cloud Services.

To insert an existing template

- 1. At the template, position the cursor where you want to insert the section of text.
- 2. Click the **#INSERT Field** button. The **INSERT Field** dialog box appears.
- 3. Select INSERT Template.
- 4. Click the **Open** button and select the template.
- 5. Click **OK**. The new INSERT instruction is added to the template.

Once you have created an INSERT instruction, you can edit it by placing your cursor in the instruction and clicking the *A***INSERT Field** button again. (See Edit an INSERT Template Instruction.)

In Microsoft Word, you can also add INSERT instructions either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **INSERT Field**, or by right-clicking in the template and choosing **INSERT Field** from the shortcut menu.

Making Part of a Template into an Inserted Template

You can select some of the content of an existing template and create a separate, inserted template from that content; once you do this, you can insert that templates content into other template's document assembly process as well.

Although you can insert templates or graphics from any location, file names for inserted templates or graphics should be unique if you plan to publish templates for use on HotDocs Server or Cloud Services.

To make part of a template into a separate, inserted template

- 1. At the template, select (highlight) the text you want to include in the inserted template and click the **INSERT Field** button. The **INSERT Field** dialog box appears.
- 2. Select INSERT Template.
- 3. Type a name for the new template in the **Template to create** field.
- 4. Click **OK**. HotDocs removes the selected text from the main template, creates the inserted template, and then replaces the original text with the **INSERT Template** instruction.

Inserted Templates and Shared Component Files

When you insert one template's contents into another template's document assembly process, you may or may not want both templates to share component files.

INSERT instructions inside a SPAN field will only work if the parent template and the inserted template use the same shared component file.

Manual Sharing of Component Files

When you insert an existing templates contents, HotDocs does not automatically apply the component file of the main (parent) template to the inserted (child) template. By default, when HotDocs processes the **INSERT** instruction during assembly, it uses the inserted template's own component file. It is often the case that the child template contains variables and other components not found in the component file of the main template, and the reverse is also often true.

On the other hand, related templates often share many of the same variables. If you want an existing child template to use the same component file, you must manually make the child template share the parent template's component file. In addition, if the inserted template already contains variables, dialogs, and so forth, you must copy those existing components into the shared component file.

Automated Sharing of Component Files

When you select text within a template and make that text into a new template, as long as you save the new template in the same folder as the parent template, HotDocs automatically points the child template to the parent template's component file. This ensures the parent template and the inserted child template continue to share a common set of variables and other components.

However, if you save the inserted child template to a different folder than the parent template's, HotDocs creates a new, empty component file. If the text you used to create the new template contains variables, you must copy the pertinent components from the parent template into the new component file.

If you later realize the two templates should stop sharing component files, you can point the two templates to separate component files.

Editing an INSERT Template Instruction

Once you create an INSERT instruction in your template, you can edit the instruction at any time, including making changes to the text of the inserted template.

Although you can insert templates or graphics from any location, file names for inserted templates or graphics should be unique if you plan to publish templates for use on HotDocs Server or Cloud Services.

To edit an INSERT Template instruction

- 1. At the template, place the cursor in the INSERT field and click the **INSERT Field** button. The set **FIELD** dialog box appears.
- 2. Perform either of the following tasks:
 - Click the **Open** button to select a new template to insert.
 - Click Edit Template and make any changes necessary.
- 3. Click Save and Close.

Header and Footer Override Scenarios

In Microsoft Word, by default, every document has at least one header, even if you do not **make the** header visible by adding text or images in the header. You should make yourself thoroughly familiar with how headers work in Word before attempting to use visible headers and footers in inserted templates. Throughout this document, what is true of headers is equally true of footers.

Content Merging Basics

Every Microsoft Word document has at least one section. When you use the INSERT instruction to merge one document's contents into another's contents, at document assembly, HotDocs seamlessly merges the contents of the inserted "child" document into the section of the "parent" document where you use the INSERT instruction:

The only parent section contents involved in the merger of the two documents is the parent section where you place the INSERT instruction.

- If you have a single section in a child document, then HotDocs merges the contents of the single child section between the contents of the parent section before and after the INSERT instruction
- If you have a child document with multiple sections, HotDocs merges the content of the first child section with the content of the parent section prior to the INSERT instruction, and also merges the content of the last child section with the content of the same parent section at the point just following the INSERT instruction

In either case, HotDocs assembles the parent document until it encounters the INSERT instruction. The INSERT instruction tells HotDocs to pause assembly of the parent document while it assembles the child document and then inserts the child document's contents into the parent document before finishing assembling the part of the parent document that follows the insert instruction.

Because HotDocs merges the contents of the child document into a single section of the parent document, the merging of these two documents is not merely additive in the sense of adding the number of child sections to the number of parent sections in the final document. Rather, the number of sections in the assembled document equals the number of parent sections plus the merger of the number of child sections into the parent section where you placed the INSERT instruction. In other words, if you have a parent document with X number of sections, and a child document with Y number of sections, this merge behavior can be restated with the following formula: X + Y-1, because the first and last sections of the child document merge with the parent section hosting the INSERT instruction.

When you use the INSERT instruction, HotDocs always follows the content merging rules outlined above. The only aspect of the INSERT at issue is which headers HotDocs applies to the sections of merged contents, and which headers HotDocs overrides.

Factors Affecting Which Headers HotDocs Overrides

Every section of a Word document has a header, visible or not. When you use the INSERT instruction in a parent template, a conflict develops as to which header should apply to the section of the assembled document where the merger of the two documents occurs—the parent template's section header, or that of the inserted child template.

When you use the INSERT instruction, there are two primary factors that affect how HotDocs processes the instruction:

• The number of sections (or section breaks) in the child template

• Which INSERT options you use

Number of Sections and Section Breaks in the Child Template

Every Word document has at least a single section. A single section has no section break. The number of sections (one, or more than one section) and section breaks (none, or at least one section break) in a child template determines whether the parent section where you place the INSERT instruction:

- Remains a single section
- Becomes two or more sections

Single Section Child (No Section Break in Template)

When you insert a child template that has no section break into a parent template, the parent template section where you place the INSERT instruction remains a single section; therefore, HotDocs only needs to resolve the conflict between the single header of the parent section and the single header of the child template. How HotDocs resolves the conflict depends on the INSERT instruction options you use (see below).

Multiple Section Child (At Least One Section Break in Template)

Where the child template has at least one section break (hence at least two sections), HotDocs effectively divides the parent template section (let's call it section A) at the point of the INSERT instruction. Because of this division, HotDocs must separately handle the conflict between the header defined in the parent section and the headers defined (respectively) in the first section and the last section of the child document.

HotDocs merges the assembled child template's sections between the two parts of the parent section as follows:

Parent section A (before INSERT)+ First Child section <Child section break> Additional Child sections <Last Child section break> Last Child section + Parent section A (after INSERT)

HotDocs resolves any conflict with the parent section's header and the header of the first child section separately from any conflict with the parent section's header and the header of the last child header. These conflict resolutions depend on two factors:

- The document type of the templates involved—DOCX or RTF
- Whether you use the default INSERT instruction, or whether you use the KEEP HEADER option

Your INSERT Options

You have two basic choices when using the INSERT instruction:

• Use the default INSERT instruction—«INSERT "childname.docx"»

• Use the INSERT instruction with the keep child header option—«INSERT "childname.docx" KEEP HEADER»

You cannot use the KEEP HEADER, KEEP FOOTER, or KEEP HEADER FOOTER options with the ASSEMBLE instruction.

Inserting a child document into a parent never affects the headers of the parent sections prior to the Parent section containing the INSERT instruction; in most cases, the same is true of parent sections following the INSERT instruction. See Final Considerations below for exceptions.

HotDocs uses a set of rules to decide which of the conflicting headers to override:

• When you use the INSERT instruction alone, HotDocs keeps the parent document's header and overrides the inserted child header

While this is the standard rule, HotDocs' insert assembly behavior diverges from the standard where the child template has multiple sections. It is in these cases that the assembly behavior also depends on the document type (DOCX or RTF) of the templates involved. For more information, see Insertion Scenarios below.

- When you add the KEEP HEADER option after the INSERT instruction for a child template with a single section (no section breaks), HotDocs applies the header of the child template to the section of the parent containing the INSERT instruction
- When you add the KEEP HEADER option after the INSERT instruction for a child template with multiple sections (one or more section breaks), HotDocs applies the first header of the child template to the section of the parent containing the INSERT instruction

Because HotDocs merges the child document's contents into the parent document before finishing assembling the part of the parent document that follows the insert instruction, the last child header applies to the part of the parent section following the INSERT instruction

Insertion Scenarios

The graphics in the following sections provide visual metaphors for HotDocs assembly behavior for inserted templates, starting with the very simple and moving to more complex cases. In the following graphics, black rectangles represent sections in a template or document; different colored rectangles are used to represent headers that are defined in a particular section of a template or document.

1. Default Insert with a Single Section Parent and Single Section Child

The simplest insert case is when you use the default INSERT instruction to insert a child with a single section into a parent with a single section. The default INSERT means that HotDocs discards the child document's header and applies the parent header to the child header. Overriding the child header does not affect the contents of the child section.

Automating Text Templates



Figure 1 Default insert, Single Section Parent, Single Section Child

This scenario means you should ensure that the parent header aptly describes the child contents as well as the contents of the parent section. Naturally, you should also ensure that the child contents merge seamlessly with the contents of the parent section.

2. KEEP HEADER Insert with a Single Section Parent and Single Section Child

Using the KEEP HEADER insert option means that the child header overrides (replaces) the parent header in the finished document. The changes to the parent header do not affect the contents of the parent section; HotDocs merely inserts the child template's contents into the midst of the parent section's contents.



Figure 2 KEEP HEADER insert, Single Section Parent, Single Section Child

This scenario means you should ensure that the child header aptly describes the contents of the parent section as well as those of the child. You should also ensure that the child contents merge seamlessly with the contents of the parent section into which you insert them.

3. Default Insert with a Multiple Section Parent and a Single Section Child

When you use the default INSERT instruction to insert a child template with a single section into a parent with multiple sections, HotDocs discards the child document's header and simply inserts the child's contents into the parent section where the insert occurs.



Figure 3 Default insert, Multiple Section Parent, Single Section Child

Just as in scenario 1 above, you should ensure that the header for the parent section aptly describes the contents of the entire child as well as the parent section. You should also ensure that the child contents merge seamlessly with any contents of the parent section where you insert them.

4. KEEP HEADER Insert with a Multiple Section Parent and Single Section Child

Using the KEEP HEADER insert option means that the child header overrides the header of the parent section into which you insert the child document. In the scenario below, the content of the parent's section 2 sandwiches the inserted content from the child template, but the child's header replaces that of the parent section in the finished document.



Figure 4 KEEP HEADER insert, Multiple Section Parent, Single Section Child

This scenario means you should ensure that the child header aptly describes the contents of Parent section 2, as well as those of the child document. You should also ensure that the child contents merge seamlessly with any contents in the parent section where you insert them.

5. Default Insert with a Single Section Parent and Multiple Section Child

When you use the default INSERT instruction to insert a child template with multiple sections into a parent template with a single section, the parent section's header overrides the headers of both the first and last child section of the child document. Child headers other than the first and last remain unchanged.

When dealing with multi-section RTF templates, the header of the last section in the child document overrides the parent header for the parent content that follows the INSERT instruction.



Figure 5 Default insert, Single Section Parent, Multiple Section Child

This scenario means that for DOCX templates, you should ensure that the parent header aptly describes the contents of the first and last child sections, as well as those of the parent section. You should also ensure that the contents of the first child section merge seamlessly with the contents of the parent section *before* the INSERT instruction, and that the contents of the last child section merge seamlessly with any contents in the parent section *after* the INSERT instruction.

For RTF templates, this scenario means:

- You should ensure that the parent header aptly describes the contents of the first child section, as well as the parent section *before* the INSERT instruction
- You should ensure that the last child header aptly describes any contents in the parent section *after* the INSERT instruction
- You should also ensure that the contents of the first child section merge seamlessly with any parent section contents *before* the INSERT instruction and that the contents of the third child section merge seamlessly with any parent section contents *after* the INSERT instruction

6. KEEP HEADER Insert with a Single Section Parent and Multiple Section Child

When you use the KEEP HEADER insert instruction to insert a child template with multiple sections into a parent with a single section:

- The first child header overrides the header of the parent section and applies to that portion of the parent section that comestiem team dark ages prior to the INSERT instruction
- The last child header applies to the contents of the parent section after the INSERT instruction
- Child headers other than the first and last remain unchanged



Figure 6 KEEP HEADER insert, Single Section Parent, Multiple Section Child

This scenario means you should ensure that the Child Header 1 aptly describes any contents of the parent section *before* the INSERT instruction, and that Child Header 3 does the same for any contents in the parent section *after* the INSERT instruction. You should also ensure that the contents of the first child section merge seamlessly with the contents of the parent section before the INSERT instruction, and that the contents of the last child section merge seamlessly with any contents in the parent section after the INSERT instruction.

7. Default Insert with a Multiple Section Parent and a Multiple Section Child

The fact that the parent document contains multiple sections does not change the HotDocs conflict resolution for resolving a default insert where a child has multiple sections. The behavior is the same whether the parent has one or many sections.

When you use the default insert instruction to insert a child with multiple sections into a parent with multiple sections the parent section's header overrides the headers of both the first and last child section of the child document.

When dealing with multi-section RTF templates, the header of the last section in the child document overrides the parent header for the parent content that follows the INSERT instruction.

Child headers other than the first and last remain unchanged.

Automating Text Templates



Figure 7 Default insert, Multiple Section Parent, Multiple Section Child

This scenario means that for DOCX templates, you should ensure that the parent header where you put the INSERT instruction aptly describes the contents of the first and last child sections, as well as those of the parent section. You should also ensure that the contents of the first child section merge seamlessly with the contents of the parent section *before* the INSERT instruction, and that the contents of the last child section merge seamlessly with any contents in the parent section *after* the INSERT instruction.

For RTF templates, this scenario means:

- You should ensure that the section 2 parent header aptly describes the contents of the first child section, as well as the parent section *before* the INSERT instruction
- You should ensure that the last child header aptly describes any contents in the parent section *after* the INSERT instruction
- You should also ensure that the contents of the first child section merge seamlessly with any parent section contents *before* the INSERT instruction and that the contents of the third child section merge seamlessly with any parent section contents *after* the INSERT instruction

8. KEEP HEADER Insert with a Multiple Section Parent and Multiple Section Child

The fact that the parent document contains multiple sections does not change the HotDocs conflict resolution for resolving a KEEP HEADER insert where a child has multiple sections. The behavior is the same whether the parent has one or many sections:

- The first child header overrides the header of the parent section where the INSERT instruction occurs and applies to portion of that section prior to the INSERT instruction
- The last child header applies to the contents of the parent section after the INSERT instruction
- Child headers other than the first and last remain unchanged



Figure 8 KEEP HEADER insert, Multiple Section Parent, Multiple Section Child

This scenario means you should ensure that the Child Header 1 aptly describes any contents of the parent section where the INSERT instruction occurs *before* the INSERT instruction, and that Child Header 3 does the same for any contents in that section *after* the INSERT instruction. You should also ensure that the contents of the first child section merge seamlessly with the contents of the parent section before the INSERT instruction, and that the contents in that the contents of the last child section merge seamlessly with any contents in the parent section after the INSERT instruction.

Final Considerations

While the foregoing information contains nearly all you need to understand to properly insert one template's contents into another's document assembly process (provided you have a good grasp of how headers work in Word), you should understand that setting parent sections to inherit a preceding section's headers can complicate the interactions outlined above. Specifically, suppose you use Word to set the parent sections following Parent section A to inherit headers from section A (before inserting the child template's contents). If you then use the KEEP HEADER option when you insert the child template's contents, parent sections following Parent section A can inherit the child headers. You should, therefore, be judicious in the use of these inheritance features in your parent template.

Different First Page and Other Section Properties

Independent of each section's Header and Footer properties (which affect the content of the headers and footers) and the override scenarios discussed above, there are other section properties that determine which headers and footers appear (as well as the other page layout features of the section):

- Different first page
- Different odd and even
- Columns

- Margins
- Section break type
- Page border

HotDocs handles conflicts between these section property settings in the parent template and the corresponding property settings in the inserted child template in a very specific way:

- HotDocs copies these section properties from the parent section where you place the INSERT instruction and overwrites the corresponding properties in the first section of the child template
- If there is more than one child section, HotDocs copies the section properties listed above from the last section of the child and overwrites these properties in the part of the parent section following the INSERT instruction

Overriding Headers and Footers in Inserted Word Templates

In Microsoft Word, by default, every document has at least one header, even if you do not make the header visible by adding text or images in the header. You should make yourself thoroughly familiar with how headers work in Word before attempting to use visible headers and footers in inserted templates. Throughout this document, what is true of headers is equally true of footers.

When using headers in your inserted templates, you must first determine whether you want the header of the parent section to override the header of the inserted child, or whether you want the child header to override the parent template's header.

Before inserting one template within another, you should become very familiar with the HotDocs Header and Footer Override Scenarios.

Overriding the Header of the Inserted "Child" Template

When you use the default INSERT instruction to insert a child template's contents into a parent's contents, the header of the parent section overrides the header of the inserted child document. This is useful if you have an optional paragraph or clause that must be inserted at a specific place within the parent document.

When dealing with multi-section RTF templates, the header of the last section in the child document overrides the parent header for the parent content that follows the INSERT instruction.

To insert an existing template

1. At the template, position the cursor where you want to insert the section of text.

- 2. Click the **#INSERT Field** button. The **INSERT Field** dialog box appears.
- 3. Select INSERT Template.
- 4. Click the **Open** button and select the template.
- 5. Click **OK**. The new INSERT instruction is added to the template.

Once you have created an INSERT instruction, you can edit it by placing your cursor in the instruction and clicking the *A***INSERT Field** button again. (See Edit an INSERT Template Instruction.)

In Microsoft Word, you can also add INSERT instructions either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **INSERT Field**, or by right-clicking in the template and choosing **INSERT Field** from the shortcut menu.

Overriding the Header of the Parent Template

Using the KEEP HEADER option, you can cause an inserted child document to retain its headers. This can be useful if you want to assemble two separate documents as one while maintaining their separate identifies.

You cannot use the KEEP HEADER, KEEP FOOTER, or KEEP HEADER FOOTER options with the ASSEMBLE instruction.

To use different headers or footers in your inserted templates

- 1. In the parent template, use the Microsoft Word functionality to add visible headers for all the parent template sections.
- 2. Do the same for all sections in your child template.
- 3. Place your cursor at the place in the parent template where you want to insert the child template's content and use Word functionality to add a **Next Page** section break.
- 4. On the HotDocs tab, click **AINSERT Field**.
- 5. Next to the **Template to insert** field, click **Open**, and then browse to and select the template you want to insert; then click **OK**.
- 6. Click Show Advanced, and then select Keep header of inserted template; then click OK.
- 7. If inserting multiple templates, repeat this process for each inserted template.

Trouble-shooting

If the results you experience differ from what you expect, review both Microsoft's materials on how headers work in Word and the materials on Header and Footer Override Scenarios, giving special attention to Final Considerations.

Insert Image Files into a Template

HotDocs supports referencing graphics with an INSERT instruction in a text **template** in Microsoft Word only. If you reference a graphic in a WordPerfect template, HotDocs ignores the **INSERT** instruction.

Using a HotDocs INSERT instruction, you can insert image files into a template. HotDocs supports inserting the following image file formats:

- JPG
- PNG
- GIF

Just as with text files, the insert instruction acts as a pointer or reference to the image files. HotDocs does not insert the actual image until it assembles the document. This feature can be very useful if, when you design the template, it is not possible to determine which particular diagrams, photos, or other graphics files the user needs to specify at document assembly.

For example, say you are a legal secretary drafting a letter to be sent to a client, and any one of several attorneys can sign the letter. Rather than use Word to add or remove signatures once you assemble the document, you can instead include a series of INSERT IMAGE instructions that define the conditions for which signature gets inserted, as in the following:

«IF Attorney Name = "Cathy Bailey"»
«INSERT "cbailey sig.png" IMAGE»
«ELSE IF Attorney Name = "Jim Johnson"»
«INSERT "jjohnson sig.png" IMAGE»
«END IF»

At document assembly, HotDocs inserts the correct signature based on how you answer the *Attorney Name* question.

When you insert a graphic into a template, following the INSERT instruction and the name of the file, you need to type the IMAGE keyword. This keyword instructs HotDocs that the file that should be inserted is a graphic.

Although you can insert templates or graphics from any location, file names for inserted templates or graphics should be unique if you plan to publish templates for use on HotDocs Server or Cloud Services.

Insert an Image File Located in the Same Template

If you want to insert a graphic already in use in the template, you can do so. HotDocs removes the image from the template, copies it to a DOCX document, and inserts the **INSERT Image** instruction. An advantage to doing this (rather than inserting the graphic directly) is that you can format the graphic properties within that DOCX template, and have those settings apply to the graphic at the point of

insertion. For example, if you want to specify placement, text wrapping, borders, and other effects, you can assign them in the DOCX document, then, when you insert the image, those properties apply at the parent template. If you need to reuse the graphic in other templates, or in other places in the same template, setting these properties on the image in it's DOCX container can save you a lot time.

Inserting Image Files into a Template

Using the INSERT instruction, you can insert JPG, PNG, and GIF image files to a template's document assembly process. You can do this in two ways:

- Specifying an image external to the current template
- Specifying an image located in the same template

If during template development, you do not know which graphic you should insert, you can use a variable INSERT instruction so that the file path and name can be inserted during document assembly. For details, see Specify File Names for Inserting Files During Assembly.

Inserting a Graphic File External to the Current Template

Although you can insert templates or graphics from any location, file names for inserted templates or graphics should be unique if you plan to publish templates for use on HotDocs Server or Cloud Services.

To insert a graphic file external to the current template

- 1. In the template, position the cursor where you want to insert the image file.
- 2. Click the **#INSERT Field** button. The **INSERT Field** dialog box appears.
- 3. Select INSERT Image.
- 4. Next to the **Image file to insert** field, click the **Open** button; then browse to and select the graphic file you want to insert.
- 5. Click **OK**. HotDocs inserts the graphic at document assembly.

Inserting a Graphic File Stored in the Same Template

You can insert a graphic file that is located in the same template.

To create an INSERT instruction using an existing image in the template

- 1. In the template, click the image you want to include in a DOCX document. Field borders appear around the graphic.
- 2. Click the **#INSERT Field** button. The **INSERT Field** dialog box appears.

3. Select **INSERT image**.

- 4. Enter a name for the DOCX document in the **Image file to create** field. (To specify a folder location different from the template, click the **Decemposition** button and browse to the folder.)
- 5. Click **OK**. HotDocs removes the image from the template, copies it to an DOCX document, and inserts the **INSERT Image** instruction.

Specify File Names for Inserting Files During Assembly

If you do not know, which files your template's users need to insert during that template's document assembly process, you can use a *variable* INSERT to enable users to specify the template's file name at that time.

To specify a file for insertion during assembly

- 1. At the template, position the cursor where you want to insert the template or graphic.
- 2. Click the **#INSERT Field** button. The **INSERT Field** dialog box appears.
- 3. Select **INSERT Template** or **INSERT image**, depending on which type of file you want to include in the assembled document.
- 4. Click **Show Advanced.** The dialog expands to show advanced options.
- 5. Select Variable INSERT.
- 6. Click the **Type** drop-down button and choose which type of variable you want to use in the INSERT instruction:
 - **Computation Variable** allows you to use a computation script to specify the file path and name of the template that will be inserted. (Make sure the script produces a text result.)
 - **Multiple Choice Variable** allows you to specify the file path and name of the template that will be inserted in the **Option** column. (If you want to provide a reader-friendly name for the template, enter the name in the **Prompt** column.)
 - **Text Variable** allows you to specify the file path and name of the template using a simple text answer field.
- 7. Click the **Variable** drop-down button and choose the variable you want to use. (To create a new variable or edit the existing variable, click the **Edit Component** button.)
- 8. Click **OK.** The new INSERT instruction is added to the template.

If you reference any specific templates in a variable INSERT instruction, make sure you include the template in the library when you run Template Manager. If you do not, Template Manager may not accurately report component usage information.
When publishing templates, you must manually add any files you reference in variable INSERT instructions to the published file at the Additional Files page of the Publishing Wizard.

Inserted Template File Locations

You can point to any template or graphic you want to INSERT or ASSEMBLE at any location, as long as you specify the correct folder path information. For example, you can insert templates using the following folder structure:

Location of Template	Example
Same folder as parent template	«INSERT "Template.docx"»
	«ASSEMBLE "Template.docx"»
Subfolder of parent template	«INSERT "Subfolder\Template.docx"»
	«ASSEMBLE "Subfolder\Template.docx"»
Full file path	«INSERT "C:\My Documents\Template.docx"»
	«ASSEMBLE "C:\My Documents\Template.docx"»
Reference path	«INSERT "^referencePath\Template.docx"»
	«ASSEMBLE "^referencePath\Template.docx"»
Default Templates folder	«INSERT "\Template.docx"»
	«ASSEMBLE "\Template.docx"»

When publishing templates for HotDocs Server or Cloud Services, it is important that you give all inserted templates unique file names because all of published templates end up in the same file folder.

HotDocs does not alter the file path of an inserted template; therefore absolute paths are not recommended. **Reference paths** allow for redirection, which provides more flexibility.

Starting a New Assembly From a Template

You can use the ASSEMBLE instruction to add templates to the Assembly Queue. You can use the ASSEMBLE instruction in the following template types:

- Word templates
- WordPerfect templates
- Form templates

Using the ASSEMBLE instruction in a Word template

You can use the ASSEMBLE instruction in a Word template to add another template to the assembly queue.

To insert an ASSEMBLE instruction to a Word template

- 1. At the template, position the cursor in the template where you want the ASSEMBLE instruction.
- 2. In the HotDocs toolbar, click the **HotDocs** drop-down menu; then from the list of options, choose **Other Field**.
- 3. In the Other Field dialog box, click the Field type drop-down list; then choose ASSEMBLE.
- 4. Next to the **Template to assemble** field, click the **Browse** button, locate the template you want to assemble, then click **OK**. HotDocs inserts the ASSEMBLE instruction for that specific template.

Using the ASSEMBLE instruction in a WordPerfect template

You can use the ASSEMBLE instruction in a WordPerfect template to add another template to the assembly queue.

To insert an ASSEMBLE instruction in a WordPerfect template

- 1. Position the cursor in the template where you want the ASSEMBLE instruction.
- 2. Click the **#INSERT Field** button. The **INSERT Field** dialog box appears.
- 3. Select INSERT Template.
- 4. Click the **Open** button to select the template file you want to have assembled.
- 5. Click **OK**. HotDocs inserts the instruction for that specific template. For example: «INSERT "subpoena.wpd"».
- 6. Delete the text **INSERT** from the instruction and type the keyword *ASSEMBLE*. For example: «ASSEMBLE "subpoena.wpd"».

If the template you are adding to the assembly queue is a form template, HotDocs will not let you browse for the file from the **INSERT Field** dialog box. Instead, at the template, you must manually enter the file path and the name of the form template file.

Using the ASSEMBLE instruction in a form template

You can use the ASSEMBLE instruction in a form template to add another template to the assembly queue.

To insert an ASSEMBLE instruction in a form template

- 1. At the template, create a field and attach a Computation variable to it.
- 2. At the **Computation Editor**, type the **ASSEMBLE** instruction, including the path and file name (if necessary) of the template you want to assemble. For example: ASSEMBLE "subpoena.hft".

You can add command-line options to an ASSEMBLE instruction (for example, ASSEMBLE "subpoena.docx /pr"). If the command-line option includes a file path and name, enclose the path and name in double quotation marks (for example, ASSEMBLE "subpoena.docx /sa /af=""L Chang"""). (Four command-line options were designed specifically for use with ASSEMBLE instructions. They are: Suggest Save, Suggest Save New, Save Answers, and Save Answers Prompt. They control the saving of answers after each ASSEMBLE instruction is processed.)

If your template uses a custom interview you must place ASSEMBLE instructions in both the body of the template and in the interview script. For more information, see ASSEMBLE "FILENAME".

At a Glance: The Other Field Dialog Box

Other Field		? 🔀
Eield type:		
ASSEMBLE	A	•
Template to assemble:	0	
	0	
	ОК	Cancel
	- On	

After opening a template for editing, you can open the **Other Field** dialog box by clicking on the **Second Provide Second Provide Provide Second Provide P**

From the Field type drop-down list A you can select a field from the following list:

- ASSEMBLE—enables you to type the name of, or navigate to, the Template to assemble in the text field labeled
- PLAY—enables you to type the name of the macro you want to play
- LANGUAGE—enables you to select the language of for this section of the template and type the terms you want to use for the thousands separator and the decimal separator
- DEBUG—causes no more options to appear
- SPAN

To learn more about the features available through the Other field dialog box follow the links below:

- Starting a New Assembly From a Template
- Insert a PLAY Instruction in a Template
- Create a Foreign Language Template
- Insert Debugging Instructions in Templates and Scripts
- Allow Users to Edit the Text of an Assembled Document

Differences Between INSERT and ASSEMBLE Instructions

The way HotDocs processes an INSERT instruction is different from the way HotDocs processes an ASSEMBLE instruction:

How HotDocs Processes an INSERT Instruction

The INSERT instruction interrupts assembly of the parent template's document (if you place the instruction before the end of the template) to assemble the inserted template. HotDocs inserts the assembled text at the location of the INSERT instruction, and then finishes assembling the parent template's document so that the final output is one seamless document.

How HotDocs Processes an ASSEMBLE Instruction

Unlike the INSERT instruction, when you start a new assembly using the ASSEMBLE instruction, HotDocs adds the template to the Assembly Queue and then waits until the main document finishes assembling before assembling the template the ASSEMBLE instruction references. Each time you send the assembled document to the word processor or to HotDocs Filler, HotDocs creates a separate file, instead of appending assembled documents to the end of the previous document. ASSEMBLE instructions can be used in both text and form templates, and they can add both text and form templates to the assembly queue, regardless of the type of template where the instruction is used.

You can also use an ASSEMBLE instruction to assemble multiple documents from within one template. For example, say providing a particular while assembling a document means the user must then assemble another document. Rather than make the user assemble the second document manually, you can have

HotDocs assemble the second document automatically. You do this by using an ASSEMBLE instruction in the template. Once the first assembly finishes, HotDocs starts the second assembly.

To test an ASSEMBLE instruction, you must exit the template completely and test it from the library.

Using SPAN Instructions (Marking Editable Text in a Template)

Introduction: Use SPAN Instructions

SPAN fields are supported in Microsoft Word only.

Frequently, users need to edit document text once a document has been assembled. To allow this, you must mark sections of template text using SPAN fields. Inserting SPAN fields in a template allows users to edit the text of the assembled document while viewing the **Document Preview** tab of the assembly window. Changes made to the text can be saved in an answer file, which allows users to later reassemble the document and still have access to the changes they made.

SPAN instructions are inserted using Span components. Also, like IF and REPEAT instructions, SPAN instructions must include an END instruction.

Insert instructions should not be used inside SPAN fields.

At a Glance: The Edit SPAN Field dialog box

🚘 New Span Component - Span Component Editor - Demo Employmen 🍞 🕰	
Properties	Notes
Componen	nt name:
Title:	
	в
	OK Cancel Sa <u>v</u> e

After opening a template to edit, you can open the **Edit SPAN Field** dialog box by clicking on the $\overset{\text{e}_{*}}{$ **Other Fields** button and selecting **SPAN** from the drop-down menu, then clicking the **description** distribution.

In the first text field A you can edit the component name or your SPAN Field and in the second text field b you can enter an optional title for the field.

In the **Notes** tab you can make development notes about this Span component. See At a Glance: The Notes tab (Component/Dialog Editor) for more information)

To learn more about the SPAN field follow the links below:

- Introduction: Use SPAN Instructions
- Allow Users to Edit the Text of an Assembled Document
- Allow Users to Enter New Text in the Assembled Document

Allow Users to Edit the Text of an Assembled Document

SPAN fields are supported in Microsoft Word only.

Sometimes users need to edit the text of a document while viewing the **Document Preview** tab of the assembly window. To allow this, you can mark sections of the template using a SPAN instruction. Text between the beginning and end instruction can be highlighted and then selected and edited by the user. Any changes a user makes to the text can be saved to an answer file so that if the user ever reassembles the document, he or she can reuse the answer file and have the changes reapplied to the document.

Like IF and REPEAT instructions, SPAN instructions include an opening instruction («SPAN») and a closing instruction («END SPAN»).

To allow users to edit document text and save it in the answer file

1. At the template, select the text you want the user to be able to edit.

When selecting template text, do not include IF instructions or REPEAT instructions between the SPAN instructions, as these instructions will no longer be processed correctly once the user edits the document text.

- 2. Click the **≦**SPAN Field button on the drop-down menu below the **□**HotDocs button. The SPAN Field dialog box appears.
- 3. Enter a component name in the **Span** field.

- 4. Optionally, click the **Notes** tab to enter notes about the span component, such as an explanation about why the span field was created or how it should function in the interview. (See Add Notes to Components for details.)
- 5. Optionally, click the **dit Component** button and enter a title in the **Title** field. The title will be used to identify the editable text in the **Document Text Editor**.

When naming SPAN components, use unique component names. If two different (but related) templates use the same SPAN component name, when users edit the document text in the first document, save the answer file, and then use that answer file to assemble the second document, those changes from the first document may overwrite text in the second.

To let users enter a new paragraph of text in the document, see Allow Users to Enter New Text in the Assembled Document.

INSERT instructions inside a SPAN field will only work if the parent template and the inserted template use the same shared component file.

You cannot insert SPAN instructions in template headers and footers.

When working with SPAN instructions, you can use buttons on the HotDocs Navigation toolbar to label instructions and match starting and ending instructions. (See Use Labels to Identify Instructions and Match Opening Instructions with Closing Instructions.) You can also assign color to SPAN instructions. (See Assign Colors to Fields and Instructions in Templates.)

You can also insert SPAN instructions either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **SPAN Field**, or by right-clicking in the template and choosing **SPAN Field** from the shortcut menu.

Allow Users to Enter New Text in the Assembled Document

SPAN fields are supported in Microsoft Word only.

Sometimes you may want to allow users to enter new text in an assembled document, rather than just modify existing document text. For example, perhaps users need to insert a new paragraph, based on answers they entered during the interview. You can insert a SPAN field in the template that allows users to do this.

When users view the assembled document, they can position their cursor either at the beginning of a paragraph or between paragraphs and click the **Edit Document Text** button. HotDocs will then display a list of editable paragraphs (if any) near the cursor position. If the cursor is in a paragraph and there are no additional Span fields near the cursor position, HotDocs will simply display the **Document Text Editor**

where users can enter their text. To help users better identify which section of text they want to edit, you should enter a descriptive title for the Span component.

To insert an empty SPAN field in a template

- 1. At the template, place your cursor where you want the user to enter text.
- 2. Click the **≝**SPAN Field button on the drop-down menu below the **⊡HotDocs** button. The SPAN Field dialog box appears.
- 3. Enter a component name in the **Span** field.
- 4. Click the **dit Component** button and enter a title in the **Title** field. The title will be used to identify the editable text in the **Document Text Editor**.
- 5. Optionally, click the **Notes** tab to enter notes about the span component, such as an explanation about why the span field was created or how it should function in the interview. (See Add Notes to Components for details.)

When naming SPAN components, use unique component names. If two different (but related) templates use the same SPAN component name, when users edit the document text in the first document, save the answer file, and then use that answer file to assemble the second document, those changes from the first document may overwrite text in the second.

To allow users to edit existing paragraph text, see Allow Users to Edit the Text of an Assembled Document.

INSERT instructions inside a SPAN field will only work if the parent template and the inserted template use the same shared component file.

You cannot insert SPAN instructions in template headers and footers.

When working with SPAN instructions, you can use buttons on the HotDocs Navigation toolbar to label instructions and match starting and ending instructions. (See Use Labels to Identify Instructions and Match Opening Instructions with Closing Instructions.) You can also assign color for SPAN instructions. (See Assign Colors to Fields and Instructions in Templates.)

You can also insert SPAN instructions either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **SPAN Field**, or by right-clicking in the template and choosing **SPAN Field** from the shortcut menu.

Using Clauses and Clause Libraries

Introduction: Use Clauses and Clause Libraries

A clause is a HotDocs component that contains text. When you create clauses, you are, in essence, creating smaller templates that can be added to one main document. Clauses can contain text as well as HotDocs components (such as variables, instructions, and so forth).

You can insert a clause component directly into a template using an INSERT or an INSERT IF instruction, or you can add clauses to a clause library so a user can select them for insertion into an assembled document.

Clauses

When you insert a clause into a template, you are actually working with three different parts of the clause—the clause *field*, the clause *component*, and the clause *text*:

- **Clause field:** This is the merge field that inserts the clause text into the assembled document. You can control whether the clause is inserted automatically, or whether the clause is inserted based on certain conditions being met. When you create a clause at the clause library, there is no clause field associated with the clause.
- **Clause component:** This is the clause itself. It includes component properties, such as a prompt and resource information.
- **Clause text:** This is the text you want inserted into the assembled document. The clause text can also contain HotDocs components, such as variables, dialogs, and instructions, which are all be processed during assembly.

Clause Libraries

Like a template library, a clause library is a collection of clauses. Grouping clauses in a library allows users to select, organize, and insert any number of clauses into the document. HotDocs then proceeds to prompt users for any variables contained in the clauses.

You can create clauses and clause libraries at two places: a HotDocs library and a template. Each location causes the clause library to function a little differently:

- A clause library assembled from a HotDocs library allows users to select clauses to insert into any text document. They can select the clause library first at the template library and assemble a document using clauses from it, or they can assemble a document and then add clauses from the library to the document after assembly.
- A clause library assembled in a template allows users to select which clauses they want to insert at that specific point in the assembled document.

For additional information on the underlying functionality of clauses, see Understand How Clauses Work.

At a Glance: The Clause Library

📴 Clause Library	
💯 🐮 New 🔮 Edit 🛛 🚽	< 🗈 🖹 📥 Add 🖄 Remove 👷 Sort 🔗 Properties 🎒 🌲 🔞 📩
····· Employment Agreeme	Properties Preview Preview Tab Tool Bar
Clause Library Contents Pane	Properties Tab Employment Agreement
<	

You can open a clause library to edit from a HotDocs Library.

The main part of the clause library window is split into two parts. On the left side of the window is the Clause Library Contents Pane where you can see all the clauses in this library. You can search the library using the **Find** check box and field at the bottom of this pane. You can type the text directly in the box, or you can click the drop-down button and select text from a previous search. On the right side of the window you can use the Properties Tab to view clause properties or the Preview Tab to see a preview of clause itself.

At the top of the window is the Toolbar. From the Toolbar you can access the following buttons:

- **Arrange Windows:** Arranges the clause library window with other open HotDocs windows so you can view the windows simultaneously. Adjust the clause library to the height you want before clicking the button.
- New Clause: Allows you to create a new clause and add it to the clause library. You can copy existing document or template text into the clause, or you can type the clause text "as you go." If you copy existing template text, make sure you copy any variables used in the template into the clause library's component file.

- **Edit Clause:** Displays the Clause Editor where you can edit the name, prompt, or text of the selected clause.
- **Cut a Clause:** Removes the selected library item or items and saves those items to the Windows Clipboard. (If you select a folder, HotDocs will remove the folder with all of its contents.) You can then paste the items into another part of the current library or into a different library.
- Copy a Clause: Copies the selected library item or items and saves them to the Windows Clipboard. (If you select a folder, HotDocs will copy the folder with all its contents.) You can then paste them into another part of the current library or into a different library.
- **Paste a Clause:** Pastes any library items that have been saved to the Windows Clipboard into the current library. The item or items will be pasted below the currently selected item in the library.
- ***** Add Clause to Library: Adds an existing clause to the clause library. You can also use this button to add a folder to the clause library.
- **EXAMPLE AND ADD** Remove Clause from Library: Removes the selected clause from the clause library.
- **Sort List:** Displays a dialog box where you can sort the contents of the selected folder in ascending (A to Z) or descending (Z to A) order.
- **View Clause Properties:** Displays the Item Properties dialog box where you can change the title and description of folders and clauses in the current clause library.
- **G** Print: Prints a list of clause titles, clause file names, and the clause library in which the clause is located.
- **Component Manager:** Opens Component Manager so you can work with the clause components or other components (such as variables) used in the clauses.
- **W** Help: Opens the relevant page of the HotDocs Help File.

To learn more about using the clause library follow the links below:

- Introduction: Use Clauses and Clause Libraries
- Understand How Clauses Work
- Open and Close a Clause Library
- Create a Clause Library at the HotDocs Library
- Add a Folder to a Clause Library
- Create Clauses at the Clause Library
- Add Existing Clauses to a Clause Library
- Customize a Clause Library
- Create Clauses at the Template and Add Them to the Library
- Create and Insert Clauses Directly in Template
- Insert a Clause Library into a Template
- Edit an Existing Clause
- Group Clause Components in One Custom Dialog

Understand How Clauses Work

When working with clauses and clause libraries, it is useful to understand the underlying functionality, including the files that control how clauses and clause libraries work.

When you create a clause library and add clauses to it, HotDocs creates and uses the following files:

- **Clause template file (.DOCX, .DOT, .RTF, .WPT):** This file contains the text of the clause component. When you create a clause template file, HotDocs generates an eight-character file name based on the clause component name, then appends the appropriate file name extension, and, finally, adds the clause to an archive file (see below). If the clause name is less than eight characters, HotDocs adds underscores to make the template name the right length.
- **Clause component file (.CMP):** When you create a clause, HotDocs creates a clause component, which is added to a component file. If you create a clause at a template, the clause component go in the template's component file. But if you create a clause at a clause library, HotDocs creates a new component file that is associated with the clause library and stores the clause component in it.

In addition to storing the clause components, the clause component file also stores other components used in the clause text, such as variables, dialogs, and scripts.

- **Clause library file (.HDL):** This library file contains an organized set of shortcuts that correspond to a set of clauses stored in the HotDocs clause archive file (see below). The clause library file gets its name based on where it is created—either at the template or at a template library. For example, if you create a clause library while editing *contract.rtf*, the clause library file name becomes *contract.hdl*. However, when you create a clause library at a regular template library, you specify the file name.
- **Clause Archive file (.HCL):** This file contains a set of related clauses in compressed, or zipped, format. When you edit or assemble a document using clauses, this file is unzipped and each clause template file is extracted as needed to a temporary folder. When the clauses and clause library are no longer in use, the clause template files are zipped and saved to the clause archive file, and all temporary files are deleted.

These files work simultaneously to organize and store clauses. The clause library file, the clause template file, and the clause component file are all located in the word processor's template folder, along with the other HotDocs templates.

If you are uncertain about a clause's shortened name or file format, you can view the name in the **Clause identifier** field at the **Clause Editor**.

Open and Close a Clause Library

You can open and close a clause library either at the template library or at the template.

To open a clause library from the template library

• At the HotDocs library, select the clause library and click **Edit.** The **Clause Library** window appears.

To open a clause library from the template

• At the text template, click the Clause Library button. The Clause Library window appears.

To close a clause library

• Click the **X** in the upper-right corner of the clause library window.

To access the clause library toolbar using the keyboard, press F10.

Word users can also open a clause library by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Clause Library**.

Creating a Clause Library at the HotDocs Library

At a template library, you can create a clause library that contains any number of clauses. You can assemble these clauses and add them to any word processor document. For example, users can select a clause library at the template library, provide any required information, and then either send the assembled clauses to a new document or paste the assembled document into an open document (perhaps another related document you have just finished assembling).

Storing sections of template text in clauses and then adding those clauses to a clause library makes it easy for the user to choose which text they will use in assembling a document. A clause library also provides a way for users to organize their clauses and view the properties of each individual clause.

Users can also preview clause text from the **Preview** tab of the clause library.

To create a clause library at a HotDocs template library

1. At the template library, select the folder in which you want the clause library to appear.

2. Click **New Template**.

- 3. Click the **Type** drop-down button and select a clause library option for the word processor you are using. Your options include **Word DOCX Clause Library**, **Word RTF Clause Library**, **Word DOT Clause Library**, and **WordPerfect Clause Library**.
- 4. In the File name field type a file name for the new template. If you decide to store the template somewhere other than the default template folder (as seen in the Target folder field), you can also **Browse** to the new destination. HotDocs adds the correct extension to the file name based on the type of text template you selected. HotDocs displays the full path of the currently selected template location in the Target folder field.
- 5. Type a title for the clause library in the **Title** field (or accept the suggestion HotDocs makes).
- 6. Optionally, type a description in the **Description** field. The description appears when the user views the clause library's properties at the HotDocs template library.
- 7. To add an existing component file to the new template, you can select a component from the **Shared component file** drop-down list. If you do so, the component file you choose is now shared between the new template and any other templates using that component file.
- 8. Click **OK**. HotDocs creates and opens the clause library.

When the clause library opens, notice that it does not contain any clauses yet. You must create the clauses that appear in the clause library.

When you create a clause library in a HotDocs template library, HotDocs doesn't create a template file. However, HotDocs creates a component file (using the same file name as the clause library) to store information for components used in the clause library.

You can use a clause library to assemble any number of documents and you can **insert a clause** library into any template if you want to associate the clauses with a specific document.

Add a Folder to a Clause Library

You can add folders to your clause library so you can organize your clauses into groups.

To add a folder to a clause library

- 1. At the template, or at the template library, open the clause library for editing. (See Open and Close a Clause Library.)
- 2. With the clause library displayed, select the folder in which you want to create a new folder.
- 3. Click **¹Add**. The **Add Clause** dialog box appears.
- 4. Click the **Type** drop-down button and select **Folder**. The dialog box changes to show folder properties.

- 5. Type a name for the folder in the **Title** field.
- 6. Optionally, type a description for the folder in the **Description** field. The description appears in the **Properties** tab of the clause library window when the folder is selected.

Once you have created folders, you can arrange the folders (as well as the contents of the folders) by selecting the library item and dragging it to a new location in the list. You can also arrange the contents of each folder by clicking $\frac{1}{2}$ **Sort**. (See Customize a Clause Library.)

At a Glance: The Edit Clause dialog box

= New CI	Suse - Clause Editor - Testi.cmp	
Properties	Resource Used In Notes	
Clause nar	ne:	
	A	
Tit <u>l</u> e:		
	0	
Prompt:		
	0	*
	G	+
Vec/No	on same line	
respire	ou same me	
-		Canard Canard

After opening a clause library, you can open the **Edit Clause** dialog box by selecting the clause you clicking the \mathbf{I} **Edit button** or clicking on the **New** button on the Toolbar.

In the first text field \boxed{A} you can enter a name for the clause. In the second text field \boxed{B} you can enter an optional title for the clause and in the third text field \boxed{C} you can enter an optional prompt to accompany the clause selection in the interview.

Below these text fields there is a check box D on the left where you can choose to have the yes and no options appear on the same line in the interview.

If you are using a RTF template then you will see a drop down list at the bottom right where you can choose the type of clause you wish to create. You can select either a **MS Word RTF clause** or an **MS Word DOT clause**. Clauses created for a DOCX template will automatically be created as DOCX clauses.

DOCX clauses must be used in DOCX templates and RTF or DOT clauses must be used in RFT templates.

To view more options for setting up this new clause you can go to the Resource, Used In and Notes tabs.

To learn more about editing clauses follow the links below:

- Create Clauses at the Clause Library
- Edit an Existing Clause

Create Clauses at the Clause Library

Once you've created a clause library at a template library (see Create a Clause Library at the HotDocs Library), you can add clause text from any word processor document to it. To do this, either copy text from an existing word processor document or template and paste it into the clause component, or write the clause text as you go.

To create a clause at the clause library

- 1. Open for editing the clause library which will contain your clause components. (See Open and Close a Clause Library.)
- 2. Click New Clause. The Clause Editor appears.
- 3. Type a name for the clause in the Clause name field. (50-character limit.)
- 4. Optionally, enter a title for the clause in the **Title** field. The title will be used as the default title for the item once it is added to the clause library.
- 5. Optionally, click the **Resource** tab to provide users with helpful information that can assist them in providing the correct answer. (See Add Resource Information to a Variable or Dialog.)
- 6. Optionally, click the **Notes** tab to enter notes about the clause, such as an explanation about why the clause was created or how it should function in the interview. (See Add Notes to Components for details.)
- 7. Click **OK**. HotDocs opens a new, empty text template.
- 8. Either copy sections of existing template text and paste it into the template (make sure you copy any existing components into the associated component file), or type the text directly in the template. You can automate the clause text by adding variables, instructions, and any other HotDocs functionality.

- 9. When you are finished, click the Save and Close Template button at the HotDocs toolbar. The clause is added to the clause library.
- 10. Repeat this process for all the clauses you need to create.

In Word, the word processor may insert an extra hard return at the end of a new clause. To fix this, select the clause at the clause library and click **d** Edit. The Clause Editor appears, where you can click Edit Clause Text. When your text appears, delete the extra hard return. (Click the Show / Hide ¶ button in the Word toolbar to view paragraph marks.)

As you create clauses, if you paste template text that already contains variables or other HotDocs components, you must copy those components into the clause library's component file in order for the clauses to assemble correctly. The easiest way to do this is to click the **Component Manager** button in the clause library toolbar and copy those components in. (See Copy Components From One File to Another.)

At a Glance: The Add Clause dialog box

Add Clause	? ×
Type: Clause	-
Select clause:	
В	-
<u>Title:</u>	
Description	
D	
	<u>C</u> ancel

After opening a clause library, you can open the **Add Clause** dialog box by clicking the th **Add** button on the Toolbar.

In the drop-down list \triangle at the top of the dialog you can choose whether you would like to add an existing **Clause** or create a **Folder**. If you chose to add a **Clause** then you can select the clause, from the current clause library, you would like to add from the next drop-down list \triangle , but if you selected **Folder** then this list \triangle will disappear and you can go straight onto enter the details in the text fields below.

In the first text field \bigcirc you can enter a title for your clause or folder and in the second text field \bigcirc you can add an optional description that appears at the **Properties** tab of the clause library window when the item is selected.

To learn more about adding clauses to a clause library follow the link below:

• Add Existing Clauses to a Clause Library

Add Existing Clauses to a Clause Library

When you create clauses at the template, you can either specify the clause text and have HotDocs immediately add the clause to the clause library (see Create Clauses at the Template and Add Them to the Library), or you can manually add the clauses at a later time.

If you are adding clauses to a library that resides in a template library, you can only add clauses that have already been created for that specific clause library. (For example, maybe you have deleted a clause from the library list and you want to re-add it.)

To add an existing clause to the clause library

- 1. Open the clause library for editing. (See Open and Close a Clause Library.) The **Clause Library** dialog box appears.
- 2. Click *** Add**. The **Add Clause** dialog box appears.
- 3. Click the **Select clause** drop-down button and select the clause you want to add.
- 4. Enter a new title for the clause in the **Title** field, or accept the suggestion HotDocs makes. (If you entered one, HotDocs uses the title you suggested when you created the clause component. If you later decide you want to change the title, select the clause in the clause library and click **Properties**. Make your change at the **Item Properties** dialog box.)
- 5. Optionally, enter a description in the **Description** field. The description, along with the title, appears in the **Properties** pane of the clause library window.
- 6. Click **OK**. The clause is added to the clause list in the library.

Customize a Clause Library

It is often helpful to organize your clauses using a clause library. At the clause library, you can create, add, and delete clauses from a clause library, as well as sort clauses in alphanumeric order. You can also create additional folders in the library to group the clauses, and change the titles and descriptions of clauses and

folders. Finally, you can view either the properties of a specific clause, or you can view the text of the clause itself.

To customize a clause library

- 1. Open a clause library for editing. (See Open and Close a Clause Library.)
- 2. Perform any of the following tasks:

То	Do This
Move clauses within the clause library	Select a clause and drag it to a different position. As you are dragging, HotDocs displays a horizontal bar, indicating where the clause will be placed when you release the mouse button.
Arrange the clause library with other open windows	Resize the clause library window to the desired height and click the Arrange button. The clause library window appears below the window from which it was opened—either the word processor or the template library.
Create a new clause component	Click New . HotDocs opens the Clause Editor where you can assign properties to a clause component. Once you assign a name and click OK , a text template opens where you can enter your clause text. (See Create Clauses at the Clause Library.)
Add an existing clause to the clause library	Click th Add . HotDocs opens the Add Clause dialog box where you can select an existing clause and add it. (See Add Existing Clauses to a Clause Library.)
Add a folder to the clause library	Click th Add . When the Add Clause dialog box appears, select Folder from the Type drop- down list. (See Add a Folder to a Clause Library.)
Remove a clause or a folder from the clause library	Select the clause or folder you want to remove and click $\overleftarrow{\mathbf{a}}$ Remove Item .
Make changes to a clause in the clause list	Select the clause and click Edit . HotDocs opens the Clause Editor where you can change the clause component properties. (To edit the text of the clause, click Edit Clause Text at this dialog box.) (See Edit an Existing Clause.) Changing the name of a clause in the Edit Clause dialog will not change the page of the

	clause in the library. To do this you must also change the name in Properties .
Rearrange clauses in the clause list in alphabetical order	Select the folder whose contents you want to arrange and click the $\frac{1}{2}$ Sort button. HotDocs displays the Folder Sort Options dialog box where you can choose either Ascending or Descending order.
Change the clause or folder title, or add a description	Select a clause or folder and click Properties . The Item Properties dialog box appears where you can type a new title in the Title field, or type a description in the Description field. Both the title and the description appear in the clause library's Properties tab.
Print a list of the clauses	Click the Print button. HotDocs prints a list of all the clauses with their associated clause identifiers.
Open the component file for the clause library	Click the Component Manager button. (See Use Component Manager to Work with Components.)
View the properties of the clause or preview the clause text	Click the Properties tab or the Preview tab, respectively.
Search the list of clause names and clause descriptions for a specific string of text	Select Find and then type the text for which you are searching. HotDocs searches the clause names as well as clause descriptions. When it finds the text string, it lists only those clauses in the clause list until you clear Find .

Any time you make changes at the clause library file, those changes are automatically saved.

To access the clause library toolbar using the keyboard, press F10.

Create Clauses at the Template and Add Them to the Library

When the text for the clauses you want to create already exists in a single template, you can create individual clauses right in the template. Once you've defined the clause text, you can add the clause to the clause library.

To create the clause and add it to the clause library

- 1. At the template containing your clause text, select the text you want to make into a clause and click the **Clause Library** button. The **Clause Editor** appears, as well as the **Clause Library**.
- 2. Type a name for the clause in the **Clause name** field.
- 3. Enter a title for the clause in the **Title** field. The title will be used as the default title for the item once it is added to the clause library.
- 4. Optionally, click the **Resource** tab to provide users with information that can assist them in providing the correct answer. (See Add Resource Information to a Variable or Dialog.)
- 5. Optionally, click the **Notes** tab to enter notes about the clause, such as an explanation about why the clause was created or how it should function in the interview. (See Add Notes to Components for details.)
- 6. Click **OK**. The clause is added to the clause list in the library.

In Word, the word processor may insert an extra hard return at the end of a new clause. To fix this, select the clause at the clause library and click **d** Edit. The Clause Editor appears, where you can click Edit Clause Text. When your text appears, delete the extra hard return. (Click the Show / Hide ¶ button in the Word toolbar to view paragraph marks.)

Create and Insert Clauses Directly in Template

When the text for the clauses you want to create already exists in a single template, you can create individual clauses right in the template instead of at the clause library.

To create a clause and insert it into the template

- 1. Edit the template that contains the text you want to convert to clauses. (See Edit a Template.)
- 2. At the template, select the text for the first clause and click the **Select Field** button. The **INSERT Field** dialog box appears.
- 3. Select **INSERT Clause**.
- 4. Type a name for the clause component in the Clause to create field. (50-character limit.)
- 5. Optionally, select **INSERT clause only IF selected** if you want the user to control whether the clause is inserted.
- 6. Click **OK**. The **Clause Editor** appears.
- 7. Perform any of the following optional tasks:
 - Enter a title for the clause in the **Title** field. The title will be used as the default title for the item once it is added to the clause library.
 - Type the information about the clause you want the user to see in the **Prompt** field. (See Create a Prompt for a Variable.)

- If you selected **INSERT clause only IF selected** at the **INSERT Field** field, select **Yes/No on same line** to have *Yes/No* options appear on the same line in the interview.
- Click the **Resource** tab to provide users with information that can assist them in providing the correct answer. (See Add Resource Information to a Variable or Dialog.)
- 8. Click **OK** when you are finished. The instruction is inserted into the template.

In Word, the word processor may insert an extra hard return at the end of a new clause. To fix this, select the clause at the clause library and click **d** Edit. The Clause Editor appears, where you can click Edit Clause Text. When your text appears, delete the extra hard return. (Click the Show / Hide ¶ button in the Word toolbar to view paragraph marks.)

To insert an existing clause in the template, position your cursor where you want the clause inserted and click the **Field** button. Then select the clause by clicking the **Clause to insert** drop-down button. (The drop-down button is not available if you have selected text.)

You can add the clauses you have created to a clause library, which can be used to assemble any other document. See Add Existing Clauses to a Clause Library for details.

Clauses not only have a text value, but they have a true/false value as well. That means if you insert a clause in a template by using an **INSERT IF** instruction, the clause will automatically appear to the user as a yes/no question during assembly. If the user answers Yes, the clause will be included in the document; if the user answers No, the clause will not be included. You can also group clauses in a single selection or multiple selection group. See **Group Clause Components in One Custom Dialog** for details.

Insert a Clause Library into a Template

Once you add clauses to a clause library (see Add Existing Clauses to a Clause Library), you can insert the library into the template so it will appear during the interview. When the library appears, users can select and order the clauses they want to appear in the document. You can insert any clause library into the template, as long as the library, its clauses (stored in the clause archive (.HCL) file), the template, and the component file are all stored in a folder HotDocs can find.

To insert a clause library into the template

- 1. At the template, place the cursor where you want the library to be inserted. (If you've divided the whole document into clauses in the library, the template may show no text at all.)
- 2. Click the 🚈 INSERT Field button. The INSERT Field dialog box appears.
- 3. Select **INSERT Clause Library**.

4. Click the **Open** button next to the **Clause library to insert** field to browse for and select the file. Click **OK** when you are finished.

To insert a clause library that is located in a folder separate from its host template, you must include the folder path, or at least some portion of it. For example, a clause library can be inserted from a subfolder of the host template (*«INSERT "Subfolder\ClauseLib.hdl"»*); or it can be inserted using a full file path (*«INSERT "C:\My Documents\ClauseLib.hdl"»*); or it can be inserted using a reference path (*«INSERT "^referencePath\ClauseLib.hdl"»*).

Edit an Existing Clause

Once you have created a clause component, you can edit it at any time. Clause components can be edited from the template where the INSERT Clause instruction is, from the clause library that contains the clause, and from Component Manager. Likewise, when you edit a clause, you can edit both the properties of the clause component (for example, the prompt, the resource, and so forth) and the clause text itself.

To edit a clause

- 1. Open the **Clause Editor** using any of the following methods:
 - Select the clause at the clause library and click **Edit**.
 - In Component Manager, select the clause in the Components list and click definition Edit Component.
 - In the template, place your cursor in the INSERT Clause instruction and click the Insert Field button.
- 2. Make any changes to the clause component by editing the **Properties** or the **Resources**.
- 3. Optionally, to edit the clause text, click **Edit Clause Text** at the **Clause Editor**. (Click **Save and Close Template** on the HotDocs toolbar to save your changes to the text.)

In Word, the word processor may insert an extra hard return at the end of a new clause. To fix this, select the clause at the clause library and click **d** Edit. The Clause Editor appears, where you can click Edit Clause Text. When your text appears, delete the extra hard return. (Click the Show / Hide ¶ button in the Word toolbar to view paragraph marks.)

Changing the name of a clause in the *d* **Edit Clause** dialog will not change the name of the clause in the library. To do this you must also change the name in *Properties*.

Group Clause Components in One Custom Dialog

If your template contains several clauses from which the user must choose (meaning the clauses are inserted using INSERT IF instructions), you can group the clauses in one dialog instead of displaying each clause option in its own dialog. You can allow the user to select either one clause or multiple clauses.

To group clauses in a dialog

- At the template, create the clauses you want the user to choose from, making sure you select INSERT clause only IF selected at each clause's INSERT Field dialog box. (See Create and Insert Clauses Directly in Template.)
- 2. Create a dialog for the clauses. (See Gather Questions into a Custom Dialog.)
- 3. Once you have added the clauses to the **Contents** field at the **Dialog Editor**, click the **Options** tab. The window changes to show several custom options.
- 4. Click the **Selection grouping** drop-down button and select either **Select One** or **Select All That Apply**, depending on how many clauses you want your users to select.

Be careful not to create a custom dialog that contains variables from two or more different clauses. If the user selects one clause and not the other, HotDocs might present a dialog that asks for unnecessary information. If you're going to create custom dialogs, limit each dialog to include only those variables used in a single clause.

Automating Form Templates

Creating and Working with Forms

HotDocs Automator Overview

When creating a HotDocs form template (.HFT) file in previous versions of HotDocs, you had to use the HotDocs HFT driver, which was only supported for use with Windows 98 and Windows Me. Starting with the release of HotDocs 2006, these operating systems are no longer supported. Now, beginning with HotDocs 10, the functionality for creating and saving form templates in PDF format is integrated with HotDocs.

HotDocs Automator helps you create templates based on graphical forms—or forms that contain static text and graphics that cannot be changed or modified by users. These kinds of forms include loan applications, tax forms, or pre-printed court forms.

To create a PDF template, you first start with a PDF document¹. Some agencies or organizations with which you work may provide you with these PDF documents, or you can create your own using PDF-creation software, such as Adobe Acrobat Professional. You then use HotDocs to convert that file to HotDocs form template format.

When automating form templates, many tasks are identical to automating text templates—you must 1) identify each place on the form template where changeable information must be entered, 2) create fields, and then 3) attach variables to these fields. (See Introduction: Create and Customize Variables.)

You can create different types of fields, depending on what type of answer the field requires. For example, edit fields can be used to merge Text, Date, and Number answers, while check box fields can be used to merge True/False and Multiple Choice answers. In contrast, control fields control the flow of information but are hidden from end users. Resource fields, the final field type, allow you to create a resource for the entire form, which appears as a hyperlink on the form.

Edit and check box fields, with variables attached, allow users to provide information to complete the form. Sometimes a user's answer is longer than the space available. By designating overflow options, you can manage the problem of limited space on your form templates. When an answer overflows a field, HotDocs provides several ways to adjust the field or the answer. One solution is to create an addendum after the last page of the document and save the answers there.

Users assemble form document files from the form templates you develop. They can print blank versions of the forms, or provide answers for the questions and produce a finished document. After a form document is saved, users can edit answers in it by using HotDocs Filler. However, once the document is saved and then viewed in Filler, much of the template functionality is lost. For example, the form fields no longer process computations; they simply act as placeholders for the user's information.

Finally, you can test assemble your form templates to see how they will appear to users. (See Test Assemble a Document.)

¹ HotDocs supports the creation of templates using fillable PDFs. For details, see Introduction: Use Fillable PDFs as Form Templates.

At a Glance: The Automator Window



You can open the **HotDocs Automator** directly from the your start menu or by opening a form template to edit from a HotDocs Library.

The main part of the Automator window is the Template Workspace where you can use the options on the Toolbar to edit the form template.

At the top of the window is the Menu bar and Toolbar. From the Toolbar you can access the following buttons:

- **d**HotDocs Library: Opens the HotDocs library (if it's not already open) and brings it to the front.
- ***** New Template:** Creates a new form template file. (The new file won't contain any static text.)
- **Derived** Open Form: Opens a form template file for editing, or opens a saved form document for printing or editing.
- **Save Form:** Saves the current form. If the form has never been saved, HotDocs will ask for the path and file name to use.
- **Print Form:** Prints a copy of the form.
- Field Properties: Changes the properties for the selected field or fields.
- **AAnswer Font:** Changes the font for the selected field or fields.
- **Detect Field:** Detects the borders of the field that contains the cursor and creates the field if it doesn't already exist. If a field is selected, its borders are redetected.
- **Align Fields:** Displays the Align dialog box where you can align the selected field or fields.
- **Undo:** Undoes the last change you made to fields in the form. (You cannot undo changes made to HotDocs components (variables, dialogs, and so forth) or to answers in an assembled form document.)
- **Redo:** Reapplies any changes you have made to the template.
- **Variable Field:** Creates a new variable and attaches it to the selected field.
- **Edit Component:** Opens the Component Editor for the field that is currently selected.
- **Component Manager:** Displays Component Manager, which you use to work with variables, dialogs, clauses, and other HotDocs components.
- **Rest Assemble:** Test assembles a form document from the template.
- **Fill Tool:** Allows you to enter information in the field, such as an answer.
- **Select Tool:** Activates the Select Tool. This button allows you to select an existing field so that you can move it, resize it, or edit its properties (including the variable properties).
- Select Text and Fields Tool: In a HotDocs form template, activates the Select Text and Fields Tool, which allows you to either select a field or select static text in the form. You can select text, copy it, and then paste the text wherever you need it -- for example, in a variable's Prompt or Variable name box.
- **Scroll Tool:** Makes the Scroll Tool the current cursor. Use the cursor to grab the current form page and drag it around the window.
- **Zoom Tool:** Activates the Zoom Tool, allowing you to increase or decrease magnification of the template.
- **Page Width:** Displays the entire width of the current form page in the window.
- **J** Full Page: Displays the entire length of the current form page in the window.
- **Show Fields:** Shows or hides the field backgrounds.
- Show Variables: Shows or hides the variable names in the fields.
- **Show Thumbnails:** Shows or hides a thumbnail view of each page in the form at the left of the window.

- **Previous Page:** Moves to the page just before the current page in the form.
- **Next Page:** Moves to the page just after the current page in the form.
- **Interpretation** HotDocs Options: Opens the HotDocs Options dialog box where you can specify several preferences that control how HotDocs works.
- **Weighted Help:** Opens the relevant page of the HotDocs Help File.

You can access further options not available on the Toolbar in the Menu Bar.

To learn more about using HotDocs Automator follow the links below:

- Introduction: HotDocs Automator
- HotDocs Automator Toolbar
- Create a Form Template
- Save Existing HFT Files as PDF-based Form Templates
- Edit a Form Template
- Globally Change Properties in a Template or Group of Templates
- Save and Close a Form Template
- Define the Interview for a Form Template

HotDocs Automator Toolbar

The following table gives a summary of the Automator toolbar buttons, many of which are also on the assembly window toolbar:

it
ł
now the
, size, or

🕅 Detect Field	Scans the static content around the selected area for lines and text, and then moves the selected field into position. If no field is selected, creates a new field.
Align Fields	Displays the Align dialog box, where you can arrange the size and layout of multiple fields.
∽Undo	Removes the most recent changes you've made to the template.
🕶 Redo	Reapplies any changes you may have undone.
Variable Field	Displays the Variable Field dialog box, where you can attach a new variable to a field, or edit the merge field properties of an existing variable.
dit Component	Displays the Variable Editor , where you can edit the component properties of the selected variable.
Component Manager	Opens Component Manager, where you can work with the components used in the template.
Test Assemble	Starts a test assembly, showing how the current template will appear to users during assembly.
🚣 Fill Tool	Allows you to click on a field and enter a test answer.
Select Tool	Creates new fields (on double-click); also selects and moves existing fields.
Select Text and Fields Tool	(HotDocs form templates (.HFT) only) Creates new fields (on double- click); also allows you to select underlying template text.
🖑 Scroll Tool	Scrolls the template vertically and horizontally.
🔍 Zoom Tool	Changes the view, increasing or decreasing the magnification.
렆 Page Width	Changes the view to fit the left and right edges of the template in the window.
🛓 Full Page	Changes the view to fit the top and bottom edges of the template in the window.
🚄 Show Fields	Shows and hides the colored HotDocs variable fields.
Show Variables	Shows and hides the names of variables attached to each field. (Hiding variable names lets you enter test answers in the fields so you can test the field properties you've assigned. See Preview the Formatting of Answers in a Form Field.)
Show Thumbnails	Shows and hides the Thumbnails pane. You can click on a thumbnail (a small image of each page in the template) to move through the template.
Previous Page	Changes the view to show the previous page.
🗈 Next Page	Changes the view to show the next page.
HotDocs Options	Opens the HotDocs Options dialog box where you can customize how HotDocs works.
😨 Help	Opens the At a Glance: The Automator Window page of the Help file.

When you highlight a command on a menu or point to a toolbar button, a description of the command displays in the status bar at the bottom of the window.

You can hide the toolbar and the status bar by choosing **Toolbar** or **Status Bar** (**View** menu). To redisplay the toolbar or status bar, choose those commands again.

Create a Form Template

When creating a HotDocs form template (.HFT) file in previous versions of HotDocs, you had to use the HotDocs HFT driver, which was only supported for use with Windows 98 and Windows Me. Starting with the release of HotDocs 2006, these operating systems are no longer supported. Now, beginning with HotDocs 10, the functionality for creating and saving form templates in PDF format is integrated with HotDocs.

For instructions on creating a PDF form template from a fillable PDF document, see Create a Form Template Using a Fillable PDF.

Using HotDocs Automator, you can create templates based on electronic forms. (Forms are documents that contain underlying text and graphics that cannot be changed. Examples include loan applications, tax forms, and pre-printed court forms.) An electronic form can be created in virtually any Windows program from which you can print, such as a word processor, design, or form creation program.

Before you can automate a form template file, you must first convert an electronic form document into HotDocs template format. You do this by first printing the document to PDF format using the HotDocs PDF driver. You then save the PDF as a form template. If the document you want to automate is already in PDF format, you can simply save the PDF as a HotDocs PDF template.

You can also use Adobe Acrobat to create PDF documents.

Once you have created a form template file, you can automate it using HotDocs Automator.

Creating a PDF template requires two parts:

- Print the document as a PDF file.
- Save the PDF file as PDF-based form template.

To create a PDF template

- 1. If you already have a PDF document, skip to Step 11.
- 2. Open the document in a Windows program with printing capability.
- 3. Select the **Print** command for that program.
- 4. At the printing dialog box, select **HotDocs PDF Driver** as the printer.
- 5. Assign any properties you need to the PDF (such as security settings, watermarks, and so forth.)
- 6. Click the **Print** or **OK** button. The **Save As** dialog box appears.

- 7. If necessary, browse and select the location for the file.
- 8. In the **File name** field, type a name for the new file.
- 9. At the Save as type drop-down list, select PDF as the file type.
- 10. Click the **Save** button. The PDF file is created and stored in the location you chose. (You can close the original document.)

You will now create a PDF template from the document you just saved.

- 11. Start HotDocs. The template library appears. (See Start HotDocs.)
- 12. Click on a folder in the file list and click the **New Template** button. The **New Template** dialog box appears.
- 13. Click the **Type** drop-down button and choose **HotDocs PDF Template** from the list.
- 14. Enter a name for the file in the **File name** field and a title for the template in the **Title** field.

Do not type a template name that includes a pound sign (#). This can cause errors when uploading.

- 15. In the Initial contents group, select Other file.
- 16. Click the **Browse** button and browse to (and open) the PDF you created in the first part of these steps.
- 17. Click **OK** at the **New Template Initial Contents** field.
- 18. Click **OK** at the **New Template** dialog box. The new template is opened in HotDocs Automator.

When you create a new form template, HotDocs automatically creates a companion file called the component file that stores information about components you use in the template, such as variables and dialogs. The component file works behind the scenes, but both files—the form template and the component file—are necessary for the template to work. (The component file works exactly the same for a form template as a text template. See Introduction: Template and Component Files.)

For information on converting existing form template (.HFT) files to PDF-based templates, see Save Existing HFT Files as PDF-based Form Templates.

If you are comfortable working directly with file name extensions, you can also use Windows Explorer to change the file name extension of a PDF file from .PDF to .HPT. Be aware, however, that when you do this (as opposed to the steps outlined above), no component file will be created until you open the template and add components to it (either by creating variable fields or by opening Component Manager and adding components that way.)

Save Existing HFT Files as PDF-based Form Templates

When creating a HotDocs form template (.HFT) file in previous versions of HotDocs, you had to use the HotDocs HFT driver, which was only supported for use with Windows 98 and Windows Me. Starting with the release of HotDocs 2006, these operating systems are no longer supported. Since HotDocs 10, the functionality for creating and saving form templates in PDF format has been integrated with HotDocs.

With HotDocs, you can convert existing Envoy-based form template (.HFT) files into PDF-based form template (.HPT) files. The advantage of PDF-based form templates comes to end-users who want to use the templates to create PDF documents.

If you want to convert an existing .HFT file to a PDF-based template file, first check to see if you have the original source file (for example, the original Omniform file) and use that file instead to generate the PDF-based template. This will maintain the quality of the template.

To convert an HFT file into an HPT file

- 1. At the HotDocs library window, select the form template you want to convert, and click **Edit**. HotDocs Automator appears, showing the selected form template. (If the form template isn't included in the library currently showing, open the necessary library. See Open a Library.)
- 2. Select Save As (File menu). The Save As dialog box appears.

If you change the name of the new template, or change the folder in which the template is saved, you must also rename the .CMP file and move it to the new folder.

- 3. At the Save as type drop-down list, select HotDocs PDF Template (*.hpt).
- 4. Click Save. The Convert to PDF dialog box appears.
- 5. Optionally, provide information for the **Title**, **Subject**, **Author**, and **Keywords** fields.
- 6. Click **OK.** The PDF-based form template is created.
- 7. You can now open the new template in HotDocs Automator to review and work with the template. (See Edit a Form Template.)

At a Glance: The Find Dialog Box

Find Text	? 💌
Eind what:	A
B Match case	
	Find Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Find Text** dialog box by clicking **Find** in the **Edit** menu.

In the text field \overrightarrow{A} at the top of the dialog you can specify the text for which to search. You can also check the box \overrightarrow{B} below to limit the search to exact matches of the text.

To learn more about navigating a form template follow the link below:

• Move Around in a Form

The Find command works only in form template (.HFT) file

Move Around in a Form

When working with a form, there are several ways to move around within the form, including changing the magnification of the form, scrolling the form vertically and horizontally, and moving between pages in the form.

To move around in a form

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Complete any of the following steps:

То	Do This
Scroll the form horizontally and vertically	Click the Scroll Tool button and drag the mouse the direction you want to view. The form is scrolled in that direction.
Zoom in on a particular section of the form	Click the Zoom Tool button and then click on the form. The magnification on the form changes. (To zoom out, press the Ctrl or Shift key while clicking with the Zoom Tool button.)
View the next or previous page in the form	Click the Next Page button or the Previous Page button.
Go to any page in the form	Choose Go To (Edit menu). The Go To Page dialog box appears, where you can enter the number of the page you want to view. (You can also click a thumbnail image to move to any page in the form.)

Find static text on the	Choose Find (Edit menu). The Find dialog box appears, where you can		
form	enter the text for which you are searching on the form. (To perform a		
	case-sensitive search, select Match case.)		
The Find command			
works only in form	To find the next instance of text, choose Find Next (Edit menu).		
template (.HFT) file			

Edit a Form Template

You can open existing form templates in three different ways:

To edit a template from the HotDocs library window

- 1. At the HotDocs library window, select the form template.
- 2. Click **Edit**. The HotDocs Automator window appears, showing the form template.

To edit a template from the HotDocs Automator window

- 1. At the HotDocs Automator window, click the **Open Form** button. The **Open dialog** box appears.
- 2. Locate and select the form template you want and click **Open**. The Automator window appears again, showing the form template.

To edit a template from Windows Explorer

- 1. Using Windows Explorer, locate the form template you want.
- 2. Double-click the icon for that form template. The Automator window appears, showing the form template.

You can also drag a template icon from Windows Explorer and drop it into the HotDocs Automator window. The form will then open for you to edit.

At a Glance: The Insert Pages dialog box

Insert Pages	? 🔀
Form contaning pages to insert:	- []
BInsert pages 1 🛫 through 🚔 below 🔻 page	e 1 💼 of the current form.
(OK Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Insert Pages** dialog box by clicking **Insert Pages...** in the **Edit** menu.

In the first text field A you can enter the file path for the form you wish to take the pages from, or you can use the **Pagen** button to the right to navigate to the form you need.

Below this is a series of options B you can set to select which pages of the new form you need to insert and where in the current form you wish to put them. In the first number field type the page number of the first page you wish to insert, in the second field type the page number of the last page you wish to insert, from the third field you can choose to insert the pages above or below a specified page, and in the forth field you can specify that page in the current form.

To learn more about adding pages to a form template follow the link below:

• Add Pages to a Form Template

Add Pages to a Form Template

You can add pages to a form template. For example, say you need to add a page from one form template to the template you are currently automating. When you insert a page, Automator copies the static content and fields, and then pastes that information into the form template you're editing. The new page becomes part of the first template.

To insert pages from another template

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click **Allnsert Pages** (Edit menu). The Insert Pages dialog box appears.

3. Click the **Form containing pages to insert** drop-down field and select the file from which you want to copy the pages. You can also click the **Open** button to locate another file and add it to the list.

When inserting, file types must match, meaning you cannot insert pages from a PDF template into a HotDocs form template.

- 4. In the **Insert pages** group, specify the pages you want to insert and where they should be inserted in the current form.
- 5. Click **OK**. The template appears again, with the pages inserted.

If the pages you inserted contain variable fields, copy the components from the inserted template's component file into the current file. (See Use Component Manager to Work with Components.)

To add pages to a form document during assembly, see Attach an INSERT Instruction to a Field.

To quickly navigate through inserted pages, click **Thumbnails**.

At a Glance: The Delete Pages Dialog Box

Í	Delete Pages	•	•	? <mark>- x</mark>
	Delete pages	▲ 1 🚔 through	 I → from the current form. 	
			ОК	Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Delete Pages** dialog box by clicking **Delete Pages...** in the **Edit** menu.

In the first number field \underline{A} enter the first page in the range of pages you want to delete and in the second number field \underline{B} enter the last page in the range of pages you want to delete. To delete a single page, set this number to match the number in the first field \underline{A} .

To learn more about deleting pages to a form template follow the link below:

• Remove Pages from a Form Template
Remove Pages from a Form Template

At times you may want to delete a page from a form template.

To remove a page from a form

- 1. Edit the form template. (See Edit a Form Template.)
- 2. Choose Delete Pages (Edit menu). The Delete Pages dialog box appears.
- 3. Enter the range of page numbers you want to remove in the **Delete** and **through** fields.
- 4. Click **OK**. The page(s) you specified are deleted.

At a Glance: The Replace Static Content dialog box

Replace Static Content	2
Replace the static content of the current	form with the static content from:
	OK Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Replace Static Content** dialog box by clicking **Replace Static Content** in the **Edit** menu.

In this text field you can choose the form you wish to use to replace the underlying text of the current form. You can type in the file path, select the file from the drop-down list of recently opened items or navigate to the form using the Open button to the right.

To learn more about changing the text on a form template follow the link below:

• Update the Underlying Text in a Form Template

Update the Underlying Text in a Form Template

Frequently, the courts or other agencies to whom you submit your form documents update the content of their forms, which, in turn requires you to update the templates you are using. Rather than creating new

templates and re-automating them each time this happens, you can simply replace the text of the automated form with the updated content of the new form. HotDocs Automator provides you the tools for doing this.

To replace the static content of a form template

- 1. Edit the form template. (See Edit a Form Template.)
- 2. Choose Replace Static Content (Edit menu). The Replace Static Content dialog box opens.
- 3. Either select the form with the correct static content from the **Replace the static content...** dropdown list, or click the **Open** button and locate the file you want.
- 4. Click **OK** at the **Replace Static Content** dialog box. You are returned to the template.
- Reposition any fields, based on the updated static content. If the update significantly changed the form, check to make sure the text still corresponds to the fields and variables. If any new questions are asked on the form, create new fields and variables for them. (See Create a Form Field and Attach a Variable to a Field.)

If you are replacing static content in a fillable PDF template, it is recommended you check the fillable fields for placement and formatting after replacing the content. See Check Fillable Fields in a PDF Template for details.

View the Form in More than One Window

When you need to copy and paste fields, viewing the current form in multiple windows within Automator can be helpful. Also, when you have several forms open at once, you can view all open windows simultaneously.

To display the form in more than one window

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the **Window** menu and make a selection, based on information in the following table:

То	Do This
Open an additional copy of the current form	Choose New Window.
Display all open forms arranged from left to right	Choose Tile Vertical .
Display all open forms arranged from top to bottom	Choose Tile Horizontal .
Display all open forms so they overlap from top-left to bottom-right	Choose Cascade .

Globally Change Properties in a Template or Group of Templates

Before making large-scale changes to templates or component files, you should back up these files. Additionally, if you plan to use old versions of the templates, make copies before you use the Property Replicator feature. Unless you have recovery options specified at the HotDocs Options dialog box (see Protect Forms By Backing Them Up), you won't be able to retrieve the old copy. Another option is to have the template open when you apply a global change to it; then you can save the new file, then click **Undo** (**Edit** menu) to restore the original version.

In major projects, you may want to quickly change the properties in several templates. Using the Property Replicator, you can change the properties of regular fields, check box fields, forms, or addenda properties—or all four—for the templates you select. The changes will affect all the regular fields and check box fields in all the selected templates.

There are two parts to making changes in multiple templates:

- Create a source template that has the properties you want to apply to the other templates.
- Apply the global changes using the Property Replicator feature.

To change properties in a set of templates

- 1. At the Automator window, click the **New Template** button. A new, blank form template appears. Create two fields and arrange them in this order:
 - A first field slightly larger than a regular edit field. (You also can copy a regular field and paste it into the new template.)
 - A second field slightly larger than a check box field. (You also can copy a check box field and paste it into the new template.)
- 2. Apply any changes to these fields that you want to be made globally. (You may want to make a note of exactly which properties you changed. You will be asked in Step 9 to select these properties.)
- 3. Specify any template or addendum properties, again making note of any changes you make.
- 4. Click the **Save** button, and name the source template.

Make sure the check box field you create is larger than the check boxes in the templates you're changing. HotDocs applies check box changes to only those fields that are equal in size to or smaller than the check box you create. If the check boxes in the templates are larger than the one you create, HotDocs treats them as regular fields.

6. In the source template, click **Property Replicator** (**Tools** menu). The **Property Replicator** dialog box appears.

- 7. In the **Source template** field, click the **Browse** button to browse and select the source template. This file contains the formatting you want to apply globally.
- 8. In the **Destination templates** field, click the **Review Browse** button to browse and select the files.

You can also use wild cards to add templates. If you are changing all the files in a folder, you can use an asterisk (*) when you type the file paths in the Files to change field. The asterisk replaces the individual file names, and will add all files in a folder at once. For example, *C:\Documents and Settings\UserName\My*

*Documents**HotDocs**Templates**.*HFT* would add all the form templates in the template folder, and *C*:*Documents* and *Settings**UserName**My Documents**HotDocs**Templates**.* would add all the files in the template folder. To replace only a single character in a file path, you can use a question mark (?). For example, *C*:*Documents* and *Settings**UserName**My Documents**HotDocs**Templates**.*HF*? would add all the form templates and form documents.

- 9. Select the properties to be changed:
 - In the **Regular fields** list, select the properties of the edit field in the source template that you want to apply globally to all fields in the destination templates.
 - In the **Check boxes** list, select the properties of the check box field in the source template that you want to apply globally to all fields in the destination templates.
 - In the **Form** list, select the properties of the form or addendum in the source template that you want to apply globally to all destination templates.
- 10. After specifying the properties to be changed, click **Replicate**. The **Replicate Properties** dialog box appears, showing which template files have been modified.
- 11. Click **OK**. The **Property Replicator** dialog box appears again. It notes the files that were processed and reports how many files were changed.
- 12. Click Close.

You can use the Property Replicator to change the properties of linked fields in a fillable PDF template. Be aware, however, that if you save the document assembled from this template as a fillable PDF, it will use the field properties defined in the original fillable PDF. See Check Fillable Fields in a PDF Template for details.

Save and Close a Form Template

As you automate a form template, you should periodically save your template. This protects your work from unexpected program or system errors. Additionally, when you finish automating a template, you should save the template to disk and close it.

To save and close a form template

• Complete any of the steps in the following table:

То	Do This
Save changes to the currently open template and component file	Click the Save button.
Close the currently open template	Click Close (File menu).
Close all open templates	Click Close All (File menu).

You can have HotDocs automatically save back-up copies of form templates you are automating. See Protect Forms By Backing Them Up for details.

Define the Interview for a Form Template

When users assemble form documents, they can provide the information in one of two ways: answering questions in an interview or typing answers directly on the document (which is called direct-fill assembly).

All forms, when first automated, are set to allow only direct-fill assembly. However, you can allow interview-based assembly by specifying an interview. The interview can be a default interview (one in which HotDocs determines the order dialogs are asked in the template) or it can be a custom interview (one in which you specify the order dialogs and variables are asked).

To allow only direct-fill assembly of a form

- 1. Edit the form template. (See Edit a Form Template.)
- 2. Open Component Manager for the template. **Component Manager** appears. (See Open and Close Component Manager.)
- 3. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 4. Click the **Interview** tab. The view changes to show interview options.
- 5. Clear both the **Generate default interview** and the **Use custom interview** options.

To define an interview for the form

• See Define a Custom Interview or Have HotDocs Generate a Default Interview, depending on your preferences.

To make form templates created with HotDocs 2005 functional with HotDocs 11, you need to move the generate default interview properties from the template into the component file. To do this, open the form template in Automator; then click **File** > **Template Properties** > **Interview.** > **Move to Component File.**

Creating Form Fields

Create a Form Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

The first step in automating a form template is creating HotDocs fields at each place on the form where a user's information must be merged.

When you create a field, HotDocs determines the type of field you are creating based on the size of the field—if a field is smaller than a certain dimension, HotDocs creates a check box field. If a field is larger, then it creates an edit field. (You can define these dimensions at HotDocs Options. See Set Properties for New Edit Fields.) Additionally, once you create the field, you can make it a Resource hyperlink or a Control field.

Sometimes when you create a field, HotDocs can detect the borders of the underlying form and adjust its size to fit within those borders. This helps ensure the field fits best in the space allotted. See Detect Borders to Create or Resize a Field.

Once a field is created, you can attach a variable to it.

Please see Check Fillable Fields in a PDF Template for information on working with fields in a template created from a fillable PDF.

To create a form field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the **Select Tool** button.
- 3. Position the mouse pointer at one end of the intended field.
- 4. Press and hold down the left mouse button, then drag the pointer to the opposite corner of the field.
- 5. Release the mouse pointer. The field is created.
- 6. Optionally, click **Detect** to have HotDocs adjust the size of the field to more closely match the underlying static line or lines.
- 7. Optionally, you can customize the appearance of your fields in the following ways:
 - Attach a variable to the field. (See Attach a Variable to a Field.)
 - Change the field type. (See Change the Field Type.)
 - Adjust the size. (See Resize Form Fields.)

- Change the field's position on the form. (See Move a Field on a Form.)
- Make a field conditional so the variable is asked only if a condition is true. (See Make a Field Conditional.)

To move multiple fields once they have been created, select the fields and press the arrow key that indicates which direction you want to move them. To move fields more quickly, press the **Shift** key while pressing the arrow keys.

You can also create a standard-size field by double-clicking on the form. Or, you can click to set the field cross-hair and press **F8**.

Select a Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

In order to work with a field, you must first select it. When you select a field, handles appear on the field borders showing that you can edit the borders or other properties. You can select a single field to work with, or you can select multiple fields to group them or to make the same change to all of the fields.

To select fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the **Select Tool** button
- 3. Click on the field you want to work with.
- 4. Optionally, to select multiple fields, either press the **Ctrl** key while clicking each field, or press the **Shift** key, then press the mouse button and drag the mouse pointer to create a bounding frame. All fields included in or touched by this frame will be selected.
- 5. Use any of the following additional methods for selecting fields or canceling the selection, as described in the following table:

То	Do This
Select all the fields on the current page	Click Select All (Edit menu). (You can also press Ctrl+A .)
Cancel the selection of all selected fields	Click outside the fields, or press the Esc key.
Cancel the selection of only one of a group of selected fields	Hold down the Ctrl key and click that field.

Add fields to your group of selected fields without canceling the selection of those already selected Hold down the **Ctrl** key and click the new fields.

If you have trouble selecting the field you want, the field may be in a group. If this is the case, you must first ungroup the fields by clicking **Ungroup** (**Field** menu). (See **Ungroup Form** Fields.)

Click the **Show Fields** button to show and hide field colors.

Create a check box Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Some forms include check boxes for users to mark. check boxes can represent either true/false (or yes/no) options, but they can also represent several predefined options for users. By default, HotDocs merges an X in a check box to indicate that it has been selected, but you can define a different check box character.

Please see Check Fillable Fields in a PDF Template for information on working with fields in a template created from a fillable PDF.

To create a check box field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Where the check box appears on the form, double-click within the field borders, or click the **Detect Field** button. A check box field is created.
- 3. Attach a variable to the field. Your options include True/False and Multiple Choice.

When assigning a True/False variable, if you want to merge a character other than an X, indicate the character by specifying it as a variable format. See Customize a True/False Variable for details. To change the font for the field, see Set Properties for New check box Fields. When assigning a Multiple Choice variable, you should group the fields before attaching the variable. See Group Form Fields So Answers Can Flow From One Field to Another for details.

4. Optionally, specify a condition to make the field dependant on other answers in the form. (See Make a Field Conditional.)

You can change the default measurements used to identify new fields as check boxes. You can also specify a default character other than \mathbf{X} to be used when selecting check box fields. (See Set Properties for New check box Fields.)

If HotDocs creates an Edit field, rather than a check box field, select the field, click the **Field Properties** button, and choose **Check box** as the **Field type**.

Attach a Variable to a Field

Once you create fields on the form template, you can assign variables to them. Variables help control the type of answer the user enters. For example, a Text variable lets the user enter text, while a Date variable requires the user to enter a valid date.

To attach variables to fields

- 1. Edit a form template.
- 2. Create the field to which you want to attach the variable.
- Select the field and click the *Variable Field* button in the Automator toolbar. The Variable Field dialog box appears.
- 4. Create a new variable or select an existing variable. Your options of variable types include Text, Date, Number, True/False, Multiple Choice, and Computation.

To attach variables to fields using a wizard-like approach, choose **Create Variables** from the **Tools** menu. HotDocs scans through the template, looking for fields that do not yet have an assigned variable. One field at a time, HotDocs then displays the **Variable Field** dialog box for you to create variables for each field. You can also use this tool for linked fields in a fillable PDF template.

Other methods for attaching variables to fields include double-clicking on the field to display the **Variable Field** dialog box, dragging variables from Component Manager to the template fields, and viewing the **Field Properties** dialog box for the field and assigning the variable there.

To attach a single variable to multiple fields, for example, to create a **run-on group**, select and group all the fields before you click the ****Variable Field** button.

Attach an ASK Instruction to a Field

When HotDocs asks the questions in a form template interview, it asks the variables (in their associated dialogs) in the order it encounters the variable fields on the form—starting from the top-left section of the template and processing to bottom-right. At times, you may find that you want greater control over the order these dialogs are asked. To control this order, you can insert an ASK field in the template. For example, if you want to gather attorney information first in the interview, but the variables that ask the questions are located at the end of the template, you can insert an ASK instruction for the *Attorney Information* near the beginning of the template.

To cause a dialog to be asked a certain place in the form interview

- 1. Edit a form template. (See Edit a Form Template.)
- 2. At the place in the form where you want the dialog to be asked, create a field. (See Create a Form Field.)
- 3. With the field selected, choose ASK (Field menu). The ASK Field dialog box appears.
- 4. Enter the name of the dialog you want to be asked in the **Dialog** box and click **OK**. (See Gather Questions into a Custom Dialog.)
- 5. Optionally, change the field type to **Control** to keep users from typing in the field during direct-fill assembly. (See Change the Field Type.)

To attach more than one ASK instruction to a field, group the ASK instructions in a Computation variable script, and then attach the variable to the field. (See Put ASK Instructions in a Computation Variable.)

Make a Field Conditional

Some fields on a form should be answered only under certain conditions. For example, say you have a series of options (designated by check box fields). Each check box field is followed by an edit field where users can enter explanations for their selection. To keep users from entering an explanation without first selecting the corresponding option, you can attach a condition (or IF instruction or expression) to the explanation field that disables it until the user first selects the check box.

For more information on using IF instructions and expressions, please see Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression and Include or Exclude Multiple Versions of Text Using IF and ELSE IF Instructions or Expressions.

To make a field conditional upon other answers

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field you want to make conditional. (See Select a Field.)
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.

- 4. At the **Type** tab, click 😭 **IF Field** button. The **IF Field** dialog box appears.
- 5. Select **IF True/False Variable** or **IF Expression** depending on how complex the condition needs to be. The dialog box changes to show the necessary options.
 - For a simple IF instruction, type a name for a new True/False variable, or select an existing one.
 - For expressions, drag expressions and variables into the **Expression** field.

When a user assembles a document using this template, the fields you are conditioning will be asked only if the True/False variable or expression is true.

Attach an INSERT Instruction to a Field

When assembling form documents, you may need to attach another form to the current one. You can do this by using an INSERT instruction.

When an INSERT instruction is used in a form template, the questions from the inserted template are asked in the same interview as the parent template, but HotDocs appends the inserted document to the end of the parent document. (This is different from text templates where an INSERT instruction merges the text of an inserted document at the point of insertion in the parent document.) When inserting templates, file types must match—for example, you cannot insert a PDF template in a HotDocs form template, nor can you insert text templates in form templates.

To insert one form template into another

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a new field. (See Create a Form Field.)
- 3. With the new field still selected, choose **INSERT** (**Field** menu). The **INSERT Field** dialog box appears.
- Click the Comparison of the select the form template file to insert.
- 5. Click **OK**. The **INSERT Field** dialog box appears again, showing the name of the inserted template in the **Template to insert** field.
- 6. Click **OK**. The INSERT instruction is attached to the field.
- 7. Optionally, make the field a control field so the user won't see it during direct-fill assembly. (See Change the Field Type.)
- 8. Optionally, make the field conditioned on other answers in the interview. (See Make a Field Conditional.)

If you need to add content from one template to another, use the **Insert Pages** command. See Add Pages to a Form Template.

Detect Borders to Create or Resize a Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

When you create fields on a form, you often use the underlying static text as a guide as to where the field should be placed. Often, you want the borders of the fields to match the borders of the static text. To help you align these borders, you can use the **NDetect** feature. This feature is useful both when you create a new field and when you need to resize a field to fit within its allotted space.

If HotDocs is unable to automatically create or resize a field to the size and position you want, you must create the field manually. Several factors may contribute to these difficulties:

- **Insufficient surrounding features:** To detect a field, HotDocs searches for surrounding features, such as lines, text, or graphics. If there are insufficient surrounding features, HotDocs may have difficulty detecting a field. In such cases, HotDocs creates a field of the default size.
- Label text: When label text is inside the field area and there is enough space between the text and the bottom of the field area, the **Detect** command will extend the field up to the bottom of the label text. If you want the field to occupy the area to the left or right of the label, or if you want to include the label inside the field, you must create the field manually.
- **Field not completely visible:** If part of the intended field is not visible (for example, it's scrolled out of the window), HotDocs attempts to scroll to detect the field. It is recommended that you use a zoom level such as 😁 **Fit Page to Width** so HotDocs can find the entire field on the screen.

To detect the borders of the underlying static text

- 1. Edit a form template. (See Edit a Form Template.)
- 2. On the form, complete one of the following steps:
 - If the field is already created, select the field and click the **Detect** button.
 - If you are creating a new field, place your cursor where the new field should be created and click the **Detect** button.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Move a Field on a Form

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

As you create fields on a form, you will frequently need to move the fields. You can do this using the mouse or keyboard. You can also specify a precise location using the **Field Properties** dialog box.

To move a field using the mouse

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Position the mouse pointer over the selected field. The cursor changes.
- 3. Hold down the mouse button and drag the field to the new position.

To move a field using the keyboard

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select a field. (See Select a Field.)
- 3. Press the arrow keys to move the field one unit of measurement in that direction. (To move the field more quickly, hold down the **Shift** key as you press the arrow keys.)

To move a field using the Position/Size dialog box

- 1. Edit a form template. (See Edit a Form Template.) Edit a form document. (See Edit a Saved Form Document.)
- 2. Select a field. (See Select a Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. Click the **Additional** tab. The view changes to show the positioning options.
- 5. In the **Position** group, enter the desired distance from the left and top margins of the page in the **Left** and **Top** fields. (You can also click the up or down arrows for each field to change the distance.)

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

If a field you want to move is part of a group, you must first ungroup the fields. (See Ungroup Form Fields.)

The **Field Properties** dialog box includes the **First**, **Previous**, **Next**, and **Last** buttons to move you between fields. When you have multiple fields selected, these buttons are not available.

At a Glance: The Align Fields dialog box



After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Align Fields** dialog box by clicking the **Hign Fields** button..

There are four ways in which you can align fields: **Horizontal** \triangle , **Vertical** \boxed{B} , **Width** \boxed{C} and **Height** \boxed{D} . For each of these there are a variety of options and you can see an example of what the option will do in the **Example** field \boxed{E} to the right of the dialog.

To learn more about aligning fields on a form template follow the link below:

• Align Two or More Fields

Align Two or More Fields

When creating fields on a form template, you frequently need to align fields, either vertically or horizontally. Other times, you may need to adjust the height or width of a group of fields so they match each other. Aligning fields can give forms a more professional appearance, as well as minimize problems the user may experience when tabbing between fields during direct-fill assembly.

To align fields in relation to each other

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the fields you want to align. (See Select a Field.)
- 3. Click the Align button. The Align Fields dialog box appears. (You can also right-click and choose Align from the shortcut menu.)
- 4. Select an option in the **Horizontal** alignment group to align the fields horizontally, or the **Vertical** alignment group to align the fields vertically. The **Example** field previews the selected alignment option.
- 5. Click **OK**. The template appears again, and the selected fields are repositioned.

To size fields in relation to each other

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the fields you want to size. (See Select a Field.)
- 3. Click the Align button. The Align Fields dialog box appears. (You can also right-click and choose Align from the shortcut menu.)
- 4. Select the options you need from the **Width** group and/or the **Height** group. The **Example** field previews the selected resizing options.
- 5. Click **OK**. The template appears again, and the selected fields are resized.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Change the Borders and Margins of a Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

You can change the borders and margins of a form template field. Borders control the thickness of the field box, and margins control the distance between the border of the field and the answer text.

To change field borders or margins

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field whose borders or margins you want to change.
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. Click the **Layout** tab. The view changes to show layout options.

5. In the **Borders** and **Margins** groupings, make your selections. (Remember, borders affect the thickness of field boxes, while margins affect the distance between the border of the field and the answer in the field.)

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Copy One or More Form Fields

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Frequently, you need to copy fields on a form. For example, say you need to create a group of fields that all have the same properties. Rather than creating each field individually and modifying its properties, you can create a single field, assign the properties, and then copy the field to create the others in the group.

When a field is copied to a new place in the same template, all the field's properties (size, font, line formatting, fill order, etc.) are copied with it. Similarly, when variables, conditions, or REPEAT instructions are attached to fields, they are also attached to the copied fields. However, if you copy fields from one template to another, the variables, dialogs, and other components used in computations, conditions, or instructions are not copied to the new template's component file. You must copy these items manually. (See Create and Edit Multiple Components Simultaneously or Use One Component File for Multiple Templates.)

When you copy fields to a new location, the cursor position tells HotDocs where to paste the copied fields. If you copy a single field, the lower-left corner of the field will be at the cursor position. Likewise, if you copy two or more fields at the same time, the lower-left corner of an invisible boundary box drawn around all of the copied fields will be at the cursor position. If there is no cursor, the field is pasted on top of the original.

You can copy fields on only one page at a time.

To copy and paste one or more fields

- 1. Edit a form template. (See Edit a Form Template.) Edit a form document. (See Edit a Saved Form Document.)
- 2. Select the field or fields you want to copy. (See Select a Field.)
- Choose Copy (Edit menu). The field is copied to the Windows Clipboard. (You can also select the field and press Ctrl+C, or you can right-click on the field and choose Copy from the shortcut menu.)

- 4. Place the mouse cursor where you want the lower-left corner of the copied field to be, and then click the mouse to set the cross-hair.
- 5. Choose **Paste** (**Edit** menu). The copied field is pasted at that location. (You can also press **Ctrl+V**, or you can right-click on the field and choose **Paste** from the shortcut menu.)

If you need to automate an updated version of the template, it may be easier to replace the static content, rather than copy all of the fields. (See Update the Underlying Text in a Form Template.)

You can also copy a field by holding down the **Ctrl** key while dragging the field to its new location.

If you are working in a fillable PDF template, remember that linked fields are associated with underlying fillable fields in the PDF. If you plan to save the assembled document as a fillable PDF, answers associated with linked fields will only be merged where there are fillable fields on the form. This means that you cannot copy a linked field to a new location on the form and expect the answer to appear there in the resulting fillable PDF. If you want to use the same variable somewhere else in the form, create a regular HotDocs field and assign it the same variable name as the fillable field. Note, however, that in the fillable PDF, the answer will appear static.

Resize Form Fields

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

You may need to adjust the size of a field to fit it into the available space on the form. You can change a field's size using the mouse, the keyboard, or the **Position/Size** tab of the **Field Properties** dialog box.

To resize a field using a mouse

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Click a handle or field border. (When the mouse pointer is on a handle, the pointer changes to a double-arrow, indicating the directions the border can be moved.)
- 4. Hold down the mouse button and drag the border to a new position.

To resize a field using the keyboard

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)

3. Press the **Page Up** or **Page Down** keys to move the top border of the field up or down, and the **End** or **Home** keys to move the right border of the field right or left. (To resize the field more quickly, hold down the **Shift** key as you press the keys.)

To resize a field using the Field Properties dialog box

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. Click the **Additional** tab and, in the **Size** group, enter the desired distance from the left and top borders of the field in the **Width** or **Height** fields. (You can also click the up or down arrows for each field to change the size.) The field size is changed.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Change the Tab Order of Fields

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

When determining the order to ask questions in the interview, HotDocs begins with the top-left field of the form and processes all of the fields until it reaches the bottom-right field. This processing affects two things—the order questions are asked in the default interview and the order in which the user is able to tab through fields on the form.

Because of this default field ordering, sometimes you may find your dialogs being asked out of order. You may also find that the tab order during direct-fill assembly isn't working as you expect. Additionally, when automating a nontypical table, you may need to change the order in which fields are asked.

Once you specify a fill order for one field, you must specify the order for every field you want processed after that field. In effect, when you specify a fill order for just one field, you set that field to be answered last. This is because all the other fields are still set to *Row 0*, *Column 0*— they are lower than the *Row 1* or *Column 1*, so they will be filled first. Therefore, to change the order of the fields in the middle of the form, you must also change the order for every remaining field.

The following rules control field ordering:

- Fill order is determined first by row, then by column.
- All field values are initially set the same—Row 0, Column 0.
- When fields have different row numbers, fields with lower row numbers are asked first, for example, Row 0, Column 0; Row 1, Column 0; etc.
- When fields have the same row numbers, but different column numbers, fields with lower column numbers are asked first, for example, Row 1, Column 0; Row 1, Column 1; etc.
- Fill order is treated separately for each page of the template—you cannot specify a fill order that runs from one page to another.

To change the tab order for fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field. (See Select a Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. Click the **Additional** tab. The view changes to show ordering options.
- 5. In the **Row** field, type the row number.
- 6. If you need to specify the order for fields in the same row, type a number in the **Column** field.
- 7. Optionally, at the **Order dialog** box, you can click the **First**, **Previous**, **Next**, or **Last** button to save the current field's settings and display the next field's order.

By selecting and ordering different groups of fields, you can use fill order to handle various situations. For example:

- Remove all fill order settings: Select all fields and set the Row and Column numbers to 0.
- Organize large sections of fields: Select the desired fields and specify the order.
- Flow answers across fields contrary to the default order (top to bottom, left to right): Set the desired order, then group the fields. (See Group Form Fields So Answers Can Flow From One Field to Another.)
- **Fill table columns in an order different from the static text:** Set the desired order, then group the fields as a table.

To control when a variable is asked without changing fill order, you can use ASK instructions. For example, if you simply want a particular field to be asked first even though it isn't the first field on the form, place an ASK instruction in the top left corner of the page. (See Attach an ASK Instruction to a Field.)

Group Form Fields So Answers Can Flow From One Field to Another

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes an answer must begin in one field and continue in other fields. For example, you may have a Text variable whose answer must span two or more pre-printed lines. You can create edit fields for each line, and then group the fields so that answers flow from one field to another. Fields grouped this way are called run-on groups.

Once fields are grouped, HotDocs treats the group as a single field. This means that the Text variable and any conditions for the variable will be assigned to the grouping, not the individual fields in the group.

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot group fillable fields since necessarily, each fillable field is linked to a separate "variable." Answers—both in HotDocs and Adobe—must be entered on individual lines. This means that when you reach the end of one field, you must tab to the next field and continue typing your answer. (This inability to group fields is a limitation of Adobe. If you have access to the form, you may consider replacing individual fields in the group with a single, multiline field. Text you enter may not match the lines perfectly, but you will not have to tab between fields to enter the answer.)

To flow an answer across multiple fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create all the text fields across which the answer should flow. (See Create a Form Field.)
- 3. Select all the fields that will be used for the answer. (See Select a Field.)
- 4. Click **Group** (**Field** menu). A bounding frame appears around the grouped fields. (You can also right-click and select **Group** from the shortcut menu.)
- 5. Attach a variable to the grouped fields. (See Attach a Variable to a Field.)

When fields are grouped, you cannot change properties for individual fields. You must first ungroup the fields. (See Ungroup Form Fields.)

A form may require an answer to appear in a series of single-character fields or boxes (for example, a Social Security number). To place one character or digit in each field requires additional formatting. See Attach a Variable to Single-Character Boxes for details.

To group fields that appear on separate pages in the form, see Flow a Single Answer Across Two or More Pages in a Form.

By default, HotDocs asks questions in the interview by reading fields in the form from left to right, top to bottom. As it encounters a field, it asks the variable associated with it. If the variable is linked to a dialog, it asks the dialog instead. When directly filling the form, HotDocs tabs through the form fields using this same method. However, sometimes grouped run-on fields can create problems with the tab order. To ensure that text in a run-on group flows in the correct

order, you may need to specify a fill order for fields in the group. (See Change the Tab Order of Fields.)

Attach a Variable to Single-Character Boxes

Sometimes an answer field is made up of several single-character boxes. Rather than create a single field that spans these boxes, you can create individual fields in each box. Then, once you group the fields, you can assign a variable to the grouped field and the answer the user enters will be split across the fields—one character in each box.

For example, perhaps you have a field on your form that requires the user to enter the digits in a U.S. Social Security number. If you just simply group the fields, HotDocs attempts to fit the entire answer in the first few boxes, like this:

Social Security Number							
So Se	Nu						

When this happens, the answer will overflow and be sent to the addendum. However, if you group the fields and assign a maximum character number of one to the group, only one character will be merged in each box, like this:

So	Social Security Number								
s	0	с	i	a		I		s	e

To split an answer across multiple single-character boxes

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create and select the appropriate fields. (See Create a Form Field and Select a Field.)
- 3. Choose **Group** (**Field** menu). A bounding box appears around the grouped fields. (See Group Form Fields So Answers Can Flow From One Field to Another.)
- 4. Attach a variable to the group. (See Attach a Variable to a Field.)
- 5. Select the grouped field and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 6. Click the **Layout** tab. The view changes to show several alignment options.
- 7. In the Line Format group, enter 1 in both the Maximum lines and the Max chars/line fields.

When a user answers the variable, each character of the answer will be merged in its own field.

Sometimes the preprinted boxes have space for only the variable characters, not for static characters (such as the hyphens in a Social Security number, or the slash marks in date formats like *MM/DD/YYYY*). However, these characters normally merge as part of the answer. This would make the answer too long, again causing it to overflow. In such situations, you must specify an example format that prevents the static characters from being merged. If the answer is for a Text variable, specify a pattern (see Use Pattern Codes to Create a Custom Pattern) and then use an example format that uses an *X* for every character you want and an underscore for every character you want suppressed (for example, *XXX_XXX* to remove hyphens from a Social Security number). For a Date variable, create a format that eliminates the characters you don't want (for example, *061390*).

Attach a Multiple Choice Variable to a Group of check boxes

Frequently, a form template includes several options from which a user can choose. These options are usually represented by check boxes. In order to assign a Multiple Choice variable to these fields, you must first group the fields and then attach the Multiple Choice variable to the grouped field. Additionally, the fields in the group must match the order of the options in the variable. During the interview, when the user selects an option, the corresponding check box is selected.

If you are working with fields in a fillable PDF template, you can only group linked check box fields if each of the field's corresponding fillable field has the same name. To view a linked field's name, select the field and click the **Field Properties** button, and then click the **Additional** tab. The **Field name** drop-down list shows the associated name.

To assign a Multiple Choice variable to check boxes

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create the check box fields. (See Create a check box Field.)
- 3. Select the fields you want grouped. (See Select a Field.)
- 4. Choose **Group** (Field menu). The fields are grouped (as indicated by a bounding outline).
- 5. Double-click on the group. The **Variable Field** dialog box appears.
- 6. Assign a Multiple Choice variable, making sure the options for the variable correspond with the fill order of the check boxes. (If you need to assign a tab order to the fields, see Change the Tab Order of Fields.)

Sometimes the wrong box gets checked in the assembled form. If this happens, make sure the options are listed in the **Multiple Choice Variable Editor** in the same order as the fields' tab order. By default, HotDocs fills fields from top-left to bottom-right. If you want the fields processed in a different order, you can ungroup them and change the field order.

The default fill character for check boxes is an *X*. To merge a different character, enter the character in the **Merge Text** column for each multiple-choice option.

Ungroup Form Fields

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

When fields are grouped, you cannot modify the individual fields. You must first ungroup the fields.

To ungroup the fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select one of the grouped fields. The whole group is selected. (See Select a Field.)
- 3. Click **Ungroup** (**Field** menu). (You can also right-click on the field and choose **Ungroup** from the shortcut menu.)

Some properties that were assigned to the group, such as text, margin, and line format properties remain applied to the individual fields that made up the group. Other properties, such as a field order or a condition, are lost until you regroup the fields. Regrouping the fields restores the properties that were previously applied to the group.

Properties of a table, including the name of the repeated dialog, are attached to the first field in a table. Because of this, you should not delete the first field in the group before you regroup the fields. Also, do not close the form before you regroup the fields. If you do either of these things, these properties are lost.

At a Glance: The Answer Wizard Field dialog box

Answer Wizard Field			? 🔀
Contents Computation Variable DEFAULT	© <u>A</u> sk (© <u>N</u> one	Dialog	⊘ ASK Data <u>b</u> ase
Computation:	B		6.7
Parameter	Туре	Expression	
	D		
			OK Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Answer Wizard Field** dialog box by clicking the **Field Properties** button and clicking the **Wizard** button on the **Type** tab.

First you can decide what type of contents you need in the Answer Wizard Field. You can do this by selecting one of the options from the multiple choice list **A** at the top of the dialog:

Computation Variable: This option tells HotDocs to use a Computation variable. When the user clicks the wizard during assembly, HotDocs displays the variables and/or dialogs that are needed to provide the answer.

ASK Dialog: This option tells HotDocs to attach a dialog to a field in the interview. When the user clicks the wizard during assembly, the dialog you specify will be displayed.

ASK Database: This option tells HotDocs to attach a database component to a field in the interview. When the user clicks the wizard during assembly, the database component you specify will be displayed.

DEFAULT: This option tells HotDocs to ask the variables used to answer the question with default dialogs.

NONE: Clears any existing answer wizards that have been attached to the field. If you have multiple fields selected, this will clear all answer wizards.

In the text field **B** below you can enter a name for a new computation, database or dialog then click the **Edit Component** button **C**. Alternatively you can choose an existing component from the drop-down list on the text field.

If you select a computation which includes parameters you will be able to see them in the table \mathbf{D} below. To add a value or an expression to the parameter click the ... button \mathbf{E} to open the script editor.

To learn more about creating an answer wizard field follow the link below:

• Attach an Answer Wizard to a Field

Attach an Answer Wizard to a Field

When a user direct-fills a form document, he or she answers questions by clicking on the field and entering the answer in the field. Sometimes, however, the user may not be able to enter an answer in the field, either because the field requires other variables be answered to calculate its answer, or because it's conditioned on another field's answer in the document. To help users answer the questions necessary to fill in the field, you can assign an answer wizard to the field. An answer wizard attaches a button to the field that, when clicked, displays a pop-up interview asking the required questions.

For example, say you create a field that merges a user's age. To merge this answer, you attach a Computation variable that calculates the age based on the user's birth date and today's date. You can attach an answer wizard that asks the dialog that contains the birth date variable. Once the user answers the date, the answer is calculated and merged in the field.

Answer wizards are most often used with computation fields, conditional fields, or fields grouped as a table.

To create an answer wizard for a variable

- 1. Edit a form template. (See Edit a Form Template.)
- Select the field to which you want to attach the answer wizard, then click the Field Properties button. The Field Properties dialog box appears. (You can also right-click and select Field Properties from the shortcut menu.)
- 3. At the **Type** tab, click the **Wizard** button. The **Answer Wizard Field** dialog box appears.
- 4. Select the answer wizard option you want to use:
 - Computation Variable: Select an existing Computation variable from the Computation drop-down list, or click the *Litter Component* button to create a new one. During direct-fill assembly, when users click the *Answer Wizard* button, the variables used in the Computation are displayed in the pop-up interview. (If the variables are linked to a dialog, the dialog appears in the pop-up interview.)
 - **ASK Instruction**: Select an existing dialog from the drop-down list, or click the **Edit Component** button and create a new one. During direct-fill assembly, when users click the **Answer Wizard** button, the specified dialog is displayed in the pop-up interview.

- **DEFAULT**: When users click the **Answer Wizard** button, any dialogs containing the required variables are asked.
- NONE: Removes any answer wizards attached to the selected field (or group of fields).

You can also create a wizard for a table. A table wizard appears as a spreadsheet button (\blacksquare) instead of as an answer wizard button (\blacksquare), and clicking it will display the repeated dialog in a pop-up interview. To create a table wizard, select the table, click the **Field Properties** button, and create the wizard. You can still create an answer wizard for a field in a table by specifying the answer wizard before you group the fields in the table.

Changing Field Properties

At a Glance: The Field Properties dialog box (Type Tab)

Field Properties	? <mark>. x)</mark>	Field Properties [Table G	roup]
Type Layout Over	flow Additional	Type Layout Overflor	w Additional
AField type	BDisplay type	Field type	Display type
Edit	() <u>T</u> ext	() <u>E</u> dit	@ <u>T</u> ext
O Check box	🔘 Image	Check box	Image
© <u>R</u> esource	@ <u>B</u> ar code	<u>R</u> esource	🕐 <u>B</u> ar code
© <u>C</u> ontrol	© Strike-through	Control	Strike-through
	O Circled		O Circled
G Font	D Non-printing field	Font	Non-printing field
Variable:		Repeat:	
			**
Condition:		Condition:	
L			
Wizard:		Wizard:	
	G		
Answer text (for testin	g only):	Answer text (for testing o	nly):

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Field Properties** dialog box by clicking the **AFT Field Properties** button.

At the top of the **Type** tab there are two multiple choice lists you can use to set the field type. You can use the first one A to select the field type from **Edit**, **Check box**, **Resource** or **Control**. **Edit** fields can have answers entered directly during assembly, **Check box** fields allow the user to click to select one or more options in a list, **Resource** fields are uneditable during assembly and are used to give the user access to further information, and **Control** fields are non-printing fields that can be used to control the assembly of the form (See Create a Field for Behind-the-Scenes Tasks).

You can the use the second list **b** to set the display type for that field (these options change depending upon your choice in the Field type list **A**). If the Field type is **Edit** or **Control** the you will have the choice of **Text**, **Image**, **Bar Code**, **Strike-through**, or **Circled**. If you select **Check Box** you can choose from **Character**, **Strike-through**, or **Circled**. Finally, if you select **Resource** you can choose from **Hyperlink**, **Image**, or **Bar Code**. See below for explanations of each Display type option.

- **Text:** Create a field that will display a text answer.
- **Image:** The field will display an image; this image can be decided by a multiple choice question during assembly. See Insert a Graphic File in a Form.
- **Bar Code:** HotDocs will display the answer in the form of a bar code, for more information on setting the bar code options see At a Glance: The Field Properties Dialog Box (Barcode Tab) and Convert Answers on the Form to Bar Code Format.
- **Strike-through:** This field will allow the user to strike out the incorrect option from the underlying static text. See Cross Out Static Text on the Form.
- **Circled:** This field will allow the user to circle the correct option from the underlying static text. See Circle Static Text on a Form.
- Character: This allows you to use a character (like an X or a ✓ D
 ♦□ 𝔅□𝔄 𝔅 𝑘𝔅𝑘𝔅 𝔄□𝔅 𝔄⁰ ♦щ 𝑘 Create a Check-Box Field.
- **Hyperlink:** This type of field adds an uneditable hyperlink to the form, it can be used to direct users to further information they may need to correctly fill out the form. See Create a Resource Link on a Form.

Below the **Field** type list is the **Font...** button \subseteq , clicking this will open the **Font** dialog where you can choose the font type and size for this field. To the right of the **Font** button is a check box \bigcirc that you can tick if you would like this to be a non-printing field.

Underneath these options is a series of four text fields where you can choose to add features to the field. Click the ****Variable** button [] to create a variable for this field, the *****IF Field button [] to make the field conditional, and the ****Answer Wizard** button [] to create an answer wizard for this field.

The fourth field \mathbf{H} will be slightly different depending on which option you selected from the Display type list \mathbf{B} , but it is always used to designate the sample you would like to use for testing the field. For example that could be a sample image or piece of text.

If you are viewing the field properties for a table the ****Variable** button will be replaced with the **** Repeat** button where you will be able to attach a Repeat Dialog or Repeat Database (see image above).

More field properties options are available in the other tabs: Layout, Overflow, and Additional.

To learn more about setting field properties follow the links below:

- Preview the Formatting of Answers in a Form Field
- Show and Hide Fields
- Change the Field Type
- Change the Font Used for a Field
- Format Lines and Paragraphs of Text in a Form Field
- Rotate Answers in a Field
- Keep Contents of Form Fields from Printing
- Create a Field for Behind-the-Scenes Tasks
- Create a Resource Link on a Form
- Cross Out Static Text on the Form
- Circle Static Text on a Form
- Insert a Graphic File in a Form
- Convert Answers on the Form to Bar Code Format
- Attach an Answer Wizard to a Field

At a Glance: The Field Properties Dialog Box (Layout Tab)

Field Properties			?	X
Type Layout	Overflow	Addition	al	
Alorizontal	BVertica	ı—_(Rotation	
● <u>L</u> eft	© <u>Т</u> ор)	00 (
© <u>C</u> enter	<u>C</u> er	nter	90 90 90	
© <u>R</u> ight	<u> Bot</u>	tom	180°	
) <u>J</u> ustified	🔘 <u>J</u> us	tified	© 270°	
DBorders		Margins		
Left: 0.010"	×	Left:	0.01"	* *
Right: 0.010*	* *	Right:	0.01"	* *
Top: 0.010"	* *	Top:	0.01"	
Bottom: 0.010"	×	Bottom	0.03"	* *
[].ine format				
First line indent:	First line indent: 0.00" 📩 Maximum lines: 1 📩			
Lines per inch:	0.0	Max cha	ars/line: 0	-

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can view the **Layout Properties** by clicking the **Field Properties** button and clicking the **Layout** tab.

The three sets of multiple choice options at the top of this dialog deal with the formatting of the text within the fields. From here you can set up A the Horizontal position of the text within the field, B the Vertical position of the text within the field, and C the angle of Rotation of the text in the field.

Below that there are two sets of number fields where you can set the space allotted to borders and margins. Using the first set **D** you can set the width of the borders for the field and using the second set **E** you can set the width of the margins between the text and the field borders.

At the bottom of the dialog you can set options for $\boxed{2}$ Line Format. Using the number fields here you can set how much indent you wish to have on the first line, the amount of lines per inch in the field, the maximum number of lines in the field, and the maximum numbers of characters allowed per line.

You can change the units of measurements used by **HotDocs Automator** in the Form Documents section of **HotDocs Options**.

To learn more about the layout options for fields follow the links below:

- Preview the Formatting of Answers in a Form Field
- Format Lines and Paragraphs of Text in a Form Field
- Rotate Answers in a Field

At a Glance: The Field Properties dialog box (Overflow Tab)

Field Properties	Field Properties [Table Group]
Type Layout Overflow Additional	Type Layout Overflow Additional
AWhen field overflows	When table overflows
Shrink answer as needed to 8.0 🚔 points	G Send answers to <u>a</u> ddendum
Send answer to <u>a</u> ddendum Split answer	HWhen sending answers to addendum
Addendum entry options	Send entire <u>table</u>
Cross-reference text:	Send complete rows
B	Send individual <u>cells</u>
Addendum label text:	Addendum entry options
G	Cross-reference text:
Begin addendum entry on a <u>n</u> ew page	
Begin new page following addendum entry	Addendum label text:
Group name:	
Begin new addendum following this field	Begin addendum entry on a <u>n</u> ew page
	Begin new page following addendum entry
	KAnswer column indent: 0.33" 🚔
	Begin ne <u>w</u> addendum following this field
	L

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can view the **Overflow Properties** by clicking the **Area Field Properties** button and clicking the **Overflow** tab.

Depending on whether the field is a regular field, run-on field or a table the **Overflow** tab can have different options. The dialog on the left has the overflow options for regular fields and the dialog on the right has the overflow options for tables.

In the **Overflow** tab for regular fields the first set of options A designates what HotDocs will do when an answer overflows. You can choose to shrink the answer and the lowest font size it should be shrunk to, send the whole answer to the addendum, or split the answer between the field and the addendum.

Below that are two text fields, in the first field \underline{B} enter the text you want HotDocs to place in the answer field if the answer overflows and in the second field \underline{C} enter the label text you would like to appear before the answer on the addendum. Underneath the second text field are two check boxes \underline{D} where you can choose to have HotDocs begin this addendum entry on a new page and/or have the next addendum entry appear on a new page.

In the text field below you can enter a name group name for your addendum entry and in the check box below you can have HotDocs start a new addendum after this field.

The first option in the **Overflow** tab for tables is a check box G that you can tick to send overflowed answers to an addendum. Below that is a choice of three options H where, if the table overflows, you can choose if you would like HotDocs to send the entire table to the addendum, send complete rows to the addendum or send individual cells to the addendum.

Below that are the two text fields and check boxes exactly the same as for regular fields (\mathbb{B} , \mathbb{C} and \mathbb{D}). Just below this is a number field \mathbb{K} where you can enter the amount of centimeters you would like the addendum entry indented from the label.

The **Overflow** tab for a run-on field just has one change from the one for regular fields. You have the option to add an addendum entry indicator check box to the fields. (see below)

Addendum entry indicator No addendum entry indicator check box Indicator check box is first field in group Indicator check box is last field in group

To learn more about setting options for addendum properties follow the links below:

- Define Overflow Properties for a Regular Field
- Define Overflow Properties for Run-on Fields
- Define Overflow Properties for a Table
- Split a Multi-Line Answer Between the Form and the Addendum
- Create Independent Addendums for Forms

At a Glance: The Field Properties Dialog Box (Additional Tab)

Field Properties	? 💌
Type Layout Overflow	Additional
A ab order	
Row: 0 🚖	Column: 0
Bsize	
Width: 1.79" 🚖	Height: 0.20*
CPosition	
Left: 5.49" 🚖	Top: 0.44"
Linked Acrobat fillable field	
Field name:	▼

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can view the **Additional Properties** by clicking the *Additional Properties* button and clicking the **Additional** tab.

At the first set of number fields A you can set the Tab order for this field. This feature allows you to set up non-standard tables form unaligned fields.

Below that you can set the exact size of the field **B** using the two number fields to specify the width and height. The next set of number fields **G** allow you to specify the exact position of the field on the page. The first number field sets how far from the left side of the page the field will be and the second number field sets how far from the top of the page the field will be.

In the final text field **D** you can link this field to an Acrobat fillable field within the form by selecting the fillable field from then drop-down list or typing in the name.

You can change the units of measurements used by **HotDocs Automator** in the Form Documents section of **HotDocs Options**.

To learn more about the additional options for fields follow the link below:

• Merge a List of Answers in a Nonstandard Table

At a Glance: The Field Properties dialog box (Bar Code Tab)

Field Properties	? 💌
Type Layout Overflow A	dditional Bar Code
PDF417 Settings	
X-dimension (module width):	🗛 10.0 mil 📑
Module aspect ratio:	B auto 🗦
Number of rows:	🖸 auto 📑
Number of columns:	D auto 📑
Error level:	🔁 auto 🛛 🚔
Encoding:	F ANSI -

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can view the **Bar Code Properties** by clicking the **Field Properties** button and selecting **Bar Code** from the display type list at the **Type** tab, then by clicking on the **Bar Code** tab.

On this tab there are 5 number fields where you can change the bar code setting. The first field \triangle allows you to set the width of the narrowest bar in the bar code, in the second field \bigcirc you can determine the range of angles the bar code can be scanned from, in the next two fields \bigcirc and \bigcirc set the amount of rows and columns that appear in the bar code, and in the last field \bigcirc you can set how much room for error and bar code degradation you would like to leave.

In the drop-down list 🗧 below you can select the type of encoding you would like to use.

To learn more about the options for barcode fields follow the links below:

- Convert Answers on the Form to Bar Code Format
- Understand Bar Code Settings

Preview the Formatting of Answers in a Form Field

As you create fields in a form template, you frequently want to see how an answer will appear in a field. For example, say you create a multi-line field and you want to test whether a three-line address will fit. As you edit the field properties, you can enter a test answer and see how the answer fits in the underlying form field.

To test the assembly of the document—including how answers are merged in fields based on the underlying variables—click the **Test Assemble** button.

To enter test answers in a field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a new field or select an existing field. (See Create a Form Field and Select a Field.)
- 3. Click the Show Variables button to hide variable names. (See Show and Hide Fields.)
- 4. Click the *Field Properties* button. The Field Properties dialog box appears.
- 5. Change the different properties for the field at the **Type** tab as necessary.
- 6. In the text field at bottom of the dialog, enter the text you want to view in the answer field, depending on the field type.
- 7. Optionally, click the **Layout** tab of the **Field Properties** dialog box and make any changes to the field's alignment, borders, margins, and multi-line format.
- 8. When you're finished entering the text, click outside of the text field. The field in the underlying form updates to show the text you entered.

Show and Hide Fields

As you are viewing a form, field variables and backgrounds are shown by default. You can hide variable names and field backgrounds, though, which may make it easier for you to scan through the form. When

backgrounds are hidden, only the variable name or answer appears. When variables are hidden, only the test answer text appears (if any is assigned to the field).

To hide or show field backgrounds

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the **Highlight Fields** button.

To hide or show variable names

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click the Show Variables button.

You can control whether field variables and field backgrounds are shown or hidden when you open form templates documents. See Control How Forms Appear When Opened for details.

Change the Field Type

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

When you first create a field, HotDocs creates it as either an edit or check box field, depending on the field's height and width. check box fields are usually used to mark a yes/no response, while edit fields are used for questions that require text, date, or number answers. In addition, you can assign some fields as control fields, which means users can't access or edit the field during direct-fill assembly. You can also create resource fields, which can display helpful information about the form.

Once you choose a field type, you can determine the type of answer that is merged in the field. Your options include text, graphics, and bar codes. You can also use fields to cross out or circle static text on a document.

Type of Field	Default Color	Description
Edit	Yellow	Allows users to enter any type of answer, including text, dates, numbers, multiple choice options, or computed values. It is the most commonly used field type.
check box	Blue	Allows users to select from two or more options. Usually check box fields are associated with True/False variables or Multiple Choice variables. Answers in a

The following are the different types of fields you can create:

		check box field are usually indicated by an X or other character. (See Create a check box Field.)
Resource	Orange	Allows users to view helpful information about the form while directly filling the form. Resource fields appear as hyperlinks on the form. When users click the link, a pop-up window containing the useful information opens. (See Create a Resource Link on a Form.)
		Resource fields provide help for the form in general, instead of for a specific variable or dialog.
Control	Green	Allows you to complete "behind-the-scenes" tasks in the template, such as inserting templates or setting the values of variables. If the field contains answer or example text, it will be visible to users, but users won't be able to access the field.

If you are working with a fillable PDF template, linked fields appear using a light blue color. This distinguishes them from regular HotDocs fields. See Check Fillable Fields in a PDF Template.

To change the field type

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a new field (See Create a Form Field) or select an existing field. (See Select a Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. At the **Type** tab, make your selection, based on the information in the table above.

You can change the default colors used for form fields at the HotDocs Options dialog box. (See Change Colors in HotDocs Forms.)

Change the Font Used for a Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

You can choose the font that is used for answers on a form. When choosing the font, you can also indicate the style (for example, bold or italics), size, effects, and color that are used.

As is always the case when working with fonts, if the form requires a specific font be used, you must ensure the font is installed on all computers where the form will be viewed.

To change the font for a field or group of fields
- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field or fields whose font you want to change.
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. At the **Type** tab, click **Font**. The **Font** dialog box appears.
- 5. Make your font selections.

You can specify default font properties for all new fields you create at **HotDocs Options**. See **Set Properties for New Edit Fields** for more details.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Format Lines and Paragraphs of Text in a Form Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes you need to format the text within a multi-line text field. For example, perhaps you need to indent the first line of a paragraph, or maybe you need to adjust the number of lines that fit in an inch of vertical space. To do this, you can adjust the line format of a field.

To format lines and paragraphs

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a multi-line field.
- 3. Attach a multi-line Text variable to the field. (See Customize a Text Variable.)
- 4. With the field selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 5. Click the **Layout** tab. The view changes to show the different options for formatting the field.
- 6. In the **Line Format** group, complete one of the following options:

То	Do This
Indent the first line of text in the paragra	bh Enter how much space to include between the margin of the field and the first character of the answer in the First line indent field. (You can either type the number or click the up or down arrows.)

Force a certain number of lines to appear in an inch of space in the field	Enter the number of lines in the Lines per inch field.
Indicate how many lines can appear in the field	Enter the number of lines in the Maximum lines field. If the answer contains more lines than is specified, the field will overflow.
Indicate the maximum number of characters that can appear in a given line of text	Enter the number of characters in the Max chars/line field. When the number of characters exceeds this limit, text will wrap to the next line.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Rotate Answers in a Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

By default, answers in fields appear in horizontal rows and can be read from left to right. You can rotate text so answers will read from bottom to top, top to bottom, or upside-down from right to left.

To rotate text in a field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the fields you want to rotate. (See Select a Field.)
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. Click the Layout tab. The view changes to show field layout properties.
- 5. In the Rotation group, select a degree of rotation: 0°, 90°, 180°, or 270°, for example:

Full Name 0°	Full Name 90°	°081 əmsN Ilu7	Full Name 270°
--------------	---------------	----------------	----------------

You can enter test text in the test answer field (of the **Type** tab) to preview the selected rotation. See Preview the Formatting of Answers in a Form Field.

When you change the rotation, HotDocs views the bottom of the letters as the bottom of the field. This means that aligning the text horizontally or vertically may have different effects.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Keep Contents of Form Fields from Printing

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes you want field text to appear on the form during direct-fill assembly, but you don't want the text to appear when you print a copy of the form. For example, if you create a resource field, you may not want the resource field text to appear on the form when you print it. You can select an option that keeps this text from printing.

To designate that a field's contents shouldn't be printed

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create or select the field whose text you want to appear only on the form.
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. At the Type tab, select Non-printing field.

If you are working with linked fields in a fillable PDF template and you plan to save the document assembled from the template as a fillable PDF, any changes you make to the linked field will not be honored in the saved fillable PDF. See Check Fillable Fields in a PDF Template for details.

Create a Field for Behind-the-Scenes Tasks

Frequently, as you automate a form template, you need to perform a task or display text in a field that isn't accessible to the user. You can do this by creating a control field.

For example, perhaps you want to automatically insert a form template (such as an attachment) if the user answers a question a specific way. To do this, you must create a field on the form and attach a conditioned INSERT instruction to the field. However, when you create the field, by default, the user can access the field by typing in it. To keep this from happening, you can create the field (and attach the instruction) and then make the field a control field. This disables the field for the user.

Additionally, perhaps you want to include text in a field but you don't want the user to access the text during direct-fill assembly. If you make such a field a control field, the text will be visible to users, but users won't be able to access it.

To make a field a control field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create the field you want to use for behind the scenes tasks. (See Create a Form Field.)
- 3. Assign whatever field properties, variables, or instructions you need to the field.
- Select the field and click the Area Field Properties button. The Field Properties dialog box appears.
- 5. At the **Type** tab, select **Control** in the **Field type** group.
- 6. Optionally, you can resize the field so only the field borders are visible. (See Resize Form Fields.) This keeps any text in the field from appearing on the assembled document, which makes the field completely invisible.

If you are working with linked fields in a fillable PDF template, some fields are marked as readonly in the underlying PDF. These become control fields in the template. However, even if you change the field's type in HotDocs, it will remain a read-only field when you save the assembled document as a fillable PDF.

Create a Resource Link on a Form

When automating a template, you can assign resources to variables that help the user know how to answer a specific question in the interview. At times, however, you may want to include a resource for an entire form. To do this, you create a resource field. A resource field appears as a hyperlink on the form document. When clicked, the resource appears in a pop-up window. Resource fields appear during direct-fill assembly only.

To create a resource field, you must create a Text variable and assign as its resource the text you want users to view while directly filling the form document. You must then define the hyperlink text at the **Field Properties** dialog box.

To create a link field

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field, and attach a Text variable to the field. (See Create a Form Field and Attach a Variable to a Field.) (This Text variable will not be asked during the interview.)
- 3. At the Variable Field dialog box, click the **dialog box**, click t
- 4. Click the **Resource** tab and enter your resource text. (See Add Resource Information to a Variable or Dialog)
- 5. Click **OK** at the **Text Variable Editor** and the **Variable Field** dialog box. The Automator window appears again.
- 6. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears. (You can also right-click and choose **Properties** from the shortcut menu.)
- 7. At the **Type** tab, select **Resource** from the **Field type** group.
- 8. Choose one of the following options from the **Display type** group:
 - **Hyperlink** displays the resource as a link on which the user clicks. Enter the hyperlink text in the **Hyperlink text** field.
 - **Image** displays the resource as graphic file on which the user clicks. Enter the file name of the graphic in the **Image file name** field.
 - **Bar code** displays the resource in bar code format. Enter the text you want to encode in the **Bar code text** field.
- 9. Select Non-printing field. (This keeps the hyperlink from printing on the document.)

Cross Out Static Text on the Form

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes a form includes static text that must be crossed out, depending on answers the user enters during the interview. For example, say a form includes a list of medical conditions. Instructions on the form tell you to cross out any conditions that do not apply to you. You could manually cross out these conditions once you've printed the form; however, HotDocs allows you to create a strike-through field, which crosses out the text for you, depending on answers you enter during the interview.

A strike-through field, which is transparent, overlays the static text. You can choose the character that will be used to cross out the text.

To create a strike-through field in a form

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field that covers the static text on the form. (See Create a Form Field.)

- 3. Click the **Field Properties** button. The **Field Properties** dialog box appears. (You can also right-click and select **Field Properties** from the shortcut menu.)
- 4. Click the **Type** tab.
- 5. In the **Field type** group, select the type of field you want to associate with the strike-through field.
- 6. In the **Display type** group, select **Strike-through**.
- 7. In the **Variable** field, insert a True/False or Multiple Choice variable. This variable sets the conditions when static text should be crossed out. (See Attach a Variable to a Field.)
 - **True/False variable:** Select an example format that merges the strike-through character when the variable is true or false, as appropriate. (See Customize a True/False Variable.)
 - **Multiple Choice variable:** Type the strike-through character in the **Merge Text** field of the option that should cause the strike through, and type **NONE** in the other choices. (See Customize a Multiple Choice Variable.)

You can use the alignment controls on the **Field Properties** dialog box to position the strike-through text more accurately.

To change the font properties of the strike-through character, click the **Font** button at the **Field Properties** dialog box and make your changes there.

Circle Static Text on a Form

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes instructions on a form ask users to circle a preprinted option. Rather than make users print the form and manually circle the option using a pen, you can create a form field that overlays a circle on the text. You can attach a variable to the field so that the user can select which option should be circled during the interview. Creating circle fields not only preserves the professional look of the form, but allows the answer associated with the selection to be saved in the answer file.

To create a field that circles text

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field that covers the static option on the form you want to select. (See Create a Form Field.)
- 3. Click the *Field Properties* button. The **Field Properties** dialog box appears. (You can also right-click and select **Field Properties** from the shortcut menu.)
- 4. Click the **Type** tab.
- 5. In the **Field type** group, select the type of field you want to associate with the circled field.

- 6. In the **Display type** group, select **Circled**.
- 7. Assign a True/False or Multiple Choice variable to the field. (See Customize a True/False Variable, Attach a Multiple Choice Variable to a Group of check boxes, and Customize a Multiple Choice Variable.)
- 8. Optionally, to control the width of the circle's border, click the **Layout** tab and change the **Line thickness** in the **Circle** group.

To preview the circle (see Preview the Formatting of Answers in a Form Field), make sure you enter text in the **Answer text** field of the **Type** tab.

Insert a Graphic File in a Form

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes you may need to include an image on your form, such as a signature or seal. You can create a field and assign as one of its properties a graphic file. Supported file formats include .JPG, .BMP, and .PNG. Image files should be saved to the same folder as the template.

You control the conditions under which the image appears on the form. For example, you can attach a Multiple Choice variable or a True/False variable that merges an image file depending on which option a user chooses. Additionally, you can make the field a control field so that users can't access the field directly.

To create a graphic field using a variable

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field (See Create a Form Field) and attach either a Multiple Choice or True/False variable to it. (See Customize a Multiple Choice Variable or Customize a True/False Variable.)
- 3. If you're using a Multiple Choice variable, edit the variable and, in the **Merge Text** column, enter the file names of the graphics, depending on which options the user chooses.
- 4. If you're using a True/False variable, edit the variable and, in the **Format** field, enter the file name of the graphic, either preceded or followed by a forward slash (to indicate yes/no status).
- 5. Click **OK** at the variable editor and at the **Variable Field** dialog box.
- 6. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 7. At the **Type** tab, select **Image** from the **Display type** group.
- 8. Optionally, change the **Field type** to **Control** to restrict the user's ability to modify the field. (Change the Field Type.)

To create a graphic field without using a variable

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field. (See Create a Form Field.)
- 3. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 4. At the **Type** tab, select **Image** from the **Display type** group.
- 5. Click the **Open** button next to the **Image file name** field. The **Open** dialog box appears.
- 6. Locate and select the desired graphic file and click **Open**. The path and file name appear in the field.
- 7. Optionally, change the **Field Type** to **Control** to restrict the user's ability to modify the field. (Change the Field Type.)
- 8. Optionally, click the Show Variables button in the Automator toolbar to view the image.

The image is visible at the Form Document tab.

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot assign an image to a linked field. Instead, use a regular HotDocs field.

Convert Answers on the Form to Bar Code Format

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

You can create a two-dimensional bar code field that displays a single answer or group of answers. This may be useful if the organization to which you submit your documents requires certain data in the document be in bar code format. This allows the organization to scan the information and save it in some data retrieval system, such as a database.

For example, say you must submit a form to the court. When the court files the form, rather than manually entering case information (such as party names, case numbers, etc.), it can simply scan the bar code and have the information automatically entered in the system.

Typically, bar codes display data in a pattern of lines and formats, which should be readable by most hand-held laser scanners or charge-coupled device (CCD) scanners. The scanner should be able to read the size of the bars, which is 0.01 inch.

Use the following tips when working with bar codes:

- The format HotDocs uses for bar codes is PDF417, or Portable Data File 417. It is one of several recognized 2-D bar code formats.
- Bar codes in HotDocs can store about 1,500 characters, including numbers and other standard text symbols.
- Printing a form that has a bar code requires a high-density printer, such as a laser printer.
- Bar code fields can overflow, but all overflow properties assigned to the field are ignored. When too much text is entered for a bar code, no bar code appears. If you are having trouble getting the bar code to fit in the field, you can enlarge the field or reduce the field margins. (See Resize Form Fields or Change the Borders and Margins of a Field.)

There are two methods for creating a bar code:

- **Create a bar code using a variable:** When the answers you want to convert to a bar code must be supplied by the user, you can create a variable, such as a Computation variable, which will then merge the answers in the field and convert them to bar code format.
- **Create a bar code using predefined text:** If the bar code text is always the same, regardless of who is completing the document, you can enter that text at the **Field Properties** dialog box.

To create a bar code using a variable

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field and attach a variable to it, such as a Computation variable. (See Attach a Variable to a Field.)
- If using a Computation variable, create a script that will create a text string that includes all the answers you want encoded. Separate each answer by using a delimiter character, such as a tilde (~) or a vertical bar (|). (See Customize a Computation Variable and Introduction: Instruction and Expression Models.)

```
For example, Court Branch or District + "|" + Court Mailing Address + "|" +
Court City + "|" + Court Zip Code.
```

- 4. Click **OK** at both the variable editor and the **Variable Field** dialog box. The template appears again.
- 5. With the field still selected, click the *Field Properties* button. The **Field Properties** dialog box appears.
- 6. At the **Type** tab, select **Bar code** as the **Display type**.
- 7. Optionally, click the **Bar Code** tab and adjust any of the bar code settings. (See Understand Bar Code Settings.)

When the user assembles the document, the answer will be encoded as a two-dimensional bar code.

To create a bar code using predefined text

1. Edit a form template. (See Edit a Form Template.)

- 2. Create a field. (See Create a Form Field.)
- 3. Click the *Field Properties* button. The Field Properties dialog box appears.
- 4. At the **Type** tab, select **Bar code**.
- 5. Type the data you want encoded in the **Bar code text field**. As you type the text, HotDocs displays the bar code in the underlying form field. (You may need to click the **Show Variables** button in the Automator toolbar to hide variable names.) (If you are entering a group of answers that must be interpreted by the bar code scanner as individual answers, you must separate each answer with some sort of delimiting character, such as a tilde (~) or vertical bar (]).)

For example, Superior Court of Ada County | 12 N. Elm Street | Boise | 99999.

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot assign a bar code to a linked field. Instead, use a regular HotDocs field.

For information on changing the bar code specifications, see Understand Bar Code Settings.

To access the **Field Properties** dialog box, you can also right-click and select **Field Properties** from the shortcut menu.

Merge a List of Answers in a Standard Table

Sometimes you may need to merge a list of answers instead of a single answer in a form template. Often this list can appear in a predefined table in which users enter answers in rows sequenced from top to bottom and in columns sequenced from left to right.

You can also create what is called a nonstandard table, or one in which the rows are sequenced from left to right, instead of top to bottom. For details, see Merge a List of Answers in a Nonstandard Table.

When you create a standard table, you create and group fields in every cell, but you only attach variables to fields in the top row, like this:

Beneficiaries		
List your beneficiaries in the order you want them to benefit. The second beneficiary will only benefit if the first cannot. Likewise, the third beneficiary will only benefit if the first and second cannot.		
Name	Phone_Number	Relationship to Applicant
Beneficiary Name	Beneficiary Telephone	Relationship to Applicant

To create a list

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create a field in each empty table cell that will contain an answer in the assembled form. (See Create a Form Field.)
- 3. Attach variables to only the first row of fields—one variable for each column. (See Attach a Variable to a Field.)
- Select all of the fields in the table, then select Group as Table (Field menu). A bounding frame appears around the grouped fields. (You can also right-click and choose Group as Table from the shortcut menu.)
- 5. With the grouped fields still selected, choose **REPEAT** (**Field** menu). The **REPEAT Field** dialog box appears. (You can also double-click the grouped fields.)
- 6. Create a repeated dialog that contains the variables used in the table, or select an existing repeated dialog. (See Create a REPEAT Instruction to Gather a List of Answers.)

During assembly, if the user enters more answers than there are rows, those answers can be sent to the addendum, which you can customize.

You can filter answers merged into the table. (See Filter a List of Answers and Filter a List Using a Computation Variable.)

Information about the REPEAT instruction is associated with the field grouping. If you ungroup the table after you have inserted a REPEAT Instruction, remember to regroup the fields before you close the template. Otherwise, the information about the instruction is lost.

To create a list in a single variable field instead of a table, select the field and insert a Computation variable. In the computation, use a REPEAT instruction. (See Create a REPEAT Instruction Using a Computation Variable.)

Merge a List of Answers in a Nonstandard Table

Sometimes you may need to merge a list of answers instead of a single answer in a form template. Often this list can appear in a predefined table. Generally, these tables ask for information in a fairly standard order—in rows that read from left to right.

However, some lists require information that flows in different patterns, such as rows going from top to bottom. These kinds of tables are considered nonstandard. In nonstandard tables, you must specify a field order for every field in the table, then insert the variables in the fields you have assigned to Row 1. The following examples show types of nonstandard tables.

In this example, the rows are sequenced from left to right instead of top to bottom:

List information for up to four members	Member 1	Member 2	Member 3	Member 4
First Name	First Name			
Middle Name	Middle Name			
Last Name	Last Name			
Street Address	Address of Member			
City	City			
State	State			
Zip Code	Zip Code			
Phone Number	Phone Number			

In this example, the table is divided into sections, but each section is actually a row in the table:



To put a list in a nonstandard table

- 1. Edit a form template.
- 2. Create a field in each empty table cell that will contain an answer in the assembled form.
- Select all of the fields for the first row (the first series of fields) in the table, then click the Field Properties button. The Field Properties dialog box appears. (You can also right-click and select Order from the shortcut menu.)
- 4. Click the **Additional** tab, and in the **Tab Order** group, enter **1** in the **Row** field.
- 5. Click **OK**.
- 6. Select all of the fields for the next row in the table and set them to **Row 2**. Repeat this process for the remaining rows in the table, increasing the row number for each subsequent row.
- 7. Select all the fields for the first column in the table and set them to **Column 1**. Repeat this process for the remaining columns in the table, increasing the column number for each subsequent column.
- 8. Attach variables to the fields in Row 1.

- 9. Select all of the fields in the table and select **Group as Table** from the **Field** menu. A bounding frame appears around the grouped fields. You can also right-click and select **Group as Table** from the shortcut menu.
- 10. With the grouped fields still selected, choose **REPEAT** from the **Field** menu. The **REPEAT Field** dialog box appears. You can also double-click the grouped fields.
- 11. Create a repeated dialog that contains the variables used in the table, or select an existing repeated dialog.

Flow a Single Answer Across Two or More Pages in a Form

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

At times, a single answer field will start at the bottom of one page and continue to the top of the next page. However, HotDocs will not allow you to create a single field that spans across both pages. You can, however, create two fields—one on each page—and then link them using the **Group Fields** command.

To group fields on separate pages

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Create the first field in the series of spanned fields. (See Create a Form Field.)
- 3. Assign a Text variable to the field. (See Customize a Text Variable.)
- 4. Select the field and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 5. Click the **Overflow** tab, and in the **Group name** field, enter a name. Usually, this is the name of the variable, but it can be any text as long as it is the same for each field in the group.
- 6. On the following page(s), create the fields that will contain any overflow. Do not assign variables to these fields.
- 7. Repeat the preceding steps for each field in the group, using the same group name as you assigned to the first field.

Now, answers that do not fit in the first field will overflow to the next field named in the group.

To span text across multiple fields on the same page, see Group Form Fields So Answers Can Flow From One Field to Another.

Defining Form Overflow Properties

Define Overflow Properties for a Regular Field

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

When an answer is too large to fit in its field, HotDocs, by default, warns you and asks how to handle the overflow. You can customize the field overflow properties, controlling how HotDocs automatically handles answer overflows.

For example, you can have HotDocs automatically create an addendum, where each answer that overflows is sent. In the field that overflows, HotDocs can insert cross-reference text that points you to a specific item in the addendum. You can designate the text that is used both for the cross-reference and for the label in the addendum.

Another option is for you to have HotDocs automatically reduce the answer's font size. You can set a minimum point size and control if the answer is automatically reduced to this size before displaying an overflow warning message. (Be careful reducing the font size since others may find the smaller size harder to read.) (See Change the Font Used for a Field.)

How you define overflow properties depends on the type of field you are creating. For example, in an addendum, grouped fields and tables are handled a little differently from regular fields. See Define Overflow Properties for Run-on Fields and Define Overflow Properties for a Table.

To set overflow properties for regular fields

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the field whose overflow properties you want to define.
- 3. Click the **Field Properties** button and click the **Overflow** tab.
- 4. Complete any of the following steps:

То	Do This
Automatically shrink an answer that overflows to a specific point size	Select Shrink answer as needed and enter a point size in the points field.
Automatically send an answer that overflows to the addendum	Select Send answer to addendum . (To send only part of a multi-line answer to the addendum, select Split answer . (See Split a Multi-Line Answer Between the Form and the Addendum.)
Define the text that appears in the answer field when an answer is sent to the addendum	Enter the text in the Cross-reference text field. (See here for a list of reference numbering codes you can use.)

Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the Addendum label text field.
Make an answer the first item on its addendum page	Select Begin addendum entry on a new page .
Make an answer the last item on its addendum page	Select Begin new page following addendum entry.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Define Overflow Properties for Run-on Fields

When an answer is too large to fit in its field, HotDocs, by default, warns you and asks how to handle the overflow. You can customize the field overflow properties, controlling how HotDocs automatically handles answer overflows.

For example, you can have HotDocs automatically create an addendum, where each answer that overflows is sent. In the field that overflows, HotDocs can insert cross-reference text that points you to a specific item in the addendum. You can designate the text that is used both for the cross-reference and for the label in the addendum.

Another option is for you to have HotDocs automatically reduce the answer's font size. You can set a minimum point size and control if the answer is automatically reduced to this size before displaying an overflow warning message. (In reducing the font size, be cautious, because others may find the smaller size harder to read.) (See Change the Font Used for a Field.)

How you define overflow properties depends on the type of field you are creating. Grouped fields and tables are handled a little differently from regular fields. Grouped fields are shown in the addendum as just one answer.

To set the properties for a run-on group

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Select the run-on group whose overflow properties you want to define.
- 3. Click the **Field Properties** button and click the **Overflow** tab.
- 4. Complete any of the following steps:

То

Do This

Automatically shrink an answer that overflows to a specific point size	Select Shrink answer as needed and enter a point size in the points field.
Automatically send an answer that overflows to the addendum	Select Send answer to addendum.
	To send only part of a multi-line answer to the addendum, select Split answer .
Use a check box in the grouped fields to	Select either Indicator check box is first in
indicate the answer overflows.	field group or Indicator check box is last field in group, depending on where the check
Sometimes forms include a check box that	box that indicates overflow is positioned.
indicates when an answer is continued elsewhere, such as in the addendum or in an	Select No addendum entry indicator check
attachment. Choosing one of these options	box if there is no check box to indicate an
overflows.	addendum.
Define the text that appears in the answer field that an answer is sent to the	Enter the text in the Cross-reference text field. (Click Use Answer Overflow and Addendum
addendum.	Text Codes for a list of reference numbering
If you want the indicator check box to be the	codes you can use.)
only designation that a field overflows,	
enter NUNE IN THIS TIEID.	
Define the text that identifies the answer	Enter the text in the Addendum label text
once it has been sent to the addendum	field.
Make an answer the first item on its addendum page	Select Begin addendum entry on a new page .
Make an answer the last item on its addendum page	Select Begin new page following addendum entry.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Define Overflow Properties for a Table

If the contents of a table overflow, by default the answers go to the addendum as one item, even if only one answer overflows its field. In the addendum, the table's answers are organized in an outline using lowercase letters to identify each column. Thus, the first row of answers appear in the addendum as a set marked "a, b, c". The second row of answers appear as a second set of "a, b, c", and so forth.

To define overflow properties for a table

1. Edit a form template.

- 2. Select the table whose overflow properties you want to define.
- 3. Click the **Field Properties** button and click the **Overflow** tab.
- 4. Complete any of the following steps:

То	Do This	
Automatically send an overflowing answer (including extra rows in the table) to the addendum	Select Send answers to addendum.	
Define how much of the table is sent to the addendum when there is overflow	 In the When sending answers to addendum group, select one of the following options: Select Send entire table to send the contents of the table to the addendum any time any answer in the table overflows. The overflow cross-reference text (for example, See 1 in Addendum) is merged in the top row of the table and no other rows contain any answers. Select Send complete rows to send only rows that overflow to the addendum. For example, if an answer in a row overflows, the entire row goes to the addendum. The overflow cross-reference text (for example, See 1 in Addendum) is merged in the affected row only. Select Send individual cells to send only the contents of a single overflowing field to the addendum. The overflow cross-reference text (for example, See 1 in Addendum) is merged in the affected cell only. 	
Define the text that appears in the answer field when an answer is sent to the addendum	Enter the text in the Cross-reference text field. (Click here for a list of reference numbering codes you can use.)	
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the Addendum label text field.	
Make an answer the first item on its addendum page	Select Begin addendum entry on a new page .	

Make an answer the last item on its addendum page	Select Begin new page following addendum entry .
Indent the answer a specific amount of space	Enter a number in the Answer column indent field (or click the up and down arrows to select a number).

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

At a Glance: The Addendum Properties dialog box

Addendum Properties	? <mark>- x</mark> -
APage margins	1.00" ≑
BHeader properties	
Addendum <admpageinfo></admpageinfo>	
Font Height: 0.67" 📥 Alignment: Center 🗸	«»
CFooter properties Footer text:	
Font Height: 0.33" 🚖 Alignment: Right 🗸	• •
Addendum entries Font Indentation: 0.33 Space between:	0.00"
Number lines to format as pleading paper	Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Addendum Properties** dialog box choosing **Addendum** from the **Template Properties** list in the **File** menu.

The first options you can set is the minimum distance for the Page margins A you can set these by changing the values in the four number fields: Left, Right, Top, and Bottom. In the next section B you can set the options for the addendum header. In the text field you can enter the text and codes you would like to use, then using the options underneath you can choose the font you would like to use in the header, how tall the header should be, the text alignment and if you click the ****Variable** button you can create a variable to be included in the header. The third section has exactly the same options but instead they control the addendum footer.

The fourth section **D** contains the options for Addendum entries. You can choose the font, amount of indentation between the label and the answer, and the amount of space between sets of answers.

At the bottom of the dialog is a check box format the addendum as pleading paper.

To learn more about setting options for addendum properties follow the link below:

• Customize the Look of the Addendum

Customize the Look of the Addendum

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes when assembling a form document, an answer is too long for the form field. One way to resolve this overflow is to send the overflowing text to an addendum, which is a section of the form designed to display such answers. You can modify the appearance of the addendum—for example, you can choose what page margins to use, define what text appears in the headers and footers, and choose the font properties for the text that is used in the addendum. You can also define how pages are numbered in the addendum.

To change the addendum format

- 1. Edit a form template. (See Edit a Form Template.)
- 2. Click File > Template Properties > Addendum. The Addendum Properties dialog box appears.
- 3. Complete any of the following steps:
 - To Do This

Define how much white space there is between the addendum page edges and the text in the addendum	Enter the dimensions in the Page margins group.
Define the text that appears at the top of	Enter the text in the Header text field.
each addendum page	To learn how to merge page numbers in the header, see Use Answer Overflow and Addendum Text Codes.
Define the text that appears at the bottom	Enter the text in the Footer text field.
or caen adaenaam page	To learn how to merge page numbers in the footer, see Use Answer Overflow and Addendum Text Codes.
Change the font properties (including font face, size, and style) of the header or footer text	In the respective Header properties and Footer properties groups, click the Font button and make the changes at the Font dialog box.
Define how much vertical space the header or footer text requires	In the respective Header properties and Footer properties groups, enter a number in the Height field.
Change the alignment of the header or footer text	In the respective Header properties and Footer properties groups, click the Alignment button and choose your alignment option.
Insert a variable in the header or footer text	In the respective Header properties and Footer properties groups, click the ** Variable Field button. The Variable Field dialog box appears, where you can define the variable whose answer you want merged in the header or footer.
Define the font properties (including font face, size, and style) for answers that appear in the addendum	In the Addendum entries group, click the Font button and make your changes at the Font dialog box.
Define how much space there should be between the addendum label and the answer	In the Addendum entries group, enter a number in the Indentation field.
Define how much space there should be between each answer in the addendum	In the Addendum entries group, enter a number in the Space between field.
Make the addendum appear as a pleading paper	Select Number lines to format as pleading paper .

Once an answer has been sent to the addendum, you cannot edit the actual addendum—you must either modify your answers at the **Form Document** tab or **Interview** tab, or you must send the addendum to the word processor (choose **File > Send Addendum To > Word Processor**). You can also send the addendum to the Windows Clipboard (choose **File > Send Addendum To > Clipboard**) to paste it into a different program for editing.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Split a Multi-Line Answer Between the Form and the Addendum

When creating multi-line fields, you can select an overflow option that sends either all of the answer to the addendum, or only the part of the answer that doesn't fit in the field to the addendum.

If you allow the user to choose whether to split the answer between the form and the addendum, you can customize the reference and label text, based on their selection. For example, if the user chooses to send the entire answer to the addendum, you can specify *See Addendum 1* as the cross-reference text. However, if the user chooses to send just part of the answer, you can specify *Continued in Addendum 1* as the reference text.

To choose overflow options for a multi-line answer

- 1. Create a multi-line field. (See Create a Form Field.)
- 2. Select the field and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 3. Click the **Overflow** tab. The view changes to show overflow options.
- 4. Complete any of the following steps:

То		Do This
Always sen without pro	d the answer to the addendum ompting the user	Select Send answer to addendum.
		To send just the part of the answer that doesn't fit in the field to the addendum, select Split answer .
Specify the field as wel chooses just	text that will be merged in the I as the addendum if the user It to send to addendum	Enter the text in the Cross-reference text and Addendum label text fields, respectively.



To specify alternate text that will be merged in the field and addendum if the user chooses to split the answer between the form and the addendum, type a vertical bar (|) and then enter the alternate text after the bar. For example:

Cross-reference text: See Addendum 1|Continued in Addendum 1

Addendum label text: Addendum 1|(con't) Addendum 1

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Create Independent Addendums for Forms

These instructions can also be used to create a form field at both the **Form Document** tab of the assembly window and in HotDocs Filler.

Sometimes your form includes duplicate pages that must be given to or filed with a specific party. (For example, perhaps you have a form where one copy must be given to the client, and another copy must be filed in your office.) You can select an option that specifies that a new addendum page should begin following a specified field. This makes it possible to create multiple, independent addendums.

To create independent addendums

- 1. Select the field(s) and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 2. Click the **Overflow** tab. The view changes to show overflow options.
- 3. Select Begin new addendum following this field.

Fillable PDFs do not support the creation of an addendum when a field overflows. This means that if a field in an assembled fillable PDF template overflows, the answer in the saved PDF document will simply appear cut off. Template users should always review and modify any answers that may potentially overflow before saving the assembled document as a fillable PDF.

Use Answer Overflow and Addendum Text Codes

These instructions can also be used to control numbering in overflow and addendum text both at the **Form Document** tab of the assembly window and in HotDocs Filler.

As you customize the appearance of an addendum, you can insert any combination of codes to merge page numbering into addendum headers and footers. Also, when defining overflow cross-references and labels, you can insert codes that number entries in the addendum, particularly answers in tables that overflow. Finally, in both cases, you can control the type of number that is used—Arabic, Roman, or alphabetic.

Addendum Header and Footer Page Numbering Codes

You can use the following codes to merge page numbers in addendum headers and footers. For example, the header text Financial Aid Application, page <PAGE> would insert at the top of each addendum page the form title followed by the number of the current page in the form (that is, Financial Aid Application, page 1). The footer text Page <PAGE> of <PAGE> would insert at the bottom of each addendum page the number of the current page followed by the total number of pages in the form (that is, Page 1 of 7).

Page Numbering Code	What It Inserts in Assembled Document
<page></page>	Current page in the document
<pages></pages>	Total number of pages in the document, including addendum pages
<docpages></docpages>	Total number of pages in the document, without including the addendum
<pageinfo></pageinfo>	Current page in the document / Total pages in the document
<admpage></admpage>	Current page in the addendum
<admpages></admpages>	Total number of pages in the addendum
<admpageinfo></admpageinfo>	Current page in addendum / Total pages in addendum.

Addendum Label and Cross-Reference Text Numbering Codes

As you define how HotDocs should handle fields that overflow, you can use the following codes to merge reference numbers in cross-references and addendum labels. For example, in a table with two rows and two columns, the number codes <ROW>, <COLUMN> would insert 2, 1 for the answer in the first column of the second row. For that same cell of the table, <MEMBER> would insert 3, since the field is the third field in the table counting from the first cell of the table. And again for that answer <MEMBER:A> would insert C, the item number as an uppercase alphabetic character (1=A, 2=B, C=3, etc.).

Numbering Code What It Inserts in Assembled Document

<refnr></refnr>	Addendum item number
<varname></varname>	The name of the variable
<row></row>	Table row number (merged in addendum only)
<column></column>	Table column number (merged in addendum only)
<member></member>	Table cell sequence number (merged in addendum only)

Numbering Format Codes

To change the format of the number, between the page or referencing number code and the closing angle bracket (>), type a colon (:) immediately followed by one of the page format codes described in the following table. (A complete entry would look like this: <DOCPAGES:A>.)

Page Format Code	What Number is Merged
1	Arabic numerals (default)
Α	Uppercase letters
а	Lowercase letters
1	Uppercase Roman numerals
i	Lowercase Roman numerals

Creating and Using Fillable PDF Form Templates

Introduction: Use Fillable PDFs as Form Templates

You can use fillable PDF documents as the basis for creating form templates. A fillable PDF is an interactive PDF document that allows users to enter answers on the form while viewing it in Adobe Acrobat or Reader. As you create the template, you will be prompted for instructions on how to use the fillable PDF. You can either create a fillable PDF template, or you can create a regular PDF-based form template.

Create a Fillable PDF Template

When you create a template from a fillable PDF, you can use the fillable fields from the Adobe document as a point of reference for automating the template. In fact, during template creation, HotDocs will automatically create fields where it detects fillable fields. These special fields are known as linked fields, since they link to and use the properties of the fillable fields.

Users can then assemble a fillable PDF template and save the completed document as a fillable PDF document. This allows them to edit answers directly in the PDF. Users can also save the assembled document as a regular (static) PDF or as a HotDocs PDF document.

When using fillable PDF templates, please note the following:

First, when saving the assembled document as a fillable PDF, if you changed any of the default field properties for the linked field, those properties will not be used in the saved document. Such properties include field position and size, font, and borders, just to name a few. Adhering to the default properties defined in the original PDF document ensures the saved document will meet any requirements specified by the agency that issued the form. Note, however, that if you save the assembled document as a regular PDF or HotDocs PDF Document, any custom properties you specified will be used for the fields instead.

Second, keeping fillable fields in the template makes it easier to match HotDocs fields to the original fillable fields, especially when you accidentally move, change, or delete a HotDocs field while editing the template. This precision may be important if you plan to submit the form to an agency that requires that answers be entered a very specific way, regardless of whether you save the document as a fillable PDF or otherwise.

Finally, when you create variables and attach them to linked fields in a PDF-based template, HotDocs may suggest some default properties for the variable, such as a name, format, pattern, and so forth. Where possible, you should accept these default properties. If you save the assembled document as a fillable PDF, Adobe Acrobat or Adobe Reader will try to format the answer coming from HotDocs so that it fits in the field as it is defined by Adobe Acrobat. If the answer contains text or uses properties that are different than what it expects, the field text may appear incorrectly. This is especially true with Number and Date formats.

Create a PDF Form Template

When creating a form template using a fillable PDF, you can choose *not* to keep fillable fields in the template. If you do, the template will be created as a regular, static PDF template, and users of the form will not be able to save it as a fillable PDF.

HotDocs supports only fillable PDFs (or Acrobat Forms or AcroForms) created with Acrobat Professional. It does not support any fillable PDF documents created or edited using Adobe LiveCycle Designer. (This includes XFA forms, XML Forms, or Designer Forms.) A small subset of XFA forms might still work, as long as there are no dynamic form features being used. Typically, if an AcroForm has been edited using LiveCycle Designer and only properties of fields have been changed, it is likely to still work in HotDocs.

Creating a Form Template Using a Fillable PDF

HotDocs supports only fillable PDFs (or Acrobat Forms or AcroForms) created with Acrobat Professional. It does not support any fillable PDF documents created or edited using Adobe LiveCycle Designer. This includes XFA forms, XML Forms, or Designer Forms. Please note that a small subset of XFA forms might still work, as long as you aren't using any dynamic form features. Typically, if you have edited an AcroForm using LiveCycle Designer and only changed properties of fields, it is likely to still work in HotDocs.

Some PDF documents you receive from various organizations contain fillable fields, which enable you to enter the data required by the document while you are viewing the PDF in Adobe Acrobat or Reader. You can create HotDocs form templates based on these fillable PDFs.

For example, say you download a fillable PDF from a court system. You can create a HotDocs PDF template using this fillable PDF as the basis for the template. When you do, you can choose to keep the fillable fields in the template, or you can choose to remove them from the template. If you choose to keep them, HotDocs creates special HotDocs fillable fields (or linked fields) that mirror exactly the underlying fields. This is useful if you need to file the resulting document with an agency that requires you to provide answers in very precise places on the form.

Keeping fillable fields in a form template also makes it easy to re-adjust fields back to their original position and format, especially if you accidentally move, change, or delete a linked field while editing the template.

If you choose to remove the fillable fields, HotDocs still creates form fields where the fillable fields were however, they are no longer associated with the fillable fields. Additionally, users are not able to save the document as a fillable PDF, and the PDF will be static.

Once you create a template that contains fillable fields, you can attach variables to those fields. When you do, HotDocs attempts to use information from the original fillable PDF to suggest what type of variable to create. It also suggests other properties, such as variable names and answer formats.

Finally, when assembling a form document that contains fillable fields, you have three options for saving the document:

- You can save the document as a regular HotDocs PDF document (.HPD) file. If you select this option, you can view the file and edit the answers in it using HotDocs Filler only.
- You can save the document as a static Adobe PDF document. If you select this option, you can only view the document in Adobe Acrobat, Reader, or HotDocs Filler. You cannot edit the answers. HotDocs removes any fillable fields that were present in the template when you save the document.
- You can save the document as a fillable PDF. If you select this option, you can edit the answers in the document in Adobe Acrobat or Reader.

To create a PDF template from a fillable PDF

- 1. Start HotDocs.
- 2. Click on a folder in the file list and click **New Template**.

- 3. Click the **Type** drop-down button and choose **HotDocs PDF Template** from the list.
- 4. In the File name field type a file name for the new template. If you decide to store the template somewhere other than the default template folder (as seen in the Target folder field), you can also **Browse** to the new destination. HotDocs adds the correct extension to the file name based on the type of text template you selected. HotDocs displays the full path of the currently selected template location in the Target folder field.
- 5. Click in the **Title** field and either accept the suggested title or replace it with your own. You can also add an optional template **Description**.
- 6. To add an existing component file to the new template, you can select a component from the **Shared component file** drop-down list. If you do so, the component file you choose is now shared between the new template and any other templates using that component file.
- 7. In the **Initial contents** group, select **Other file**.
- 8. Type in the full path of the fillable PDF document or click **Growse** and browse to it.
- 9. Click **OK** at the **New Template Initial Contents** field. A message box appears, asking if you want to leave fillable fields in the form.
- 10. Decide, based on the following options:
 - Select **Yes, leave fillable fields in the form** if you want to use the fillable fields from the PDF as the basis for fields in the new template. This enables you to create HotDocs fields that match exactly those fields in the underlying PDF.
 - Select **No**, **remove fillable fields from the form** if you do not want the template to use the fillable fields from the source PDF document.
- 11. If the fillable PDF document contains any annotations (such as notes, stamps, highlighting, etc.), a second message box appears, asking if you want to leave annotations in the form.
 - Select **Yes, leave annotations in the form** if you want the assembled Adobe document to show these annotations. Be aware, however, that annotations do not appear on the form in HotDocs Automator.
 - Select **No**, **remove annotations from the form** if you want to remove all annotations so they do not appear in the assembled PDF.

HotDocs creates the template and adds fields to it. If you selected to keep fillable fields in the template, HotDocs has linked these HotDocs fields to the fillable fields. Otherwise, HotDocs creates regular HotDocs fields.

At a Glance: The Check Fillable Fields dialog box

Check Fillable Fields Options		
Select the properties y and fillable fields.	ou want to compare bet	ween HotDocs fields
APosition/Size		
Position	▼ Si <u>z</u> e	
Втуре ———		
Field <u>t</u> ype	✓ Eont	Non-printing
Clayout		
Horizontal	Rotation	Ma <u>x</u> imum lines
Vertical	✓ Borders	🔽 Max <u>c</u> hars/line
Doverflow		<u>@</u>
Shrink answer	Minimum font size	
E Select <u>A</u> ll C	ear All OK	Cancel

After opening **HotDocs Automator**, directly from the your start menu or by opening a form template to edit from a HotDocs Library, you can open the **Check Fillable Fields** dialog box choosing **Check Fillable Fields** from the **Tools** menu.

There are a range of properties you can choose to compare and they are split into four sections. The first section \boxed{A} is Position/Size where you have the option to compare the position and the size of the fillable fields versus the HotDocs check boxes. The second section \boxed{B} is Type and you can compare field type, font and non-printing. The third section \boxed{C} is Layout where you can compare; horizontal, vertical, rotations, borders, maximum lines and maximum characters per line. The last section \boxed{D} is Overflow and you can compare shrink answer and minimum font size.

At the bottom of the dialog there are two buttons \mathbf{E} where you can choose to select all of the options or clear all of the options.

To learn more about fillable fields in PDF templates follow the links below:

- Introduction: Use Fillable PDFs as Form Templates
- Create a Form Template Using a Fillable PDF
- Check Fillable Fields in a PDF Template
- Attach Variables to Fillable Fields
- Tips for Working with Fillable Fields

Check Fillable Fields in a PDF Template

When working with a fillable PDF, you can choose to create HotDocs fields that overlay and link to the PDF's fillable fields. You can then attach variables to these linked fields and answers entered during the interview will be merged in the assembled document.

One benefit of using fillable fields as the basis for HotDocs fields is the fillable fields often include properties that indicate exactly where data must be entered in order for the completed PDF to be filed with a court or agency. For example, you may download a court form that requires answers to appear using a certain font and size, and the answer must be placed in an exact position in the field. When you save the assembled document as a fillable PDF, HotDocs honors the fillable field properties over any HotDocs-specific properties you may assign to fields. This ensures the completed PDF will meet the court's specifications.

Another benefit of using fillable fields is that saving an assembled document as a fillable PDF allows you to edit answers while viewing the document in Adobe Acrobat or Reader.

Finally, you can use fillable fields as a "blueprint" for where HotDocs fields should be placed in the template, especially if you delete or move fields, or if you accidentally edit field properties as you work in the template.

You can assign custom properties to fillable fields while editing the template in HotDocs Automator. However, these properties will only be honored if the assembled document is saved as a regular (or static) PDF or HotDocs PDF document (.HPD).

To help you make sure the linked fields match exactly the fillable fields, you can use the **Check Fillable Fields** tool, which scans through the template and, on a field-by-field basis, allows you to compare the properties of a HotDocs field to its corresponding fillable field. If there are discrepancies, you can adjust the HotDocs field to match the fillable field. Specific properties this tool examines include size, position, field type, font properties, and text alignment within the field, just to name a few.

If no HotDocs field is associated with a fillable field, HotDocs prompts you to create the HotDocs field and link it to the fillable field.

There are two ways you can tell whether a HotDocs field is linked to a fillable field. First, linked fields appear using a light blue color. Second, HotDocs displays the fillable field name at the **Field Properties** dialog box. (To view this name, click the **Additional** tab of the **Field Properties** dialog box. The **Field name** drop-down list in the **Linked Acrobat fillable field** group shows the name.)

You can view where the underlying PDF fillable fields are in the template by choosing **Show Fillable Fields** from the **View** menu.

To check fillable fields in the template

1. Edit a form template that uses fillable fields. (See Edit a Form Template.)

- Choose Check Fillable Fields (Tools menu). The Check Fillable Fields Options dialog box appears.
- Select the field properties you want to compare between the HotDocs fields and the fillable fields. (For example, if you want to compare only the fonts assigned to the fields, select **Font** and clear all other options.)
- 4. Click **OK**. If there are discrepancies between the HotDocs fields and the fillable fields, the **Check Fillable Fields** dialog box appears.

This dialog box shows a table that lists the differences between the two fields it is comparing. The **Field Property** column of the table shows the property name, while the **HotDocs Field** column shows the values currently assigned to the HotDocs field. The **Fillable Field** column shows the properties assigned to the fillable field.

- 5. In the **Change** column, select which HotDocs properties you want to change so they match the fillable field.
- 6. Proceed by choosing one of the following commands:
 - Click **Change** to change the selected properties for the current HotDocs field only. HotDocs makes the change and then displays the next field comparison.
 - Click **Change All** to change the selected properties for all HotDocs fields that have different properties from the fillable fields.
 - Click **Ignore** to ignore the suggested changes for the current field.
 - Click **Ignore All** to ignore all suggested changes for all HotDocs fields that have different properties from the fillable fields.

As you are checking fillable fields in a template, if a fillable field exists, but does not have a HotDocs field linked to it, HotDocs prompts you to create a linked field. To create the current field being suggested, click **Yes**. To create all fields that are missing, click **Yes to All**. To ignore the suggestion to create fields, click either **No** or **No to All**.

If you are showing fillable fields, you can double-click on a gray fillable field and HotDocs will create a linked field. (To view fillable fields, choose **Show Fillable Fields** (**View** menu).)

Attach Variables to Fillable Fields

With HotDocs, you can automate a fillable PDF document. As you attach variables to each linked field in the template, HotDocs attempts to use field names and other properties from the underlying PDF field to create the variable. This includes assigning any formats specified in Adobe to the HotDocs field. For example, if the Adobe document uses a field named *Signature Date* and this field is formatted to appear in the *mm/dd/yyyy* format, HotDocs can attach a Date variable with the same name to the linked field and assign as its format *mm/dd/yyyy*.

As you create these variable fields, you can accept the suggested Adobe defaults, or you can set your own field properties. For best results, accept the suggested field formats when possible. If you save the assembled document as a fillable PDF, Adobe Acrobat or Adobe Reader tries to format the answer to fit the field. If the answer has text or properties Adobe does not expect, the field text may appear incorrectly.

When attaching variables to linked fields in a fillable PDF template, be aware of the following rules:

- To determine what type of variable to create, HotDocs looks at the different formats assigned to the underlying fillable field. If the field has no formats, HotDocs suggests a Text variable. If the field uses formats, HotDocs tries to match the assigned format to its equivalent in HotDocs. For example, if the Adobe field uses a Number format, HotDocs suggests Number as the variable type. If the Adobe field has a telephone number format assigned to it, HotDocs suggests a Text variable with a telephone number pattern.
- HotDocs requires users to follow a set of rules for naming variables. If a name assigned to an Adobe field breaks a rule, HotDocs will change the name to a valid one.
- If an Adobe field is required or has a default answer assigned to it, HotDocs notes this in the **Notes** tab of the **Variable Editor**. (When HotDocs adds a note to the tab, it marks the tab using an asterisk (*) character.)
- If an Adobe field has a Tooltip, HotDocs uses the Tooltip text as a variable prompt.
- Adobe supports time fields that include seconds. It also supports date fields that include hours and minutes. Since HotDocs does not work with either of these types of formats, HotDocs creates a Text variable for the field and includes a prompt to instruct the user on how to enter the answer.
- Calculated fields in a PDF file use a JavaScript expression to compute the actual value. When this field is converted to a HotDocs field, HotDocs places a commented-out copy of the expression in the **Script** field of the **Computation Editor**. You can use this expression as a guide for writing a HotDocs script that calculates the same value.

To attach variables to fillable fields using a wizard

- 1. Edit a form template that uses fillable fields.
- 2. Choose one of the following tasks for attaching a variable to a field:
 - Choose **Create Variables** from the **Tools** menu. HotDocs opens the **Variable Field** dialog box for the first field in the template that does not have a variable attached to it.
 - Select a field and click the ***Variable Field* button. The Variable Field dialog box appears.
- 3. Review the suggested variable type and name, and make any necessary changes.
- 4. Click the **dit Component** button. The **Variable Editor** appears.
- 5. Review the properties assigned to the variable, including its format, prompt, and any other patterns, limits, or other options.
- 6. Optionally, if the **Notes** tab shows an asterisk (*) in its label, click it to view any additional fillable field requirements.
- 7. Click OK to close the Variable Editor,

- 8. Click **OK** to close the **Variable Field** dialog box. The variable is created and attached to the field, and HotDocs moves to the next empty field on the form.
- 9. Repeat this process for each field in the template.

To skip a field and not create a variable for it, click the **Skip** button on the **Variable Field** dialog box.

Tips for Working with Fillable Fields

As you work with fillable PDFs, please note the following information:

Creating Fillable PDF Form Templates

HotDocs supports only fillable PDFs (or Acrobat Forms or AcroForms) created with Acrobat Professional. It does not support any fillable PDF documents created or edited using Adobe LiveCycle Designer. (This includes XFA forms, XML Forms, or Designer Forms.)

A small subset of XFA forms might still work, as long as there are no dynamic form features being used. Typically, if an AcroForm has been edited using LiveCycle Designer and only properties of fields have been changed, it is likely to still work in HotDocs.

Using Fonts

If you have Type 1 (PostScript) fonts installed on your computer, you should not assign those fonts to fields in PDF-based template files. If you assign a Type 1 font to a form field, the field will appear correctly on screen during assembly and when the form is saved as a fillable PDF. However, if you print the form or save it as a static PDF, HotDocs will automatically substitute a TrueType font, which will keep the text in the field from appearing as expected. For best results, use TrueType (.ttf) fonts (or OpenType (.otf) fonts containing TrueType outlines) with linked fields in a fillable PDF template.

Fillable PDF authors may embed fonts they use in the fillable PDF. However, when you create a template using a fillable PDF, HotDocs will not use the embedded font. Instead, it will check to see if the font is installed locally. If it is, HotDocs will use it. If the font is not installed on the system, HotDocs will attempt to map the font to something similar. Once you save the assembled document as a fillable PDF, however, the embedded font will once again be used.

Honoring Adobe Field Formats

When you create variables and attach them to linked fields in a PDF-based template, HotDocs may suggest some default properties for the variable, such as a format, pattern, and so forth. Where possible, you should accept these default properties. If you save the assembled document as a fillable PDF, Adobe Acrobat or Adobe Reader will try to format the answer coming from HotDocs so that it fits in the field as

its defined by Adobe Acrobat. If the answer contains text or uses formats that are different than what it expects, the field text may appear incorrectly. This is especially true with Number and Date formats.

Grouping Fields

If you are automating a fillable PDF template and you plan to save the assembled document as a fillable PDF, you cannot group run-on fields since necessarily, each fillable field is linked to a separate "variable." Answers—both in HotDocs and Adobe—must be entered on individual lines. This means that as you enter an answer in a field, when you reach the end of one field, you must tab to the next field and continue typing your answer.

This inability to group fields is a limitation of Adobe. If you have write-access to the form using Adobe Acrobat, you may consider replacing individual fields in the group with a single, multiline field with a suitable font size. Text you enter may not match the lines perfectly, but you will not have to tab between fields to enter the answer.

Check Box Groups

Multiple check boxes or option buttons with the same field name in a PDF file are converted to a HotDocs check box group. HotDocs relies on the sort order of the member fields when answers from HotDocs are saved back to the PDF fields. This means that it is not a good idea to change the ordering of the member fields using row and column settings.

Additionally, in the fillable PDF, check boxes and option buttons should all have different **Export values** since HotDocs doesn't allow duplicate options in a Multiple Choice variable.

True/False Check Box Groups

If you need to group two check boxes that have different Acrobat field names there is a simple method:

- 1. Select the check box for true and open the Field Properties dialog box.
- 2. In the **Variable** text field enter the name of the True/False variable you are using and in the **Check Box Character** text field type **X**.
- 3. Select the check box for false and open the Field Properties dialog box.
- 4. In the **Variable** text field enter the name of the True/False variable you are using followed by **/X** then in the **Check Box Character** text field type **X**.
- 5. While the variable attached the first check box will cause the **X** to appear in the box when answer is true, the **/X** after the variable name in the second box causes the **X** to appear when the answer is false.

For more information on working with check boxes see Create a check box Field and Attach a Multiple Choice Variable to a Group of check boxes.

Working with Rights-Enabled PDFs

When creating a fillable PDF document, form designers can enable usage rights within the document. These rights allow Adobe Reader users to save copies of the document with changes they have made to the document—including entering or changing answers they have entered in the fillable PDF. However, when you convert a rights-enabled fillable PDF to template format, the document is changed in such a way that usage rights are no longer in effect. This means that users will not be able to save changes they make when editing the assembled document in Adobe Reader.

Understanding Variables

Using Variable Types

Introduction: Create and Customize Variables

Once your document text is in a template file, you must replace variable text (text that changes each time you assemble a document) with HotDocs variables. Examples of variable text include names, dates, pronouns, numbers, and calculations. A variable is what prompts HotDocs to ask for that information when the template is used to assemble a document.

You can create seven types of variables: Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information. The variable type determines what information users can enter when they complete an interview. For example, if you create a Date variable, users must enter a valid date—they cannot enter text, such as a name, for an answer.

The following graphic shows a paragraph that contains two items that change depending

on the user—the name (Aaron Jameson) and the date (12th day of October, 2002):

EMPLOYMENT AGREEMENT

This Employment Agreement, by and between Hobble Creek Publishing and Aaron Jameson, is entered into this 12th day of October, 2002.

You replace both of these items with variables:

EMPLOYMENT AGREEMENT

This Employment Agreement, by and between Hobble Creek Publishing and «Employee Name», is entered into this «Agreement Date:3rd day of June, 1990».

There are two parts to creating variables—creating the variable component, and creating the variable field:

• Variable Component: The variable component is the core part of a HotDocs variable. It contains all the information about a variable and how it should be processed, including the component name and prompt, any resource information, and any special patterns or other options that help control how the variable appears during the interview and how it will be processed when the final document is assembled. A variable component also creates an association between the components and the answers the user enters, so that the answer can be saved in an answer file. (Answer files can be reused when assembling other documents.)

• Variable Field: When you insert a variable into a HotDocs template, HotDocs creates a field where the user's answer is merged into the assembled document. In a text template, the field is denoted by HotDocs chevrons (« »), while in a form template, the field is denoted by a colored box that is overlaid on the form's static text. There are certain properties that are assigned to a variable's field that affect the way the variable is merged into the document—specifically, formats and merge text for Multiple Choice variables. You can also include fill characters that force the answer to be a certain number of characters, regardless of how long the answer is.

If you want to include a chevron in a HotDocs text template either as a part of your document text or a literal text string, you need to double the opening chevron (but *not* the closing chevron), so that HotDocs does not interpret the chevrons (and any text between them) as a field. The output is a single chevron, as follows: input: «some text», output: «some text».

When you insert a variable into the template, you automatically create a merge field for the variable. However, when you create a variable using Component Manager, there is not a merge field associated with the variable until the variable is inserted into the template.

At a Glance: The Text Variable Editor

A New Text Variable - Text Variable Editor - Demo Employment Agreem (? <mark>×</mark>				
Properties Advanced Resource Used In Notes					
Variable <u>n</u> ame: Default <u>f</u> ormat: No	n- <u>b</u> reaking T				
Title:					
G					
Prompt:					
D	*				
Ē=ield <u>h</u> eight: 1 🚔 Ma <u>x</u> imum characters: 0 🚔 Patt <u>e</u> rn:	•				
Enter key in multi-line answers inserts new paragraph mark (¶)					
Test Update OK Cancel	Sa <u>v</u> e				

You can open the **Text Variable Editor** by creating a new text variable or by editing an existing variable, either while editing a template or from the **Component Manager.**
In the first text field A you can enter a name for your variable. To the right of this is a drop-down list B where you can choose the default format from a list of choices or type your own. You also have the option to check a box so that spaces used in the variable's output text will be non-breaking, ensuring it won't be split over two lines of text.

Below this you can enter a variable title C and a prompt D that will both appear on the interview dialog next to the variable answer field.

You can set the field height in the interview by entering the amount of lines (up to twelve) in the number field field , and the maximum amount of characters by entering a number (up to 15,000) in the number field to the right (the default value - zero - is the same as the maximum of 15,000). To the right of the **Maximum character** setting is the **Pattern** field where you can enter your own text pattern or choose from the available patterns in the drop-down list. This option is only available when the field height is set to 1.

Below these options is a check box f that you can tick to have HotDocs insert a new paragraph mark every time the user clicks enter in this answer field in the interview. This option is only available when the field height greater than 1.

More dialog options are available in the other tabs: Advanced, Resource, Used In and Notes.

To learn more about editing text variables follow the links below:

- Introduction: Create and Customize Variable
- Customize a Text Variable
- Tips on Naming Your Variables

Customize a Text Variable

Text variables are used to merge text, such as names or descriptions, into assembled documents. They also merge numbers that are never added, subtracted, and so on, such as telephone numbers, U.S. Social Security numbers, and times of day.

When you create a Text variable, you can add a title or prompt to help the user know what information is required. You can also decide how answers will be formatted when they are merged into the assembled document. Finally, you can add a resource or select advanced options that determine how the variable will be processed and merged into the document.

Once a variable is created, you can edit it at any time. See Edit a Variable.

To edit the properties of a Text variable

- 1. Follow the instructions for inserting a variable, selecting **Text** as the variable type. (See Insert a Variable Field in a Text Template.)
- 2. At the Variable Field dialog box, click the **Edit Component** button. The **Text Variable Editor** appears.
- 3. Make any changes, as described in the following table:

То	Do This
Specify a default format that will be applied to all instances of the variable (except when a field-specific format has been assigned)	Select a format from the Default format drop- down list. (See Format the Variable and How Example Formats are Interpreted.)
	Optionally, to keep the answer from breaking across lines in the document, select Non- breaking .
Specify an alternate name for the variable	Enter a title in the Title field.
	See Understand How Component Titles and Prompts are Used for a description of how and when variable titles are used.
Provide users with additional information about the variable	Enter a prompt in the Prompt field. This information replaces the Variable name and Variable title when the variable is presented during the interview. (See Create a Prompt for a Variable.)
Change the height of the answer field that appears in the interview, allowing it to show more than a single line	Either type the number of answer field lines (up to 12) in the Field height field, or click the up or down arrows to select a number.
	Changing the answer field height affects only how large the answer field appears in the interview—not how long the answer can be. To limit the answer length, enter a number in the Maximum characters field.
Control how an answer is merged in a Word document when the user presses Enter in a multi-line answer field.	Select Enter key in multi-line answers inserts new paragraph mark (1) .
The type of answer required determines which break should be used. For example, if the user must enter separate paragraphs of text, HotDocs should merge a paragraph	To control whether line or paragraph breaks are inserted in literal text strings used in computation scripts, see Use Line Breaks,

break. However, if the user must enter separate lines in a single paragraph (such as lines in an address), HotDocs should insert a line break.	Paragraph Ends, and Tabs in Computation Scripts.
Control the number of characters allowed in the answer	Either type a value in the Maximum characters field, or click the up or down arrows to select a value (up to 15,000 characters).
Control what type of information a user types and format it as they type	Select a pattern from the Pattern drop-down list. (HotDocs provides a telephone number pattern, time of day patterns, and a Social Security number pattern. You can also create your own. See Use Pattern Codes to Create a Custom Pattern.)
Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document	Click the Advanced tab and select any of the options. (See Control How HotDocs Processes a Variable, Specify the Width of Answer Fields in the Interview, and Control How HotDocs Processes a Variable, respectively.)
Provide users with helpful information that can assist them in providing the correct answer	Click the Resource tab and provide a resource option. (See Add Resource Information to a Variable or Dialog.)
View a list of all components that refer to this variable	Click the Used In tab. (See View Relationship Between the Current Component and Other Components.)
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments. (See Add Notes to Components for details.)
Save changes to the variable without closing the variable editor	Click Save.

At a Glance: The Number Variable Editor

2 New Nu	imber Variable	- Number Variable Ed	ditor - Demo Employ	ment 💡 💌
Properties	Advanced Re	source Used In Not	tes	
Variable <u>n</u> a	ime:		Default <u>f</u> orma	t: Non- <u>b</u> reaking
	A			B -
Tit <u>l</u> e:				
		C		
Prompt:				
		D		*
<u>()</u> inimum:		Ma <u>x</u> imum:		
B Decimal pla	ices:	Currency symbol:	•	
	Update	2	OK Cancel	Sa <u>v</u> e

You can open the **Number Variable Editor** by creating a new number variable or by editing an existing variable, either while editing a template or from the Component Manager.

In the first text field A you can enter a name for your variable. To the right of this is a drop-down list B where you can choose the default format from a list of choices or type your own. You also have the option to check a box so that spaces used in the variable's output text will be non-breaking, ensuring it won't be split over two lines of text.

Below this you can enter a variable title C and a prompt D that will both appear on the interview dialog next to the variable answer field.

Below the **Prompt** field are two number fields \mathbf{E} where you can set a minimum and maximum amount for the answer given in the interview. If the user's answer is too large or too small they will see a warning message and have to enter another amount.

Under these fields is a number field \mathbf{F} where you can set the amount of decimal places that will be shown in the dialog and in the assembled document. To the right of this is a drop-down list where you can choose to have a currency symbol appear automatically next to the answer field in the interview.

More dialog options are available in the other tabs: Advanced, Resource, Used In and Notes.

To learn more about editing number variables follow the links below:

- Introduction: Create and Customize Variable
- Customize a Number Variable

• Tips on Naming Your Variables

Customize a Number Variable

You can merge numbers into your assembled document using Number variables. You can format numbers as currency, use decimals, and specify minimum and maximum values.

Typically you use a Number variable to represent text in your template that can be calculated—for example, dollar amounts or other sums. Even though Social Security numbers, telephone numbers, and times are numbers, you must use a Text variable with the appropriate pattern to merge one of these numbers into your document. (See Customize a Text Variable.)

When you create a Number variable, you can add a title or prompt to help the user know what information is required. You can also decide how answers will be formatted when they are merged into the assembled document. Finally, you can add a resource or select advanced options that determine how the variable will be processed and merged into the document.

Once a variable is created, you can edit it at any time. (See Edit a Variable.)

To edit the properties of a Number variable

- 1. Follow the instructions for inserting a variable, selecting Number as the variable type. (See Insert a Variable Field in a Text Template.)
- 2. At the Variable Field dialog box, click the **Edit Component** button. The Number Variable Editor appears.
- 3. Make any changes, as described in the following table:

То	Do This
Specify a default format that will be applied to all instances of the variable (except when a field-specific format has been assigned)	Select a format from the Default format drop- down list. (See Format the Variable and How Example Formats are Interpreted.)
	Optionally, to keep the answer from breaking across lines in the document, select Non-breaking .
Specify an alternate name for the variable	Enter a title in the Title field.

	See Understand How Component Titles and Prompts are Used for a description of how and when variable titles are used.
Provide users with additional information about the variable	Enter a prompt in the Prompt field. This information replaces the Variable name and Variable title when the variable is presented during the interview. (See Create a Prompt for a Variable.)
Force users to enter a number that falls within a certain numeric range	Type numbers in the Minimum and Maximum fields.
Specify a certain number of decimal places allowed in the users' answer	Either type a number (0-7) in the Decimal places field, or click the up or down arrows to select a number.
Make HotDocs show a currency symbol next to the answer field when it displays the variable in the interview	Select a symbol from the Currency drop-down list.
Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document	Click the Advanced tab and select any of the options. (See Control How HotDocs Processes a Variable, Specify the Width of Answer Fields in the Interview, and Control How HotDocs Processes a Variable, respectively.)
Provide users with helpful information that can assist them in providing the correct answer	Click the Resource tab and provide a resource option. (See Add Resource Information to a Variable or Dialog.)
View a list of all components that refer to this variable	Click the Used In tab. (See View Relationship Between the Current Component and Other Components.)
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments. (See Add Notes to Components for details.)
Save changes to the variable without closing	Click Save .

If you are creating a list of answers, you can use a built-in Number variable called COUNTER to count the number of entries in a list. See Count the Number of Entries in a List.

You can automatically number paragraphs or other blocks of text. See Use Automatic Paragraph Numbering.

At a Glance: The Date Variable Editor

🛐 New Date Variable - Date Variable Editor - Demo B	mployment Agreem 💡 💌
Properties Advanced Resource Used In Notes	
Variable <u>n</u> ame:	Default format: 🔲 Non-breaking
	B 👻
Title:	
G	
Prompt:	
	A
	Ψ
Test Update OK	Cancel Sa <u>v</u> e

You can open the **Date Variable Editor** by creating a new date variable or by editing an existing variable, either while editing a template or from the **Component Manager**.

In the first text field A you can enter a name for your variable. To the right of this is a drop-down list B where you can choose the default format from a list of choices or type your own. You also have the option to check a box so that spaces used in the variable's output text will be non-breaking, ensuring it won't be split over two lines of text.

Below this you can enter a variable title C and a prompt D that will both appear on the interview dialog next to the variable answer field.

More dialog options are available in the other tabs: Advanced, Resource, Used In and Notes.

To learn more about editing date variables follow the links below:

- Introduction: Create and Customize Variable
- Customize a Date Variable
- Tips on Naming Your Variables

Customize a Date Variable

You can create variables that merge dates into your assembled document. When you create a Date variable, you can add a title or prompt to help the user know what information is required. You can also decide how answers will be formatted when they are merged into the assembled document. Finally, you can add a resource or select advanced options that determine how the variable will be processed and merged into the document.

Once a variable is created, you can edit it at any time. See Edit a Variable.

To edit the properties of a Date variable

- 1. Follow the instructions for inserting a variable, selecting **Date** as the variable type. (See Insert a Variable Field in a Text Template.)
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- 3. Make any changes, as described in the following table:

То	Do This
Specify a default format that will be applied to all instances of the variable (except when a field-specific format has been assigned)	Select a format from the Default format drop- down list. (See Format the Variable and How Example Formats are Interpreted.)
	Optionally, to keep the answer from breaking across lines in the document, select Non- breaking.
Specify an alternate name for the variable	Enter a title in the Title field.
	See Understand How Component Titles and Prompts are Used for a description of how and when variable titles are used.
Provide users with additional information about the variable	Enter a prompt in the Prompt field. This information replaces the Variable name and Variable title when the variable is presented during the interview. (See Create a Prompt for a Variable.)
Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document	Click the Advanced tab and select any of the options. (See Control How HotDocs Processes a Variable, and Specify the Width of Answer Fields in the Interview.)

Provide users with helpful information that can assist them in providing the correct answer	Click the Resource tab and provide a resource option. (See Add Resource Information to a Variable or Dialog.)
View a list of all components that refer to this variable	Click the Used In tab. (See View Relationship Between the Current Component and Other Components.)
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments. (See Add Notes to Components for details.)
Save changes to the variable without closing the variable editor	Click Save.

At a Glance: The True/False Variable Editor

Fo New True/False Variable - True/False Varial	ble Editor - Demo Employm 👔 💌
Properties Advanced Resource Used In No	tes
Variable <u>n</u> ame:	Default <u>f</u> ormat: Non- <u>b</u> reaking
Tit <u>l</u> e:	
Prompt:	
D	*
Yes/No on <u>s</u> ame line	
Test Update	OK Cancel Sa <u>v</u> e

You can open the **True/False Variable Editor** by creating a new true/false variable or by editing an existing variable, either while editing a template or from the Component Manager.

In the first text field A you can enter a name for your variable. To the right of this is a drop-down list B where you can choose the default format from a list of choices or type your own. You also have the option

to check a box so that spaces used in the variable's output text will be non-breaking, ensuring it won't be split over two lines of text.

Below this you can enter a variable title **G** and a prompt **D** that will both appear on the interview dialog next to the variable answer field.

Under the prompt field a check box government of the same line instead of vertically aligned.

More dialog options are available in the other tabs: Advanced, Resource, Used In and Notes.

To learn more about editing true/false variables follow the links below:

- Introduction: Create and Customize Variable
- Customize a True/False Variable
- Tips on Naming Your Variables

Customize a True/False Variable

Using a True/False variable, you can merge a word or phrase into an assembled document based on the user's answer to a yes/no question.

Likewise, if you are developing a form template using HotDocs Automator, you can use True/False variables to select and clear check boxes.

When you create a True/False variable, you can add a title or prompt to help the user know what information is required. You can also decide how answers will be formatted when they are merged into the assembled document. Finally, you can add a resource or select advanced options that determine how the variable will be processed and merged into the document.

Once a variable is created, you can edit it at any time. See Edit a Variable.

To edit the properties of a True/False variable

- 1. Follow the instructions for inserting a variable, selecting **True/False** as the variable type. (See Insert a Variable Field in a Text Template.)
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- 3. Make any changes, as described in the following table:

То

Do This

Specify a default format that will be applied to all instances of the variable (except when a field-specific format	Select a format from the Default format drop-down list.
has been assigned)	You can customize this format by entering other <i>true/false</i> text to the left and right of the slash. If the user chooses <i>yes</i> , the text entered to the left of the slash merges. If the user chooses <i>no</i> , the text entered to the right of the slash merges. If you do not enter a format, either <i>true</i> or <i>false</i> is merged. (See Format the Variable.)
	Optionally, to keep the answer from breaking across lines in the document, select Non-breaking .
	The total number of characters you can use in a merge format is 1,000, but only the first 50 characters appear in the template.
Specify an alternate name for the variable	Enter a title in the Title field.
	See Understand How Component Titles and Prompts are Used for a description of how and when variable titles are used.
Provide users with additional information about the variable	Enter a prompt in the Prompt field. This information replaces the Variable name and Variable title when the variable is presented during the interview. (See Create a Prompt for a Variable.)
Have Yes and No appear on the same line in the interview	Select Yes/No on same line.
Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document	Click the Advanced tab and select any of the options. (See Control How HotDocs Processes a Variable, and Specify the Width of Answer Fields in the Interview.)
Provide users with helpful information that can assist them in providing the correct answer	Click the Resource tab and provide a resource option. (See Add Resource Information to a Variable or Dialog.)
View a list of all components that refer to this variable	Click the Used In tab. (See View Relationship Between the Current Component and Other Components.)
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments. (See Add Notes to Components for details.)

Save changes to the variable without closing the variable editor

Click Save.

If you have multiple True/False variables in a dialog, you can group them so that each variable is preceded with a check box or option button. This allows users to select either one option or multiple options from a list of possible answers. See Change a Dialog's Options.

You can use True/False variables to determine whether sections of text should be included in a document. See Include or Exclude a Single Version of Text Using a Simple IF Instruction or Expression.

At a Glance: The Multiple Choice Variable Editor

Ĵ≣ New Multiple Choice	Variable - Multip	le Choice Variab	ole Editor - Dem 🔋	×
Properties Advanced O	ptions Resource	Used In Note	s	
Variable <u>n</u> ame:		De	fault <u>f</u> ormat: 🔲 Non- <u>b</u> re	aking
			в	•
Title:				
		C		
Prompt:				
		D		*
		·		*
Opt <u>i</u> on 👻	Prompt		Default Merge Text	-
1		_		
2		0		
4				··· +
	1		•	
In <u>s</u> ert <u>D</u> elete				
S <u>e</u> lect: One Only	•	Othe <u>r</u>		
St <u>v</u> le: Buttons	G 🗸			
Test Upda	te	<u>o</u> k (Cancel Sa <u>v</u> e	

To open the **Multiple Choice Variable Editor**, create a new multiple choice variable or edit an existing variable, either while editing a template or from the **Component Manager**.

In the first text field A you can enter a name for your variable. To the right of this text field is a drop-down list where you can choose the default format from a list of choices or type your own. You also have the option to check a box so that spaces used in the variable's output text will be non-breaking, ensuring it won't be split over two lines of text.

Below this you can enter a variable title **C** and a prompt **D** that both appear on the interview dialog next to the variable answer field.

Underneath these options is a table where you can enter the options for the user to choose, prompt text for each option, and the default merge text for each option. The two buttons at the bottom of the table insert or delete rows.

Next there are two drop-down menus. The first menu choice or select all that apply. You can also select the check box to the right to allow users to choose **Other** and enter their own option. The second menu choice of lets you choose the style of the multiple choice question in the interview:

- Buttons
- Button Grid
- List
- Drop-down List

More dialog options are available in the other tabs: Advanced, Resource, Used In and Notes.

Customize a Multiple Choice Variable

Multiple Choice variables let users select one or more answers from a list, and the selected option or its corresponding merge text is merged into the assembled document. Merge text can be specified either at a variable field or at the variable editor.

When you create a Multiple Choice variable, you can add a title or prompt to help the user know what information is required. You can also decide how answers are formatted when merged into the assembled document. Finally, you can add a resource or select advanced options that determine how the variable is processed and merged into the document.

Once a variable is created, you can edit it at any time.

To edit the properties of a Multiple Choice variable

- 1. Insert a variable and select **Multiple Choice** as the variable type.
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- 3. In the **Option** column, enter at least two options for users to choose from. (Do not create an option with the exact text *None of the Above* or *Other*. Instead, select **None of the Above** or **Other** from the **Select** option.)
- 4. Optionally, type a prompt for the option in the **Prompt** column. The prompt (instead of the corresponding **Option** text) appears for the options during the interview. (If you need more space for the prompt, click the **Options** tab and type the text there.)
- Optionally, type the text you want merged into the assembled document in the Default Merge Text column. (If you need more space for the merge text, click the Options tab and type the text there.)
- 6. From the Select drop-down list, select either One Only or All That Apply.
- 7. Depending on your selection in Step 6, select **Other** or **None of the Above**. (These options allow users to either type an answer not in the list or specify that the answer needed is not available in the list.)
- 8. Choose a display style from the **Style** drop-down list. The available styles depend on the selection style.
- 9. Make any of these changes:

То	Do This
Add longer prompts and merge text for each option, as well as suggest an answer for the user	Click the Options tab and make your changes.
Specify a default format that will be applied to all instances of the variable (except when a field-specific format has been assigned)	Select a format from the Default format drop- down list. Optionally, to keep the answer from breaking across lines in the document, select Non- breaking .
Specify an alternate name for the variable	Enter a title in the Title field.

Provide users with additional information about the variable	Enter a prompt in the Prompt field. This information replaces the Variable name and Variable title when the variable is presented during the interview.
Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document	Click the Advanced tab and select any of the options. (See Control How HotDocs Processes a Variable, and Specify the Width of Answer Fields in the Interview.)
Provide users with helpful information that can assist them in providing the correct answer	Click the Resource tab and provide a resource option.
View a list of all components that refer to this variable	Click the Used In tab. (See View Relationship Between the Current Component and Other Components.)
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments.
Save changes to the variable without closing the variable editor	Click Save.

You can **control whether merge text** is a default property of the variable component, or whether it's a field-specific property of a variable field.

Add and delete rows of options by clicking **Insert** or **Delete**.

You can use existing option or merge text lists when you create other Multiple Choice variables in the template. To do this, click the **Option** and **Merge Text** column headings and choose the group.

You can copy and paste options either to or from the Multiple Choice Variable Editor.

To SET two or more options for a Multiple Choice variable in a computation script, separate each option with a vertical bar (for example, SET MC Variable TO "Option1|Option2|Option3"). (The Multiple Choice variable must have the Select All That Apply property selected.)

Multiple Choice variables to insert gender references, such as pronouns, are common in templates. HotDocs automatically includes common pronoun sets in the **Merge Text** list. Click the **Merge Text** column heading and select an option.

You can customize Multiple Choice option prompts and merge text based on answers the user enters in the interview.

At a Glance: The Computation Editor



You can open the **Computation Variable Editor** by creating a new computation variable or by editing an existing variable, either while editing a template or from the **Component Manager**.

In the first text field A you can enter a name for your variable. To the right of this is a drop-down list B where you can choose the default format from a list of choices or type your own. You also have the option

to check a box so that spaces used in the variable's output text will be non-breaking, ensuring it won't be split over two lines of text.

Below this is the **Script Toolbar** where you can choose from the following buttons to search and edit the **Script** field **C**:

- **Auto Format:** Indents matching pairs of IF and REPEAT instructions, based on the level of their insertion.
- **Undo:** Removes any changes you have made to the script.
- **Redo:** Reapplies any changes you have made to the script.
- **Cut:** Removes the selected text and copies it to the Clipboard.
- **Copy:** Copies the selected text to the Clipboard.
- **Paste:** Pastes the selected text on the Clipboard at the cursor position in the script.
- **Export Computation:** Copies the entire computation variable to clipboard (See Import or Export Computation Variables for more details)
- **# Find:** Displays the Find dialog box where you can specify the word or text string for which you are searching.
- A Find Next: Finds the next instance of the word or text string for which you are searching.
- text string and replace: Searches for a specific word or text string and replaces it with word or text string you specify.
- **Go To:** Displays the Go To dialog box where you can specify the location in the script to which you want to move your cursor. You can go to either a specific line or character in the script.
- **Indent Block:** Indents the selected text.
- **E**Outdent Block: Realigns the indented text with the left margin of the scripting box. (If the text has been indented more than once, realigns the text with the previous tab stop.)
- **Comment Block:** Causes the selected text to become inactive, meaning HotDocs won't process it when it processes the computation. Often, template developers "comment out" scripting if they need to close the scripting dialog box but HotDocs won't let them because the script is invalid. Additionally, developers often add personal explanations to the scripts they are writing, and commenting them out keeps the computation operational.
- **Uncomment Block:** Removes any commenting that has been applied to the selected script. Once the script is uncommented, HotDocs will attempt to process it.
- **Options:** Displays the HotDocs Options dialog box where you can customize the way script editing features work.
- **Weighted Help:** Opens the relevant page of the HotDocs Help File.

You have the option of writing directly in the **Script** field, using your mouse to drag variables, operators, and models from the lists below to this box, or you can press Ctrl+Space to have HotDocs display these lists directly in the script so you can use the keyboard to choose the syntax item you need.

On the left is the Component list \bigcirc showing a list of all the components in this template. You can search it using the find field at the bottom, filter it by selecting a component type from the drop-down list at the top or create a new component by clicking on the **New Component** button. In the centre \sqsubseteq is a list of useful scripting operators, at the top right is a list of Instruction models \boxed{E} , and at the bottom right is a list of Expression models \boxed{E} .

To access help for each instruction or expression, first select the model in one of the lists and then press **Ctrl+F1**.

For further information on how to use the operators, see Use Operators when Scripting, and for further information what the Instruction and expression models do, see Full List of Instruction Models and Full List of Expression Models.

More dialog options are available in the other tabs: Advanced, Resource, Used In and Notes.

To learn more about editing computation variables follow the links below:

- Introduction: Create and Customize Variable
- Customize a Computation Variable
- Use the Script Editor
- At a Glance: The Locals tab (Variable/Dialog Editor)
- At a Glance: The Advanced tab (Variable Editor)
- At a Glance: The Resource tab (Variable/Dialog Editor)
- At a Glance: The Used In tab (Variable/Dialog Editor)
- At a Glance: The Notes tab (Variable/Dialog Editor)
- Tips on Naming Your Variables

Customize a Computation Variable

You can use a Computation variable to calculate number, date, true/false, and text values based on answers a user enters. The computed value can then be merged into the assembled document. Computation variables can also be used to group several instructions or expressions for insertion into a template, which results in quicker assembly.

Creating a computation requires you to write a script using the HotDocs scripting language. The actual process of writing a script depends on your skill level in using this language. If you're a beginner, you can drag scripting models into the **Script** field inside the script editor and allow HotDocs to guide you in filling in the required information. With experience, you can type the script directly in the field. Some developers may wish to create Local Variables and Parameters that are specific to the computation.

The title bar of the Computation Editor contains information that can help you evaluate the correctness of the script as you write it. When a script is invalid (meaning it does not adhere to the rules of the HotDocs scripting language), HotDocs includes the word **[Invalid]** in the title bar. Once a script is valid, HotDocs then notes what type of result the script produces (for example, **[Text]** or **[Number]**).

Once a variable is created, you can edit the variable at any time.

To edit the properties of a Computation variable

- 1. Insert a variable, selecting **Computation** as the variable type.
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- 3. In the **Script** field, enter a computation script.
- 4. Make any changes, as described in the following table:

То	Do This
Indent matching pairs of IF and REPEAT instructions	Click the Auto Format button. HotDocs indents the instructions based on the level of their insertion.
Specify a default format that will be applied to all instances of the variable (except when a field-specific format has been assigned)	Select a format from the Default format drop- down list.
	Optionally, to keep the answer from breaking across lines in the document, select Non-breaking .
Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document	Click the Advanced tab and select any of the options. (See Control How HotDocs Processes a Variable, and Specify the Width of Answer Fields in the Interview.)
Add a title to the Computation variable	Click the Advanced tab and enter the title in the Title field.
Provide users with helpful information that can assist them in providing the correct answer	Click the Resource tab and provide a resource option.
View a list of all components that refer to this variable	Click the Used In tab.
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments.

Save changes to the variable without closing Click Save. the variable editor

You can create new variables directly at the **Computation Editor**. Make sure the **Components** list contains the type of variable you want to create, and then click the **New Component** button. Similarly, you can edit an existing variable in the list by double-clicking it.

If you experience **errors** when writing a script, it may be because you are using the instruction and expression keywords incorrectly. If you aren't receiving errors, but the script produces results you don't expect, you can **examine the script line by line** to see where the logic is flawed.

HotDocs will not let you close the Computation Editor or save a script if the script is invalid. If you need to close or save, you must either comment out the script (by highlighting the script and clicking the **Comment Block** button) or place a **QUIT** instruction at the beginning of the script (see **QUIT**).

Insert a Personal Information Variable

Personal Information variables store basic information about the user, such as a name, a company name, and a phone number in the Current User key of the Windows System Registry. Those answers can then be used in any document that uses the same Personal Information variables, regardless of which answer file is being used. HotDocs provides 12 Personal Information variables, but you can also create your own.

To insert a Personal Information variable

- 1. At the template, place your cursor where you want the Personal Information variable and click the ****Variable Field** button. The **Variable Field** dialog box appears.
- 2. Select Personal Information.
- 3. Either type a name for a new Personal Information variable in the **Variable** field, or click the **Variable** drop-down button to select an existing Personal Information variable.
- 4. Optionally, clear **Use default** and select an example format from the **Format** drop-down list. This format will be applied to this instance of the variable only. (See Format the Variable.)
- 5. Optionally, click **Show Advanced** to assign options that control how that specific instance of the variable is merged into the assembled document. See Control How Answers Appear in the Assembled Document for information about each option.

When a Personal Information variable is inserted in the template, its name is preceded by the word *MY*. This notation distinguishes it from other variables in the template.

When users assemble documents that use Personal Information variables, HotDocs asks for the information and then stores it in the HotDocs section of the system registry. Once entered, the variable is never asked again; however, users can change their information at the **HotDocs Options** dialog box. See Edit Answers for Built-In Personal Information Variables.

Create a Date Variable that Inserts the Current Date

You can insert the current date in a template using the built-in variable, TODAY. The variable won't be asked during the interview—instead TODAY automatically merges the current date, based on the Windows System Clock.

To insert the TODAY variable

- 1. Follow the instructions for inserting a variable, selecting **Date** as the variable type. (See Insert a Variable Field in a Text Template.)
- 2. At the Variable Field dialog box, click the Variable drop-down button and select TODAY.
- 3. Optionally, clear **Use default** and select a format from the **Format** drop-down list. This format will be applied to this instance of the variable only. (See Format the Variable.)

Merge the Name of the Answer File in the Document

In some documents, you may need to include the name of the answer file that was used to assemble the document. You can insert an instruction that merges the path and file name of the answer file directly in the document. If the answer file is new and unsaved, the text, *New Answer File*, is merged into the assembled document.

To create an answer file variable

- 1. Open a template for editing. (See Edit a Template.)
- 2. Position the cursor in the template where you want to merge the name of the answer file.
- 3. Click the Variable Field dialog box and choose Text as the Variable type.
- 4. Click the Variable drop-down button and choose ANSWER FILE NAME.

Use Automatic Paragraph Numbering

You can create Number variables that supply automatic paragraph numbering to lists of answers in your HotDocs text template. This feature is not available in JavaScript or Silverlight Interviews.

HotDocs paragraph numbering values are merged into an assembled document as text, not codes. This means that if users add or delete a paragraph after the document has been assembled and sent to the word processor, the numbering will not adjust. To have automatic paragraph numbering that adjusts to changes made in the assembled document, use your word processor's numbering codes.

To create automatic paragraph numbering

- 1. Position your cursor at the place in the template where you want the paragraph numbering to take effect.
- 2. Follow the instructions for inserting a variable, selecting **Number** as the variable type. (See Insert a Variable Field in a Text Template.)
- 3. With the **Variable Field** dialog box displayed, type **PN#** in the **Variable** field. (Replace # with a number representing the paragraph level. Numbers 1–9 can be entered.)
- 4. Optionally, clear **Use default** and select one of the following formats: **9.9**, **9.9.a**, **abc**, or **IX**. (If no example format is selected, the default—9—is used.)
- 5. Insert this Number variable before every paragraph of that level.

You can use SET instructions to customize HotDocs paragraph numbering. For example, the computation script, SET PN1 TO 10, would automatically start the paragraph numbering at 10. Additionally, if you assign the **abc** example format to the PN1 variable, HotDocs will merge in the alphabetic character that corresponds with the number used in the SET instruction. For example, the computation script, SET PN1 TO 10, would start numbering the variable **«PN1:ABC»** with the letter J.

Editing Variables

Edit a Variable

You can edit a variable, including changing a variable's properties. Changes to the component properties of a variable affect all instances of the variable in the template.

To edit a variable's field-specific properties, edit them at the **Variable Field** dialog box. To do this, place your cursor in the variable field (at the template) and click the ****Variable Field** button. Make sure **Use default** is cleared.

You can edit a variable directly at the template, or you can edit it using Component Manager.

To edit a variable at the template

- In the text template, click in the variable field and click the difference difference
- 2. Make any changes. (Click any of the following links for information on customizing specific variable types: Text, Date, Number, True/False, Multiple Choice, and Computation)

To edit a variable at Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Select the variable from the list and click the **dit Component** button. The **Variable Editor** appears.
- 3. Make any changes. (Click any of the following links for information on customizing specific variable types: Text, Date, Number, True/False, Multiple Choice, and Computation)

If you're using Microsoft Word 2000 and above, you can double-click in a variable field in the template to open the **Variable Field** dialog box.

For instructions on renaming a variable, see Rename Components in a Single Template.

Format the Variable

You can cause an answer to be formatted a certain way in an assembled document—even if the user types the answer using a different format. For example, a number can be merged as *ONE HUNDRED DOLLARS* in the assembled document, even if the user types *100*. (See How Example Formats are Interpreted.)

You can assign the format either when you insert the variable field or when you create the variable component. Specifically, if you assign a format at the **Variable Field** dialog box, the format applies to that specific instance of the variable only—the format is a property of the field. However, if you assign a format to the component, that format is used each time the variable is inserted into the template. The format becomes a property of the component.

When you insert a new variable in a template (without replacing any template text), you can specify which format you want HotDocs to use for that instance. However, when you create a variable by replacing existing template text (see Insert a Variable Field in a Text Template), HotDocs looks at the text you have selected and attempts to suggest a field-specific format based on what it sees. For example, if you select

the template text, *14 November 1973* and then create a Date variable, HotDocs suggests a field-specific format of *3 June 1990*. This format will be applied to this instance of the variable only.

Finally, if users' answers shouldn't break across lines in the document, you can force the answer to be non-breaking. You do this by selecting the **Non-breaking** property next to the **Format** drop-down list at either the **Component Editor** or at the **Variable Field** dialog box.

To assign a field-specific format to a variable

- At the template, place your cursor inside a variable field and click the *Variable Field* button. The Variable Field dialog box appears.
- 2. If it's selected, clear **Use default**. The format options become available.
- 3. Click the **Format** drop-down button and select an appropriate format.
- 4. Optionally, select **Non-breaking**. This keeps the answer from breaking across lines in the document. (You can assign this property to a variable field even if you do not assign a specific format.)
- 5. Click **OK**.

When you assign a field-specific format to the variable, the example format is merged with the variable name in the field.

To assign a default format to a variable

- 1. At the template, place your cursor inside a variable field and click the **dit Component** button. The **Variable Editor** appears.
- 2. Click the **Default format** drop-down button and select an example format.
- 3. Optionally, select **Non-breaking** to keep the answer from breaking across lines in the document. (You can assign this property to a variable field even if you do not assign a specific format.)
- 4. Click **OK**.

Now, when you insert this variable in other places in the template, the format you have specified will be used. (You can, of course, override this by assigning a field-specific format.)

When you view the **Variable Field** dialog box, the default example format you have selected appears grayed in the **Format** field, and **Use default** is selected. If you clear this check box, the default format is suggested as the field-specific format. It also becomes a property of the field.

The **Use defaults** option at a **Variable Field** dialog box controls *all* field properties of a variable. (Field properties include answer formats, Multiple Choice merge text, and field formats.) This means that you cannot assign a field-specific property to a field and then assign a default property to the variable component. If you choose one type of property, all other properties must be the same type.

You can create your own formats by typing an example in the **Format** field. Follow the format of the existing examples—for example, use a variation on the number *nine* for number examples, and use a variation on the date *June 3, 1990* for date examples.

Create a Prompt for a Variable

By default, when you assign a name to a variable, that name is what appears in the interview to guide users in providing an answer. However, sometimes you may name your variables using a scheme that would be confusing for your users to understand. Likewise, some situations require more detailed instructions. HotDocs enables you to provide longer explanations by using a prompt. During the interview, the prompt replaces the variable name (and title, if one has been provided).

To create a prompt for a variable

- 1. Edit the variable. (See Edit a Variable.) The Variable Editor appears.
- 2. In the **Prompt** field, type the information about the variable you want the user to see.
- 3. Optionally, to assign font properties to the prompt text (such as bold or underline), enter a formatting dot code. (See Introduction: Dot Codes.)

You can include a variable in the **Prompt** field, which merges the user's answer into the prompt during the interview. See Use Variables and Scripts in Prompts, Dialog Element Text, and Resources.

When you group variables in a dialog, you can specify the visual relationship between variable prompts and answer fields by choosing an alignment option at the **Options** tab of the **Dialog Editor**. See Change a Dialog's Options.

You can make HotDocs show only an answer field—with no prompt or variable name—in the interview. To do this, type **NONE** in the **Prompt** field.

You can type an ampersand (&) immediately before a character in a variable prompt to make that character an accelerator key. During the interview, an accelerator key enables the user to move the cursor to that answer field by pressing the Alt key while also pressing the underlined character in the prompt. This functionality can create an issue if you need to include a word in your prompt that has an ampersand immediately preceding a character. To do this without creating an accelerator key, you must type two ampersands (&&) where you want the ampersand to appear as a single ampersand. For example, to have ''R&D'' as your variable prompt you would need to type ''R&&D''.

At a Glance: The Advanced tab (Variable Editor)

Advanced
Ask automatically When irrelevant: Default B
Warn when unanswered
☑ Save in answer file
Answer field width
G Regular: Full
Spreadsheet: Calculated
Default merge field properties
DField width: 0 🚔 Align answer: Left 🕒 🔻 Fill character: 🕞
U <u>n</u> answered text:
Test Update QK Cancel Save

You can open the **Advanced** tab from a Component Editor by opening the editor from the **Component Manager** or while editing the template.

Using the first three check boxes A you can select whether HotDocs will Ask the component automatically, Warn the user when this component is unanswered in the interview, and/or Save the answer entered for this component in the answer file. To the right of these check boxes is a drop-down list b that lets you decide that the component will do when it is irrelevant.

Below this is a section of options relating to the Answer field width C. Using the drop-down list you can select from the widths; **Full**, **Calculated** or **Exactly**. If you choose **Full** the answer field will fill the maximum width allotted to it in the interview, if you choose **Calculated** HotDocs will choose an estimated width based on the type of component, or if you choose **Exactly** the number field to the right will ungray and you can select the exact amount of units width the answer field requires. You can set this option for both Regular and Spreadsheet fields.

At the bottom of this dialog is a section where you can set the Default merge field properties. Use the first number field \boxed{D} to specify the minimum number of character spaces you want to merge into the assembled document. You can use the drop-down \boxed{D} menu to the right of that to select the alignment of the answer within a merge field. You can select from **Left** or **Right**. In the text field \boxed{D} to the right you can enter a character that HotDocs will use to fill any extra spaces in the answer field instead of the default; which is a space. The text field \boxed{D} at the bottom of the tab can be used to enter specific text you would like HotDocs to merge into the document in place of this variable if it is unanswered.

The **Advanced** tab on a **Computation Variable Editor** is slightly different. There is a text box at the top you can enter a title for the component then below this there is only the options for **Default merge field properties**. These are the same as on the other editors.

To learn more about using the Advanced tab follow the link below:

- Automatically Disable Irrelevant Variables in Interviews
- Control How HotDocs Processes a Variable
- Specify the Width of Answer Fields in the Interview
- Control How Answers Appear in the Assembled Document

Control How HotDocs Processes a Variable

You can have greater control over how HotDocs processes a variable. For instance, you can control whether a variable appears during the interview, whether the answer is saved, and whether users are warned when they leave the question unanswered. These options are properties of the variable component, and not the variable field.

By clearing **Ask automatically**, you cause a variable to be asked only when it is in a dialog that is specifically asked. (See How Variables and Dialogs Are Asked for details on how variable and dialogs are asked.)

When users leave questions unanswered during the interview, HotDocs leaves the dialog's icon in the interview outline partially answered and notes the unanswered question in the *End of Interview* dialog. When you clear **Warn when unanswered**, HotDocs suppresses these warnings. You might want to assign this option to variables that won't always be answered in all cases—for example, a variable that asks a middle name or a second address line.

Clearing **Save in answer file** prevents a variable's answer from being saved. You might use this feature if the variable is a temporary variable (for example, a variable that has its value SET to a temporary value for use in a script). This will keep HotDocs from marking the variable as answered and prompting the user to save the answer file.

When HotDocs Server assembles a template with the **Save in answer file** option cleared on some variables, the answers are still submitted to the server for document assembly, after that the host application needs to handle the removal of answers marked as don't save from the answer file.

These options are not available for Computation variables.

To control how HotDocs processes a variable

- 1. At the HotDocs template, edit a variable. (See Edit a Variable.)
- 2. Click the **Advanced** tab. The dialog box changes to show several advanced options.
- 3. Clear any of the following options, as explained in the following table:

То	Do This
Prevent the variable question from appearing in an interview unless the variable is used in a dialog that is specifically asked	Clear Ask Automatically.
Prevent the unanswered variable warning from appearing for this variable when a user finishes the interview. (The warning appears in both the <i>End of Interview</i> dialog and when you attempt to close an assembly window. HotDocs also uses the unanswered or partially answered dialog icon in the interview outline.)	Clear Warn when unanswered.
Prevent the variable's answer from being saved in an answer file	Clear Save in answer file.

Specify the Width of Answer Fields in the Interview

You can control the width of an answer field in an interview. This includes controlling column widths if the variable is repeated in a spreadsheet dialog.

To control the appearance of the variable's answer field

- 1. At the HotDocs template, edit a variable. (See Edit a Variable.)
- 2. At the **Variable Editor**, click the **Advanced** tab. The dialog box changes to show several advanced options, including answer field widths.
- 3. Click the **Regular** drop-down button and choose one of the following options:
 - Select **Full** to have HotDocs use the full width of the dialog pane to display the answer field.
 - Select **Calculated** to have HotDocs calculate an approximate width, based on the possible answer length.
 - Select **Exactly** and enter the specific number of units you want HotDocs to use for the answer field width.
- 4. Click the **Spreadsheet** drop-down button and choose one of the following options:
 - Select **Calculated** to have HotDocs calculate an approximate column width, based on the possible length of the answer and the actual width of the dialog pane.

• Select **At Least** and then enter the minimum number of characters that should determine the column width.

These options are not available for Computation variables.

The **Exactly** field width measurement takes into account the width of the entire field, including field borders, group boxes (on True/False and Multiple Choice variables), and the calendar icon for Date variables. One unit is about equal to the width of the character 5.

For text fields, **Calculated** field widths are calculated by looking at the **Maximum units** setting. If this isn't set, HotDocs will assume 30 units. For date and number fields, HotDocs figures out the longest field required to accept the date or maximum number. For multiple choice and true/false answer fields (or group boxes), HotDocs uses the width required to show all the options. One unit is about equal to the width of the character 5.

Control How Answers Appear in the Assembled Document

You can control some of the ways a variable's answer is merged into an assembled document. For example, you can control the characters that are filled in when you need to "lengthen" an answer. You can also control the text that is merged if the user chooses not to provide an answer.

Specifically, with the **Fill character** and **Field width** fields, you can specify how many characters wide an answer should be, and which character will be used to fill spaces the answer doesn't use. If the answer is longer than the width of the field, the entire answer is merged into the document. If the answer is shorter than the field width, HotDocs inserts the necessary number of fill characters. (The default fill character is a space.) If no answer is given, the minimum number of spaces doesn't affect the merge field. However, you can specify text you want merged into the document when the variable is unanswered by entering it in the **Unanswered text** field. This text will override the **Unanswered variable placeholder** users can select at HotDocs Options. (See Tips on Using Unanswered Variable Placeholders.)

The **Align answer** option lets you align the merged answer, based on the fill characters that are given. For example, if you select **Left**, the answer appears first, followed by any fill characters. If you select **Right**, the answer appears after any fill characters.

Finally, at times you may need to format the answer using a symbolic font. (For example, perhaps you need to merge a bar code.) To do this, you can assign a **Font** at the **Variable Field** dialog box.

To control how the variable's answer is merged into the assembled document

- 1. At the template, edit a variable by placing your cursor in the field and clicking the ****Variable Field** button. The **Variable Field** dialog box appears.
- 2. If it's selected, clear Use defaults. All the merge field properties become active.

- 3. Click **Show Advanced**. The dialog box changes to show several advanced options.
- 4. Select the options that apply, as explained in the following table:

То	Do This
Specify the minimum number of character spaces you want the merge field to use	Type a number in the Field width field, or use the up or down arrows to select a number
Specify the alignment of the answer within the merge field. (For example, if you are creating a	Choose Right or Left from the Align answer drop- down list.
column of dollar amounts, you can right-align the numbers.)	When aligning Number variables, HotDocs will align the answer in the answer field as well as in the merge field in the assembled document.
Fill extra spaces in an answer field with a specific character (instead of a space character, which is the default)	Type a character in the Fill character field.
Merge specific text when the variable is unanswered	Type the text in the Unanswered text field. (See Tips on Using Unanswered Variable Placeholders.)
Format answers using symbolic character fonts, such as bar code fonts	Click the Font drop-down button and choose a font from the list.
	This is a field-specific property available only in text templates. Additionally, you can assign this property only at the Variable Field dialog box.
	Make sure your users have the same fonts installed on their computers.

The **Use default** option at a **Variable Field** dialog box controls all of the properties of a variable field. (Field-specific properties include variable formats, Multiple Choice merge text, and formatting styles for answer fields. These options are visible when you click **Show Advanced**.) This means that you cannot assign a field-specific property to a field and then assign a default property to the variable component. If you choose one type of property, all other properties must be the same type.

Where you specify the **Advanced** options determines where the formatting occurs. If you change them at the **Variable Field** dialog box, the changes will apply to that instance of the variable only. If you change the options at the **Advanced** tab of the **Variable Editor**, the changes are used as defaults for every instance of the variable.

At a Glance: The Resource tab (Variable/Dialog Editor)

	Resource	
lesource type: Plain Text A	▼ Pop-up	Resource pane
		A
		-

You can open the **Resource** tab from a Dialog or Component Editor by opening the editor from the **Component Manager** or while editing the template.

On this tab you can assign a resource to the dialog to give the user more information when they are filling out answers in the interview. Using the drop-down list A at the top of the tab you can select from:

- Plain Text
- HTML Help
- Windows Help
- Folio Infobase
- Custom Program
- URL

The check boxes **B** let you choose different display options and are only available for certain resource types. Depending upon which type of resource you choose, you will see different options on the rest of the tab. See the table below:

Understanding Variables

Resource Type	Display Type	Options
Plain Text	Pop-up	Enter the resource text in the Text field.
HTML Help	Pop-up Resource Pane	In the first text field use the Browse button to locate the relevant HTML Help file. In the second text field type in the name of the Help file topic you wish to display.
Windows Help	Pop-up	In the first text field use the Browse button to locate the relevant Windows Help file. In the second text field type in the name of the Help file topic you wish to display.
Folio Infobase	None	In the first text field use the Rowse button to locate the relevant Infobase file. In the second text field type in the jump link to the page you wish to display.
Custom Program	None	In the first text field use the Rowse button to locate the relevant program file. In the second text field type in the command-line parameters you wish to use.
URL	Resource Pane	In the first text field use the Q Browse button to locate the relevant web page. In the second text field, type the name of the bookmark you want to link to on the page.

To learn more about using the Resource tab follow the link below:

- Add Resource Information to a Variable or Dialog
- Assign HTML Help as a Resource
- Assign Windows Help as a Resource
- Assign Folio Help as a Resource
- Assign a Custom Program as a Resource
- Assign a Web Page as a Resource

Add Resource Information to a Variable or Dialog

Sometimes including additional information about questions users are answering during an interview can make it easier for them to enter the correct answer. To provide this extra information, you can assign resources either to variables or to dialogs. Users can view this information by viewing the **Resource** pane during the interview.

By default, resources are displayed as plain text. However, you can use any of the following other formats or programs to create and display resources. (Click the associated link for instructions on creating the resource type.)

- WinHelp
- HTML Help
- Folio infobase
- Custom program
- Web page

To add a plain text resource to a variable or dialog

- 1. Open the variable or dialog to which you want to assign resources. (See Edit a Variable or Edit a Custom Dialog.)
- 2. Click the **Resource** tab.
- 3. Click the **Resource type** drop-down button and select **Plain Text**.
- 4. Type the information in the **Text** field.
- 5. Optionally, to use variables in the resource or make text in the resource conditional, enter the variable name, or enter an IF instruction or expression. (See Use Variables and Scripts in Prompts, Dialog Element Text, and Resources.)
- 6. Optionally, select **Pop-up** to display the information in a pop-up help window, rather than a typical window. (The pop-up window only appears when the user clicks the **Resource** button next to the answer field.)

During assembly, HotDocs displays the resource text either in the resource pane or in a text-only pop-up window, depending on your selection. (Pop-up windows stay open only until the user clicks somewhere else.)

You cannot use angle brackets with text between them in a plain-text resource (for example, **<Your Name>**). Instead, you must use the codes **<**; and **>**; to insert the brackets (for example, **<Your Name>**). (You can use brackets, however, if you put a space immediately after the opening angle bracket and immediately before the closing angle bracket, like this **< Your Name >**.)

When editing Multiple Choice variables, you can assign resource text to the entire variable or to individual options. At the **Multiple Choice Variable Editor**, click the **Resource** tab and make your selection at the **Resource for** drop-down list.

At a Glance: The Used In tab (Variable/Dialog Editor)

<u>)</u> ialog a	and database compo	nents that inc	clude this comp	oonent:		ĄŻ
			8			
List A	Il Components	Always lis	st all componen	nts when vie	wing this tab	

You can open the **Used In** tab from a Dialog or Component Editor by opening the editor from the **Component Manager** or while editing the template.

If the dialog has been used in any other components you will see a list of them in the main area \underline{B} by default it shows only dialog and database components. To sort this list into alphabetical order you can click on the $\frac{1}{2}$ **Sort** button A.

Below this main section are two further options **C** for the list. Clicking on the **List All Components** button will show every component that uses this dialog, not just dialog and database components. Checking the box next to it will have HotDocs list all components every time this tab is viewed.

To learn more about using the Used In tab follow the link below:

• View Relationship Between the Current Component and Other Components

View Relationship Between the Current Component and Other Components

When editing a component (such as a variable or a dialog), you can view a list of other components in the template that refer to it. You can view this list at the **Used In** tab of the component editor.

By default, when you first click the **Used In** tab, HotDocs displays only the dialog or database component to which the variable is linked, as well as any other dialogs in which the variable is used. To generate a complete list of component references (including other components that refer to the current component), you can click the **List All Components** button. If you want a complete list to be displayed every time you click the **Used In** tab, you must select **Always list all components when viewing this tab.** (Depending on the complexity of your template, you may experience a slight delay each time you view this list, which is why displaying it is optional.)

To view component relationships

- 1. At the template or at Component Manager, select the component and edit it. (See Edit a Variable.) The **Component Editor** appears.
- 2. Click the **Used In** tab. The window changes to show a field listing the dialog(s) (and database components) in which the component is used.
- Optionally, to view a list of other components that refer to this component, click List All Components. HotDocs creates a list of all components that refer to this particular component. (Depending on the complexity of your template, this may take several seconds.)
- Optionally, to always view this list of component cross-references each time you view the Used In tab (regardless of the component you are editing), select Always list all components when viewing this tab.

The dialog to which the variable is linked is marked with a red arrow. Dialogs that simply use the component are marked with a plain arrow. See Use the Same Variable or Clause in Two or More Dialogs.

If you make changes to other components in the component file while you have the component editor open, you can click **List All Components** again to regenerate the list of used components.

At a Glance: The Notes tab (Variable/Dialog Editor)

	Notes	
Notes:		
		*
	•	
	A	
		-
Test Update	<u>O</u> K Cancel	Sa <u>v</u> e

You can open the **Notes** tab from a Dialog or Component Editor by opening the editor from the **Component Manager** or while editing the template.

On this tab you can write any note about the dialog in the text field A. These do not show up anywhere in the interview and are strictly for the template developers.

To learn more about using the Notes tab follow the link below:

• Add Notes to Components

Add Notes to Components

As you create components, you may want to include notes about the component. For example, maybe you need to include information about restrictions you applied to a Number variable, or maybe you need to explain the purpose of the variable or dialog or how it should be used in the template. To do this, you
can add a note to the component. You add notes at the **Notes** tab of the component editor. Notes can be viewed by anyone editing the component.

You can add a comment to a variable field or instruction field so it can be visible by anyone editing the template text. See Add a Comment to a Variable or Instruction Field for details.

To add development notes to a component editor

- 1. Edit the component. (See Edit a Variable or Edit a Custom Dialog.) The component editor appears.
- 2. Click the **Notes** tab. The view changes to show a multi-line text field.
- 3. Enter the notes you want associated with the component.

Add a Comment to a Variable or Instruction Field

When you insert a variable or instruction in a template, you can add a comment to the insertion field. Comments are useful when you want to document information about the variable or instruction you are inserting and you want those comments to be viewable by anyone editing the text of the template.

Comments can also be useful when viewing a markup view of the template—the comment can describe (in easy-to-understand terms) the purpose of the field or provide a more reader-friendly name for the field.

If you want to add comments to a specific *component*, enter it at the **Notes** tab of the component editor. See Add Notes to Components for details.

To add a comment to a variable or instruction field

- Insert a new variable or edit an existing variable. (See Insert a Variable Field in a Text Template or Edit a Variable.) The Variable Field dialog box appears. Or, insert a new instruction or edit an existing instruction. The instruction field dialog box appears.
- 2. Click Show Advanced. The view expands to show advanced options.
- 3. Enter your comments in the **Comment** field.

When you click **OK**, HotDocs merges the field, with the comments merged after the variable name, like this: «Employee Salary //Calculates biweekly salary based on hourly rate».

To merge a comment in an END instruction (such as END IF, END REPEAT, OR END SPAN), place your cursor before the closing chevron, enter two forward slashes (//), and type your comment, like this: «END IF //Closing». If the END instruction contains a Keep Return code (|), enter the comment before the return code, like this: «END IF //Closing |».

Format an Answer as the Time of Day

You can merge an answer that is formatted as a time of day. To do this, you assign a pattern to a Text variable. During the interview, users will be prompted to enter the hours and minutes in 12-hour or 24-hour format.

To assign a time pattern to a Text variable

- 1. Follow the instructions for inserting a variable, selecting **Text** as the variable type. (See Insert a Variable Field in a Text Template.)
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- 3. Click the **Pattern** drop-down button and choose the time of day pattern you wish to use, based on the following information:
 - Choose **99:99** to insert a time in 24-hour format.
 - Choose **99:99 am** to insert a time in 12-hour format. The morning or afternoon designation will not include periods (for example, **am** or **pm**).
 - Choose **99:99 a.m.** to insert a time in 12-hour format. The morning or afternoon designation will include periods (for example, **a.m.** or **p.m.**).

If your component file was created before the release of HotDocs 2006, you must create the time-of-day pattern. To do this, type **99:99**, **99:99 a.m.**, or **99:99 a.m.** in the **Pattern** field.

Number Variable Formats, Percentages, and Fractions

Formats

In Number variable formats, HotDocs distinguishes between 9's and 0's. A 9 displays a digit unless it is a leading zero or a trailing zero after the decimal point. A 0 displays digits when it is a trailing zero. For example, if the number is 850, the example format 9999.99 would merge 850 while the example format 0009.00 would merge 850.00. Also, if you are using a Number variable to merge a decimal, make sure the **Decimal places** field contains a value greater than 0.

If you want commas to be merged into the number answer, make sure your format does not use leading zeros.

Another set of similar example formats includes the examples *NINE*, *NONE*, and *ZERO*. These three example formats all merge an uppercase alphabetic number unless the answer is 0. When the answer is 0, *NINE* merges *NO*, *NONE* merges *NONE*, and *ZERO* merges *ZERO*. (You can also enter these formats in lowercase or initial caps.)

Percentages

You can format number answers as percentages. To do this, you must assign an example format of 99% to the variable, assign 2 decimal places to the variable, and set the **Maximum** to **1**. You should also include a variable prompt to instruct the user to enter the answer as a decimal—for example, 0.50 for a result of 50%.

Another option is to include a percent symbol in the template text, so that the user can enter a whole number. If you do, you might want to set a **Maximum** limit of *100*, and you might want to include a resource or prompt that explains how the user should enter the answer.

Fractions

Example formats can also be used to enter fractions. While HotDocs does not allow users to directly enter fractions, it can convert a decimal value to a fraction before it merges the value into the document. To merge a fraction, first make sure the **Decimal places** field contains a number greater than 0 so users can enter a decimal answer. Then specify a fraction as an example format. The number in the denominator of the example will be the number used in the denominator of the merged value. Fractions are rounded and simplified. For example:

Format Examples	The user types 2.3	The user types 2.5	The user types 2.6
9 1/3	2 1/3	2 2/3	2 2/3
9 1/8	2 1/4	2 1/2	2 5/8

Work with Multiple Choice Options

By default, when you first create a Multiple Choice variable, you have limited space for entering prompts and merge text. If you need more space, click the **Options** tab and enter the information there. You can also have HotDocs suggest one of the options as a possible answer for the user.

To provide additional information for option prompts and merge texts

- 1. Create a Multiple Choice variable.
- 2. After specifying the variable properties at the **Properties** tab of the **Multiple Choice Variable Editor**, click the **Options** tab.

- 3. Click the **Option** drop-down button and select the option you want to modify.
- 4. Change any of the following settings:
 - Type a new prompt in the **Prompt** field or edit the existing prompt. (500-character limit)
 - Type a new merge text option in the **Default merge text** field or edit the existing merge text. (1,000-character limit)
 - Select **Automatically select this option if variable is unanswered when displayed** to have HotDocs pre-select one or more options for the user. During the interview, the user can either accept this selection, or make a different selection.

Do not use the **Automatically select this option if variable is unanswered when displayed** option if you are using the Multiple Choice variable in a repeated dialog. Because of the way HotDocs processes dialogs, whenever you set a default value for a variable, there will always be one extra repetition in the dialog. To set a default option for a repeated Multiple Choice variable, use a computation script that conditions the default value.

You can assign resources to each specific option of a Multiple Choice variable. See the note at the end of Add Resource Information to a Variable or Dialog.

If you are using a SET instruction, you can set two or more options for a Multiple Choice variable. To do this, separate each option with a vertical bar (for example, SET MC Variable TO "Option1|Option2|Option3"). The Multiple Choice variable must have the Select All That Apply property set.

Specify Merge Text Options as Default or Field-Specific Properties

When you create a Multiple Choice variable, you can control whether merge text options are automatically available each time you insert the variable, or whether they are only available for a specific instance of the variable. You determine this by typing your merge text either at the **Multiple Choice Variable Editor** or at the **Variable Field** dialog box.

To assign default merge text to a Multiple Choice variable

- 1. Create or edit a Multiple Choice variable. (See Customize a Number Variable.)
- 2. At the **Multiple Choice Variable Editor**, enter your options in the **Option** column and enter the corresponding merge text in the **Default Merge Text** column.
- 3. Click **OK.** The merge text options you just entered are now saved with the other component properties.
- 4. At the **Variable Field** dialog box for the Multiple Choice variable, select **Use default**. HotDocs grays all field properties for the variable.

To assign field-specific merge text to a Multiple Choice variable

- 1. Create or edit a Multiple Choice variable (see Customize a Multiple Choice Variable and Edit a Variable.)
- 2. At the Variable Field dialog box, clear Use defaults. HotDocs ungrays the Merge Text column.
- 3. Enter the merge text that corresponds with the **Options** you have specified.

The **Use defaults** option at a **Variable Field** dialog box controls all field properties of a variable. (Field-specific properties include variable formats, Multiple Choice merge text, and formatting styles for answer fields, which are visible when you click **Show Advanced**.) This means that you cannot assign a field-specific property to a field and then assign a default property to the component. If you choose one type of property, all other properties must be the same type.

When assigning merge text to a variable, you can assign an existing merge text group by clicking the **Default Merge Text** column heading and choosing the option from the drop-down list.

Copy and Paste Columns in a Multiple Choice Variable Spreadsheet

At times, you may have an Excel spreadsheet or Microsoft Word table that contains the data you want to use as options, prompts, and merge text for a Multiple Choice variable. Other times you may want to copy this information from a Multiple Choice variable to a spreadsheet or table. Using the shortcut menu at the Multiple Choice Variable Editor, you can do this.

To copy data from a spreadsheet or table and paste it into the Multiple Choice Variable Editor

- 1. In the spreadsheet or table, select or highlight the columns and rows that contain the information you want to copy and copy them. (See the application's help for information on copying.)
- 2. Edit the Multiple Choice variable. (See Edit a Variable.) The **Multiple Choice Variable Editor** appears.
- 3. Place your cursor in the spreadsheet column where you want the new data to be used.
- 4. Right-click and choose **Paste Multiple** from the shortcut menu. The data is pasted into the spreadsheet.

To copy data from the Multiple Choice Variable Editor and paste it into a spreadsheet or table

1. In the Multiple Choice Variable Editor, place your cursor in the spreadsheet column where you want to copy the data.

- 2. Right-click and choose **Copy Column** (copies the text from the current cell down for this column only) or **Copy All Columns** (copies the text from the current cell down in all columns) from the shortcut menu.
- 3. In the spreadsheet or table, place your cursor where you want to paste the data and paste it. (See the application's help for information on pasting.)

To copy the spreadsheet from one Multiple Choice variable and paste it into another Multiple Choice variable, right-click on the first cell in the spreadsheet and choose **Copy All Columns** from the shortcut menu. Then edit the second variable, place your cursor in the first cell of the spreadsheet, and choose **Paste Multiple** from the shortcut menu.

Name a Group of Merge Text Values

When you add merge text values to a Multiple Choice variable, the list of values you enter becomes a separate HotDocs component, which can be used in other Multiple Choice variables. By default, HotDocs names this component using all the values separated by slashes (for example,

Boise/Olympia/Salem/Helena/Etc.) Sometimes, the merge text list can be quite long. In these situations, it may work to assign a different component name (for example, *Western States Region*) to the merge text values. You must use Component Manager to assign this new name.

To edit merge text and assign a new name

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Click the **Components** drop-down button and select **Merge Text**. HotDocs displays the different merge text groups available in your component file.
- 3. Select a merge text group and click the ***Rename Component** button. The **Rename Component** dialog box appears.
- 4. Enter a name for the merge text group in the **New name** field.
- 5. Click **Rename**. The component is renamed.

Work with Variables in Headers, Footers, Footnotes, and Text Boxes

Depending on whether you use Word or WordPerfect, how you insert a variable in a header, footer, footnote, or text box is different. The following table explains how to use variables in the various areas of a word processor template.

	Microsoft Word	WordPerfect
Headers and footers	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.
Footnotes	Create the variable in Component Manager and drag it into the field. Edit the variable using Component Manager.	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.
Text Boxes	Variables in text boxes are ignored during document assembly.	Create the variable directly in the text box, or create the variable in Component Manager and drag it into the text box. Edit the variable directly in the box or use Component Manager.

Using buttons in the HotDocs Navigation toolbar will have no effect while editing the contents of a header or footer.

When inserting variables or instructions in a WordPerfect header, footer, footnote, or text box, you cannot highlight text and then replace it with the field. You must simply insert the variable by clicking in the text.

HotDocs 5 users: In HotDocs 5, you had to specify a component file property that instructed HotDocs to assemble variables in Word headers and footers. HotDocs 11 automatically assembles these, so the option has been removed from the **Component File Properties** dialog box.

If a Word template contains headers and footers with variables, you may find your variables being asked "out of order." You can create a custom interview to control the order your variables and dialogs are asked. See **Define a Custom Interview** for details.

Use Pattern Codes to Create a Custom Pattern

HotDocs provides Text variable patterns: a telephone number pattern, time of day patterns, and a U.S. Social Security number pattern. You can also create your own pattern by typing it directly in the **Pattern** field (at the **Text Variable Editor**) or by clicking the **Component Manager** button at the template, selecting **Text Pattern** from the **Components** list, and clicking the **New Component** button.

Use the following codes to create patterns. These codes represent the types of characters the user can enter. Additionally, you can insert hyphens, periods, static text, and so forth with these codes.

Code	What HotDocs Will Require and Insert in the Document
х	any character
9	any numeric character
Α	any alphabetic character
U	uppercase alphabetic character
L	lowercase alphabetic character
Ν	alphanumeric character (9 and A)
н	hexadecimal character (0-9, A-F, a-f)

If you need a character in your pattern that HotDocs may interpret as a code, you can force the character to appear as part of the pattern and not as a code by typing a forward slash (/) and then the character. For example, if you need a pattern to use the X character, you must place the forward slash before the X so that HotDocs doesn't try to replace the X with a character the user types.

Use Variables and Scripts in Prompts, Dialog Element Text, and Resources

You can further customize the interview process by including variables in prompts, in plain text resources, and in dialog element text. You can also include variables in dialog titles and Multiple Choice option prompts and merge text.

Additionally, you can include IF and REPEAT instructions in plain text resources. You should note, however, that placing an instruction in a resource will not cause variables used in the instruction to be asked during the interview. If an instruction relies on a variable being asked in order to be processed, you must make sure those references are resolved earlier in the interview, before the resource is displayed.

Referring to variables and instructions in other components can provide users with more specific information about the answer they need to enter during the interview. For example, if you ask for a defendant's name at the beginning of an interview, whatever name the user enters can appear in a prompt or resource later in the interview.

To customize prompts, dialog element text, dialog titles, plain-text resources, and Multiple Choice merge text and prompts

- 1. Edit the component where you want to merge the variable reference or script. (See Edit a Variable or Gather Questions into a Custom Dialog.)
- 2. Place your cursor where you want the variable reference (for example, a prompt, title, or dialog element text field), right-click, and choose **Variable Field**. The **Variable Field** dialog box appears.

To include variable answers in Multiple Choice merge text and prompts, you must either enter the variables at the **Options** tab, or you must manually enter the variable references in the options spreadsheet.

- 3. Enter the variable information and click **OK**. The variable is merged in the field.
- Optionally, to enter a script in a plain-text resource, right-click in the **Text** field and choose **IF** Field or **REPEAT Field** from the shortcut menu, depending on the type of instruction you are inserting.

You can also merge variable field references by typing double-angle brackets (<< >>), followed by the variable name or instruction keyword. HotDocs will convert the double-angle brackets to HotDocs chevrons (« »).

If Component Manager is open, you can also drag a variable from Component Manager and HotDocs will automatically include the chevrons.

At a Glance: The Go To Dialog Box



After opening the **Computation Editor** of a computation variable, from the **Component Manager** or while editing the template, you can click on the $\begin{bmatrix} I \\ I \end{bmatrix} = Go To$ button on the toolbar to open the **Go To** dialog box where you can jump to any position in the script.

Using the check boxes A you can choose if you would like to go to a line or a character in the computation. The first option, **Line number**, specifies the line number in the script where you want the cursor to move to. Each line of script is counted, even if there is nothing on the line. While the second option, **Character position**, specifies the character position in the script where you want the cursor to move to. Each character, including space characters between script items, is counted.

In the field **B** below you can enter the line number or character position number you require.

To learn more about using the Script Editor features follow the link below:

• Use the Script Editor

Use the Script Editor

When writing a computation script, there are several tools you can use to make the process easier.

To use the script editor

- 1. Edit the script or expression.
- 2. At the **Script** or **Expression** field, complete any of the following tasks:

То	Do This
Have HotDocs automatically complete keywords, component names, and Multiple Choice options as you type	Place your cursor in the Script field and press Ctrl+Spacebar . HotDocs displays a list of instruction and expression keywords, constant values (such as month abbreviations), and components. Type a portion of the keyword, component name, or Multiple Choice option for which you are searching. As you type, HotDocs filters the list to show only those keywords that contain the text you have typed. Once selected, press Enter to merge it into your script. To keep seldom-used keywords out of the auto-complete list, click the Options button and clear Include seldom-used keywords in auto-complete list .
Access just a list of components	Press the F5 key, select the component, and press Enter .
Access just a list of keywords	Press Shift+F5 , select the keyword, and press Enter .
Display a syntactical hint of how an instruction or expression should be used	Place your cursor inside the keyword and press the F7 key. HotDocs displays a small ToolTip that shows the entire model as well as the type of value it produces, if it's an expression.
Indent matching pairs of IF and REPEAT instructions based on the level of their insertion	Click the Auto Format button.

Undo (or cancel) an action you just performed	Click the Dundo button. HotDocs reverses the change you made.
Redo (or re-implement) an action you just performed	Click the 🔉 Redo button. HotDocs reapplies the change you made.
Cut or copy and paste a selected portion of the script	Select the portion of the script you want to cut or copy, and then click the Cut button or the Copy button. The script is copied to the Clipboard. To paste the script in a new location, insert your cursor at that location and click the Caste Paste button.
Find a specific string of text in the script	Click the A Find button and enter your search text in the Find what field. To find the next instance of the text, click the A Find Next button.
	To find only those instances of text that are complete words, select Find whole words only . To find only those instances that have the same capitalization as the text for which you are searching, select Match case .
Find a specific string of text in the script and replace it with another string of text	Click the defined and Replace button. Enter the search text for which you are searching in the Find what field, and then enter the replacement text in the Replace with field. Once you have entered the required text, click Replace , Replace All , or Find Next .
	To find and replace script text in a specific block of script, select the block of script first and then click the the Find and Replace button. At the Find and Replace dialog box, select Replace only in selected text .
Move your cursor to a specific location in the script	Click the L E Go To button. This displays the Go To dialog box, where you can enter the line number or character position of where you want your cursor to move.
Indent or outdent a block of the script	Select the portion of the script you want to indent or outdent and click the Findent button or the FOUTDENT button. (Click repeatedly to increase or decrease the indent.)
Cause HotDocs to ignore a section of the script when it processes it, or insert a comment in the script	Highlight the block of the script you want to comment and click the Comment Block button. This puts two forward slashes in front of each line of the script, which instructs HotDocs to ignore this section. To uncomment it, highlight the text and click the Comment Block button.

Match an IF or REPEAT instruction with its END IF or END REPEAT instruction (or vice versa)	Place your cursor inside the instruction, right-click, and select Match IF/REPEAT from the shortcut menu. (You can also press Ctrl+M .)	
Highlight an entire IF or REPEAT instruction block (meaning everything between a beginning and ending IF/REPEAT instruction)	Place your cursor inside the instruction, right-click, and select Select IF/REPEAT from the shortcut menu. (You can also press Ctrl+Shift+M .)	
View helpful information while using the script editor, including accessing help topics for the different instructions and expressions you can use in your script	Click the HotDocs Help button.	
Customize the way the script editor works	Click the Options button and make your changes. (See Change Script Editing Options.)	
Insert line breaks and tab characters in a script	See Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts.	
Assign formatting characteristics to literal text strings in a script	Insert the corresponding dot code. See one of the following topics for details:	
	Change Font Properties of Text	
	Insert Characters in Text Strings	
	 Add Punctuation and Capitalization to Sentences 	

To access the script editor toolbar using the keyboard, press F10.

To access help for each instruction or expression, first select the model in one of the lists and then press **Ctrl+F1**.

For an explanation of the HotDocs scripting language, see Understand the HotDocs Scripting Language. For details on creating a Computation variable, see Customize a Computation Variable. For a list of instruction and expression models, see Introduction: Instruction and Expression Models.

At a Glance: The Locals tab (Variable/Dialog Editor)

	Type	
1 🔼	B	
<u>I</u> nsert <u>D</u> elete		
ocal variables:		
	E	T

You can open the **Locals** tab from a Dialog or Component Editor by opening the editor from the **Component Manager** or while editing the template. In a Dialog Editor you would only see the option to use Local variables rather than both Local variables and Parameters.

The first section in this tab is Parameters. In the text box \underline{A} you can enter a name for the Parameter then using the drop-down list \underline{B} you can select the variable type for your Parameter from the following options:

- Text
- Number
- Date
- True/False

Once you have one Parameter in the list you can Insert another or Delete this one using the buttons C below.

The Local variable section is used in the same way; entering the name in the text field \mathbf{D} , choosing the type from the drop-down list \mathbf{E} and using the buttons \mathbf{F} below to edit the list.

To learn more about using Parameters and Local Variables follow the link below:

- Local Variables
- Parameters

Tips on Using and Editing Variables

Tips on Naming Your Variables

How you name variables in your templates depends largely on personal preference or your project guidelines. However, there are a few rules and suggestions that can make the automation and assembly process easier.

Assign Common, Useful Names

A variable name guides the user in answering each question, so you should choose meaningful names as you design your templates. A variable name can have up to 50 characters, including letters, numbers, and some symbols. However, the first character must be a letter. Each variable name must be unique—even if the variables are different types, their names cannot be identical.

It is often useful to put the subject of the variable name first. This allows you to sort variables alphabetically and bring up the variables with the same subject together. For example using **ClientName** rather than **Name of Client** (and **ClientAddress**, **ClientEmail** etc) will allow you to see all Client variables together rather than all Name variables together.

Some developers find it useful to append a type identifier to the name. For example, **Beneficiary Age CO** where CO marks it as a computation variable.. Following this method allows you to quickly identify types of variables and use the same name with a different identifier for related variables.

Do not use any words typed in all uppercase letters in your variable names. Because HotDocs instruction and expression keywords use uppercase letters, you may inadvertently use a word that may someday become a keyword, which will prevent HotDocs from reading your variable name correctly. (See Introduction: Instruction and Expression Models.)

Finally, there are certain characters you CAN NOT use when naming your components. They are:

Character Name

•	Period
\$	Dollar sign
u	Quotation mark
:	Colon
[]	Brackets
,	Comma
()	Parenthesis
%	Percent

These characters can be used only if there is a character other than a space immediately before or after it:

Character	Name
+	Plus
-	Hyphen
*	Asterisk
/	Forward slash
> <	Greater Than and Less Than signs
>= <=	Greater Than or Equal To and Less Than or Equal To signs
=	Equals
!=	Does not equal

In addition, if you are developing templates for use with HotDocs Server, the following characters should be avoided when naming components:

Character	Name
?	Question mark
&	Ampersand
@	At symbol

Share Answers Between Templates

If you have multiple templates that use the same information, your users can share answers across a set of templates. To do this, you must create variables with the same names in each template, and users must use the same answer file when assembling documents. (See Assemble a Text or Form Document and Create a New Answer File.) When the variable names are identical, the answers in separate documents are

also identical. However, variable names are context-sensitive and any limits entered for the first variable (such as maximum number of characters, patterns, and decimal places) must be entered for each duplicate variable or the answers will not flow from one document to the next.

When many variables are shared throughout your template set, you may find it easier to use one component file for all of your templates. (See Use One Component File for Multiple Templates.) Or, if you want each template to use an independent component file, copy the contents from the first template's component file into each subsequent component file. (See Copy Components From One File to Another.) Using either method, you can use variables you have already created, rather than recreating each variable.

How Example Formats are Interpreted

The following describes how answers will be formatted, based on the assigned example format:

Text Variables

Example Format	User's Answer	How Answer is Formatted in Document
like this	Marianne Stevens	marianne stevens
Like this	Marianne Stevens	Marianne stevens
Like This	Marianne Stevens	Marianne Stevens
LIKE THIS	Marianne Stevens	MARIANNE STEVENS
Like a This	our client, Marianne Stevens, is the Plaintiff in the case	Our client, Marianne Stevens, is the Plaintiff in the case

Number Variables

Example Format	User's Answer	How Answer is Formatted in Document
09	4	04
	78	78
9	7,898	7,898
9 1/8	2.3	2 1/4

9,999.00	9214	9,214.00
	9214.36	9.214.36
9,999.99999	87.5984	87.5984
9.9	87.6	87.6
9.9.a	287.3	287.3.a
(this format is best used with PN variables)		
9999	12587	12587
9th	23	23rd
nine	1,278	one thousand two hundred seventy-eight
NINE	1,278	ONE THOUSAND TWO HUNDRED SEVENTY- EIGHT
Nine Dollars and Twelve Cents	5.36	Five Dollars and Thirty-Six Cents
ninth	782	seven hundred eighty-second
alpha	12	twelve
Alpha		Twelve
ALPHA		TWELVE
ordinal	25	twenty-fifth
Ordinal		Twenty-Fifth
ORDINAL		TWENTY-FIFTH

Many of these formats will change depending on whether you assign decimal places to the answer. How many decimal places you assign also affects the outcome.

Date Variables

Example Format	User's Answer	How Answer is Formatted in Document
03 JUN 90	November 6, 2000	06 NOV 00
06/30/90	November 6, 2000	11/06/00
3 June 1990	November 6, 2000	6 November 2000
3rd day of June, 1990	November 6, 2000	6th day of November, 2000
6/3/90	November 6, 2000	11/6/00
June 3, 1990	November 6, 2000	November 6, 2000
June 3rd	November 6, 2000	November 6th
June Third, One Thousand Nine Hundred Ninety	November 6, 2000	November Sixth, Two Thousand
Sunday	November 6, 2000	Monday
Sunday, June 3, 1990	November 6, 2000	Monday, November 6, 2000
Third day of June, 1990	November 6, 2000	Sixth day of November, 2000

For additional date formats, please see Additional Date Formats.

True/False Variables

Example Format	User's Answer	How Answer is Formatted in Document
/x	Yes	(nothing)
	No	Х
true/false	Yes	true
	No	false
x/	Yes	Х
	No	(nothing)
yes/no	Yes	yes
	No	no

Multiple Choice Variables

The format assigned to a Multiple Choice variable depends on whether the variable is set to allow only one selection (**Select One**) or multiple selections (**Select All That Apply**). If set to **Select One**, the answer will be formatted like Text variables are formatted. Please see the formats table for Text Variables for a description of each option. If the variable is set to **Select All That Apply**, the answer will be formatted like repeated answer lists are formatted. Please see the table for answer lists for a description of each option.

Computation Variables

Formats for Computation variables depend on the type of computation script you are creating. If you are calculating numbers, the formats are the same as Number variables. If you are working with text, the formats available are the same as Text variables.

Answer Lists

To create a list of answers, you can either create a Multiple Choice variable and assign the **Select All That Apply** property for it, or you can create a REPEAT instruction. The following list of formats describes the format for each type of list.

When creating lists of answers using a REPEAT instruction, the formats with all capital letters only capitalize the conjunction in the sentence, for example, the word AND. Individual answers in the list are either merged as the user types them or as you have specified for each variable.

Example Format	User's Answer	How Answer is Formatted in Document
a, and b	apples and oranges	Multiple Choice: apples, and oranges
		Repeated List: apples, and oranges
a, b	apples, oranges, and cherries	Multiple Choice: apples, oranges, cherries
		Repeated List: apples, oranges, cherries
a, b and c	apples, oranges, and cherries	Multiple Choice: apples, oranges and cherries
		Repeated List: apples, oranges and cherries
A, b and c	apples, oranges, and cherries	Multiple Choice: Apples, oranges and cherries
		Repeated List: apples, oranges and cherries

A, B and C	apples, oranges, and cherries	Multiple Choice: Apples, Oranges and Cherries
		Repeated List: apples, oranges and cherries
A, B AND C	apples, oranges, and cherries	Multiple Choice: Apples, Oranges AND Cherries
		Repeated List: apples, oranges AND cherries
A, b or c	apples, oranges, and cherries	Multiple Choice: Apples, oranges or cherries
		Repeated List: apples, oranges or cherries
a, b, and c	apples, oranges, and cherries	Multiple Choice: apples, oranges, and cherries
		Repeated List: apples, oranges, and cherries
a; b; and c	apples, oranges, and cherries	Multiple Choice: apples; oranges; and cherries
		Repeated List: apples; oranges; and cherries

Additional Date Formats

You can also create date variables using the following set of format codes:

All case sensitive formats operate the same way—if the first character of the format is uppercase and the second is lowercase, then the first letter of the formatted value is uppercase and all the rest are lowercase. If the first and second characters of the format are uppercase, then the entire formatted value is uppercase.

The following formats are not supported in HotDocs 2008 and earlier. If you plan to use your templates with any of these previous versions, use these formats.

Property Name	Description	Example Usage	Example Formatted Answer
d	Numeric day	d	1
dd	Two-digit numeric day	dd	01

ddd	Case-sensitive weekday abbreviation	ddd, Ddd, DDD	mon, Mon, MON
dddd	Case-sensitive spelled-out weekday	dddd, Dddd, DDDD	monday, Monday, MONDAY
dth	Case-sensitive numeric ordinal day	dth, dTH	1st, 1ST
dy	Case-sensitive spelled-out day	dy, Dy, DY	first, First, FIRST
m	Numeric month	m	1
mm	Two-digit numeric month	mm	01
mmm	Case-sensitive month abbreviation	mmm, Mmm, MMM	jan, Jan, JAN
mmmm	Case-sensitive spelled-out month	mmmm, Mmmm, MMMM	january, January, JANUARY
у	Numeric year	у	2010
уу	Two-digit numeric year	уу	09
уууу	Four-digit numeric year	уууу	2010
yr	Case-sensitive spelled-out year	yr, Yr, YR	two thousand and nine, Two Thousand And Nine, TWO THOUSAND AND NINE

Following are some examples of how to use these types of date formats:

How It's Used	How the Answer is Formatted
d/m/yy	1/1/08
dd/mm/yy	01/01/08
dth day of Mmm	1st day of Jan
dy day of the month	first day of the month
Мттт, уууу	January, 2008
Mmm yyyy	Jan 2008
Yr	Two Thousand And Eight
Ddd, dd/mm/yy	Mon, 01/01/07
Dddd, Mmm dth	Monday, Jan 1st

Understand the HotDocs Scripting Language

Learning the Language

As you use HotDocs, you may need to tell HotDocs to perform a particular task, such as insert one template into another, hide variables in dialogs, or perform some action based on an answer the user provides. Similarly, you may need to add up several dollar amounts, or find the number of years between two given dates. Or, you may need to search a user's answer for a certain block of text.

To accomplish this, you must use the HotDocs scripting language, which consists of instructions, expressions, operators, and values—such as text, numbers, dates, or answers users enter. To help you learn this language, HotDocs provides you with instruction and expression *models*.

Specifically, an *instruction model* tells HotDocs to perform some sort of function, while an *expression model* retrieves a special value. Most models also include *placeholders*, which you must replace with a value. Possible values include text strings, number amounts, other models, and HotDocs variables. A model will not work until all its placeholders are replaced.

In addition to using instructions and expressions, you can use operators to control how the script is processed. Most operators are common mathematical signs, but there are also Boolean operators such as *AND* and *OR*. The operator *AND* means the statement to the left and the statement to the right must both be true. The operator *OR* means either the statement to the left or the statement to the right must be true. The operator () means to perform the operation between the parentheses first. (See Use Operators When Scripting.)

Rules for Writing Scripts

Instruction and expression models are designed to help you accurately write scripts. However, there are still certain rules you must follow, particularly when it comes to replacing placeholders with values. Failure to follow these rules may result in syntax errors when HotDocs attempts to process the script during assembly.

When using literal values:

- Decimal numbers less than one must have a zero before the decimal point (0.125 not .125).
- Numbers cannot contain commas (1250 not 1,250).
- Dates must be in the form 3 JUN 1990.
- Text strings must be inside quotation marks.

To help you format these numbers correctly, use the **Enter a Date**, **Enter a Number**, **Enter some Text**, and **Enter True or False** expression models.

When manually typing the script:

- Component names must be entered exactly as they appear in the **Components** list, including capitalization.
- Components and operators must be separated by spaces.
- Operators must be entered exactly as they appear in the **Operators** list, including capitalization.
- Variables used inside a literal text string (or between quotation marks) in a computation must be inside chevrons (« »). If the variable name is not inside chevrons, the variable name—not the variable's value—will be merged as the answer. (For example, "«Employee Name» must wait «Number of Months» months before applying.")

If you want to include a chevron in a HotDocs text template either as a part of your document text or a literal text string, you need to double the opening chevron (but *not* the closing chevron), so that HotDocs does not interpret the chevrons (and any text between them) as a field. The output is a single chevron, as follows: input: «some text»,output: «some text».

- Keywords, such as TRUE, FALSE, AND, END IF, and so forth, must be in all capital letters.
- Only **TRUE** or **FALSE** can be used for a true/false value.
- If you want the computation to return a combination of text, variables, and values, you must concatenate (or join) the data with the **Add** (+) operator. Otherwise, HotDocs returns only the last piece of text, variable, or value.
- When using the RESULT expression, you must first set the computation to a value. If the computation will return text, set it to a specific value by typing text inside quotation marks at the start of the script. (To set a text computation to an empty value, type nothing between the quotation marks—not even a space.) If the computation will return a number, set it to a value by typing a number at the very beginning of the script. (To set a number computation to nothing, type *0* (zero).)

Writing the Script

There are three places in HotDocs where you can write scripts: The **Computation Editor**, the **Script** tab of the **Dialog Editor**, and the **Expression** field of the **IF Field** dialog box. Each of these places provides you with the tools you need to write the script. Which tools you use, however, depends on personal preference, as well as your skill level.

If you are learning the scripting language, it is suggested that you select and drag instructions and expressions from the models lists to the **Script** field. You can also drag operators and components from their respective lists into the **Script** field. If there are placeholders, replace them by dragging components or other models onto them. To correctly format literal values (such as text, numbers, dates, and true/false values) use the **Enter...** expression models.

If you prefer to use the keyboard rather than the mouse, press the **Tab** key to move from one field in the dialog box to the next. Use the arrow keys to select specific components, operators, and models, and use the **Insert** key to bring the selected element into the **Script** field. To replace placeholders in the script, insert your cursor in the placeholder text and then locate the value you want to replace it with in the lists below and press the **Insert** key.

Once you become familiar with the instructions and expressions, you can type your scripts directly in the **Script** or **Expression** field. To ensure you use the correct instruction and expression keywords and component names, you can access lists of these things using keyboard shortcuts and other options available in the script editor. When you do this, HotDocs displays an auto-complete list from which you can choose the instruction or expression you are typing. For complete instructions, See Use the Script Editor.

Understand How Component Titles and Prompts are Used

In HotDocs, you can assign titles to different components, including variables, dialogs, clauses, and databases. Titles are used in place of component names, which, when used in the context of an interview, may not make much sense to the user. (For example, a component may be named *Client Birth Date DA*, but the name you want users to see in the interview is *Client's Birth Date*.)

How HotDocs uses these titles, especially for components that may also have a prompt, depends on where the component appears in the interview. Following is a description of the components, where they are used in the interview, and the order in which titles, prompts, and names are used when identifying the component.

For example, say you have a Text variable that is used in a dialog. The first property HotDocs will try to use as the question for the Text variable in the dialog is the prompt itself. If no prompt is specified, it will use the title. If no title is given, it will use the variable name.

Type of Component	Where Used in Interview	Order in Which Properties are Used
Variable (used or not used in a custom dialog)	Answer field in the dialog pane	Prompt / Title / Variable name
Variable (used in a custom dialog)	Text in the column heading of a spreadsheet (including in an answer source column)	Prompt / Title / Variable name
Variable (not used in a custom dialog)	Dialog title in the interview outline or dialog title in the dialog pane title bar	Title / Prompt / Variable name
Variable (used or not used in a custom dialog)	Reference in a Question or Answer Summary	Prompt / Title / Variable name
Clause (referenced in a clause library)	Clause title in the clause library	Title / Clause name

Clause (referenced using an IF instruction in a template; is not used in a custom dialog)	Dialog title in interview outline or dialog title in the dialog pane title bar	Title / Prompt / Clause name
Clause (referenced using an IF instruction in a template; is not used in a custom dialog)	Answer field in the dialog pane	Prompt / Title / Clause name
Database component	Dialog title in interview outline or dialog title in dialog pane title bar	Title / Component name
Database component	Table in dialog pane	Prompt / Title / Component name
Dialog (nonrepeated)	Dialog title in interview outline or dialog title in dialog pane title bar	Dialog title / Dialog name
Dialog (repeated as series)	Dialog title in interview outline	Repeated series label / Dialog title / Dialog name
Dialog (repeated as series)	Dialog title in dialog pane title bar	Dialog title / Dialog name
Dialog (repeated as spreadsheet)	Dialog title in interview outline or dialog title in dialog pane title bar	Dialog title / Dialog name

Remember that when variables are linked to or used in dialogs, the text that appears in the interview outline and in the dialog pane title bar comes from the dialog, not the variable.

You should always test assemble your template to make sure these combinations of titles, prompts, and variable names make sense for the interview. See Introduction: Test HotDocs Templates.

Using Dialogs

Creating and Using Dialogs

Introduction: Create Your Own Dialogs

When you insert variables in your templates, HotDocs automatically creates a *default dialog* (or a question window) for each variable. When the user assembles the document, HotDocs presents each of these individual dialogs in the order the variables are used in the template.

However, if you want to have more control over the information-gathering process, you can create custom dialogs. To do this, you group related questions together and add text (and other design elements) to the dialog to help the user provide correct answers. You can also control the order in which custom dialogs appear during the interview.

When you add a variable to a dialog, HotDocs creates a link between the variable and the dialog. When an answer is needed for the variable during the interview, HotDocs displays the linked dialog instead of displaying the variable by itself.

When creating dialogs, you can add visual elements to the dialog, such as lines, graphics, and white space. You can also include links to computations, applications, or even a Web page. All of these tools give your users a better experience answering questions in the dialog.

The following shows the Dialog Editor with a list of the variables that have been linked to it.

Using Dialogs

🚍 Employee Information - Dialog Editor - Demo Employment Agreement.cmp 🛛 💌			
Properties Options Script Locals Resource	e Layout Used In Notes		
Dialog <u>n</u> ame:	<u>S</u> tyle:		
Employee Information	Regular 🔻		
Title:			
<u>C</u> ontents:	Com <u>p</u> onents: Available Components 🔹 🎽		
A Employee Name	Agreement Information Job Information Salary and Benefits White Space Agreement Date Agreement Date Annual Salary Company Representative Employee to Complete Trial Period Employee to Receive Paid Seminar D Employment Status Hourly Salary		
	▲ Job Description ▲ Job Title ■ Find: ▼		

During the interview, HotDocs displays the dialog and allows the user to enter answers.

📋 Employment Agreement - HotDocs [leveloper	
<u> </u>	lelp	
🗐 🛛 New Answer File	• 🖄 🖆 🖪 🚽 📶 💙 ኞ 🐼 🔞	
∠ n n % b & b ≥ b	·∃+ ≜↓ 🖉 🚈 🖅	
Interview Document Preview	Question Summary Answer Summary	
Employee Information	Employee Information	
Agreement Information	Employee Name	
Salary and Benefits	Alice Chamberlain	
End of Interview	Employee Gender	
	Male Female	
	M First M Previous Next Met Las	t 🎽 Fini <u>s</u> h 📄
		<u>^</u>
	,	
		NUM

You can also add scripts to your dialogs, which let you conditionally hide or dim variable questions and require that certain questions be answered. You can use text from a Windows or HTML Help file, Folio Infobase, custom program, or Web page as a resource for a dialog. You can also create an answer source—an answer file from which a user can select existing answers during the interview. Finally, you can change the width and placement of answer fields in the dialog.

For more information on using the dialog editor follow the links below:

- At a Glance: The Dialog Editor (Properties Tab)
- At a Glance: The Dialog Editor (Options tab)
- At a Glance: The Dialog Editor (Script tab)
- At a Glance: The Locals tab (Variable/Dialog Editor)
- At a Glance: The Resource tab (Variable/Dialog Editor)
- At a Glance: The Dialog Editor (Layout tab)
- At a Glance: The Used In tab (Variable/Dialog Editor)
- At a Glance: The Notes tab (Variable/Dialog Editor)

At a Glance: The Dialog Editor (Properties Tab)

🖃 New Dialog - Dialog Editor - Demo Life Insurance Application.cmp		
Properties Options Script Locals Resour	ce Layout Used In Notes	
Dialog <u>n</u> ame:	<u>S</u> tyle:	
	Repeated Serie 🖪 🛛 👻	
Title:		
Add button text:	Series lab <u>e</u> l:	
Contents:	Components: Available Components 🔻 🎽	
F	Authorization Beneficiary Designation Coverage Information Employee Information Medical History G Medical History G Medical History Amount of Coverage Beneficiary Name Deserveduation	
🚹 📾 A 2 🛐 🕌 🗮 🔲	■ Fin <u>d</u> : K -	
Test Update	OK Cancel Sa <u>v</u> e	

You can open the **Dialog Editor** from the **Component Manager** or while editing a template by creating a new dialog or opening an existing dialog.

In the first text field A you can enter the dialog name. In the drop-down list **b** to the right of the **Dialog Name** field you can choose the style of your dialog from the following list:

- Regular
- Repeated Series
- Spreadsheet
- Spreadsheet on Parent

The option you choose here changes the next couple of options. If you select **Regular** you will only see the **Title** field c where you can enter a title for your dialog. If you select **Repeated Series** you will see the **Title** field c then below it you will have the options to add button text **D** and label the series (see image above). If you select **Spreadsheet** or **Spreadsheet on Parent** you will see the **Title** field c and the

Add Button Text field D, but instead of the Series Label field E you will have the option to select how many rows of the spreadsheet you would like HotDocs to display and if you would like to hide the spreadsheet buttons.

This feature will alter the amount of spreadsheet rows displayed in the interview, it will not stop the user from entering information in further rows. If they do this, the user can view the extra rows by using the scroll bars on the spreadsheet. To reduce the amount of rows that the user can enter information into see **LIMIT NUM**.

The **Rows to display** feature is not compatible with templates published for **HotDocs Server**. Any alteration you make to this field will not be visible in the **Server** interview.

Regardless of what option you have selected the rest of the window will remain the same. On the left is the **Contents** field where you can see what components are in our dialog. On the right is the **Available Components** field where you can see all the components in the Component File for this template. You can drag components from this field into the **Contents** field to have them in this dialog; any component with a faded symbol is already assigned to another dialog. At the top of the **Available Component** field is a drop down list that you can use to filter the displayed components and a button you can click to create a new component with

At the bottom left of the window you have a small toolbar \mathbf{H} where you can generate new components, dialogs and dialog elements by clicking on the component type icon:

- 🔊 Dialog Element
- A Text Variable
- ² Number Variable
- I Date Variable
- False Variable
- **E**Multiple Choice Variable
- Dialog

You can also use the **Find** field \mathbf{K} on the bottom right to search the **Available Components** field.

More dialog options are available in the other tabs: Options, Script, Layout, Used In and Notes.

To learn more about using the Dialog Editor follow the links below:

- Introduction: Create Your Own Dialogs
- Gather Questions into a Custom Dialog
- Edit a Custom Dialog
- At a Glance: The Dialog Editor (Options tab)

- At a Glance: The Dialog Editor (Script tab)
- At a Glance: The Locals tab (Variable/Dialog Editor)
- At a Glance: The Resource tab (Variable/Dialog Editor)
- At a Glance: The Dialog Editor (Layout tab)
- At a Glance: The Used In tab (Variable/Dialog Editor)
- At a Glance: The Notes tab (Variable/Dialog Editor)
- Tips on Naming Your Variables

At a Glance: The Dialog Editor (Options Tab)

🖃 New Dialog - Dialog Editor - Demo Employment Agreement.cmp		
Properties Options Script Locals Resource Layout Used In Notes		
Variables		
Selection grouping: None \triangle \neg \square None of the above		
Prompt position: Automatic 🚯 🔻 🗹 Right-align prompts Maximum units: 12 🛁 🧿		
Interview		
Prompt to use when displayed as <u>c</u> hild dialog:		
E I Show <u>b</u> uttons for child dialogs		
✓ Prevent this dialog from being asked when irrelevant		
Trim endmost iterations whose answered variables are grayed or hidden		
Always stop at this dialog when moving to next unanswered		
Advanced		
■ Sk automatically		
Link variables to this dialog		
Answer source: G Map Variables		
Test Update OK Cancel Save		

You can open the **Options** tab of the **Dialog Editor** from the **Component Manager** or while editing a template by creating a new dialog or opening an existing dialog then clicking on the **Options** tab.

The first section of this tab allows you to set options for the variables on the dialog. Using the first dropdown list A you can choose whether True/False variables on this dialog should exhibit "grouped" behavior

or not. behave as **Select One** grouping causes True/False variables to be displayed radio buttons, while **Select All That Apply** grouping causes them to be displayed as check boxes. Turning on grouping also allows you to add a **None of the above** option to each group of True/False variables. Using the next drop-down list **B** you can set the options for the variable prompts on the dialog. You can select from **Above**, **Left**, or **Automatic** (where HotDocs will choose the style it thinks most appropriate. If you select **Left** or **Automatic** then the check box and number field **C** to the right ungray and you will be able to choose whether you would like the prompts to align right and, by editing the number in the field **C**, how wide the prompt can be.

The second section of the tab allows you to set options for the Interview. If this dialog is to be used as a child dialog during the interview it can have its own prompt to help the user. You can enter that in the text field **D**. Below that are four check boxes **E** where you can set the following options:

- Show buttons for child dialogs
- Prevent this dialog from being asked when irrelevant
- Trim endmost iterations whose answered variables are grayed or hidden.
- Always stop at this dialog when moving to next unanswered.

The final section allows you to set advanced options for this dialog. Using the two check boxes **F** you can decide if the dialog should be Asked Automatically and if HotDocs should Link variables to this dialog. Below that is the option **G** to assign an answer source to this dialog (see Suggest an Answer Source for Dialogs for more information). You can select one from the drop-down list or by clicking the Browse button. Once you have selected an answer source you will be able to click the Map Variables button. You will be taken to the Variable Mapping dialog box where you can map variables from the answer source to variable in the dialog (see At a Glance: The Variable Mapping dialog box for more details).

To learn more about using the Dialog Editor follow the links below:

- Change a Dialog's Options
- Control Whether Dialogs are Asked Automatically
- Use the Same Variable or Clause in Two or More Dialogs
- At a Glance: The Dialog Editor (Properties Tab)
- At a Glance: The Dialog Editor (Script tab)
- At a Glance: The Locals tab (Variable/Dialog Editor)
- At a Glance: The Resource tab (Variable/Dialog Editor)
- At a Glance: The Dialog Editor (Layout tab)
- At a Glance: The Used In tab (Variable/Dialog Editor)
- At a Glance: The Notes tab (Variable/Dialog Editor)

At a Glance: The Dialog Editor (Script Tab)

🖃 New Dialog - Dialog Editor - Demo Employment Agreement.cmp		
Properties Options Script Locals Resource	Layout Used In Notes	
Script: 🔼 🕅 🖛 🖝 👉 🛍 🛍 🕅	🛱 💩 🖙 🖅 ோ 🖅 🖉 🖉	
B		
Components: Variables in Dialog 🔹	= Instruction models: != CONCEAL VAR CONCEAL VAR DEBUG	
Θ	Expression models:	
Fin <u>d</u> :	>= Enter a Date + Enter a Number - Enter some Text * Enter True or False	
Test Update	<u>QK</u> Cancel Sa <u>v</u> e	

You can open the **Script** tab of the **Dialog Editor** from the **Component Manager** or while editing a template by creating a new dialog or opening an existing dialog then clicking on the **Script** tab.

At the top of this tab is the **Script Toolbar** \underline{A} where you can choose from the following buttons to search and edit the **Script** field \underline{B} :

- **Auto Format:** Indents matching pairs of IF and REPEAT instructions, based on the level of their insertion.
- **• Undo:** Removes any changes you have made to the script.
- **Redo:** Reapplies any changes you have made to the script.
- **Cut:** Removes the selected text and copies it to the Clipboard.
- **Copy:** Copies the selected text to the Clipboard.
- **Paste:** Pastes the selected text on the Clipboard at the cursor position in the script.

- **# Find:** Displays the Find dialog box where you can specify the word or text string for which you are searching.
- **B** Find Next: Finds the next instance of the word or text string for which you are searching.
- text string and replace: Searches for a specific word or text string and replaces it with word or text string you specify.
- **Go To:** Displays the Go To dialog box where you can specify the location in the script to which you want to move your cursor. You can go to either a specific line or character in the script.
- **Findent Block:** Indents the selected text.
- **©Outdent Block:** Realigns the indented text with the left margin of the scripting box. (If the text has been indented more than once, realigns the text with the previous tab stop.)
- **Comment Block:** Causes the selected text to become inactive, meaning HotDocs won't process it when it processes the computation. Often, template developers "comment out" scripting if they need to close the scripting dialog box but HotDocs won't let them because the script is invalid. Additionally, developers often add personal explanations to the scripts they are writing, and commenting them out keeps the computation operational.
- **Uncomment Block:** Removes any commenting that has been applied to the selected script. Once the script is uncommented, HotDocs will attempt to process it.
- **Options:** Displays the HotDocs Options dialog box where you can customize the way script editing features work.
- **Whelp:** Opens the relevant page of the HotDocs Help File.

You have the option of writing directly in the **Script** field **B**, using your mouse to drag variables, operators, and models from the lists below to this box, or you can press Ctrl+Space to have HotDocs display these lists directly in the script so you can use the keyboard to choose the syntax item you need.

On the left is the Component menu showing a list of all the components in this template. You can search it using the find field at the bottom, filter it by selecting a component type from the drop-down list at the top or create a new component by clicking on the **New Component** button. In the centre is a list of useful scripting operators, at the top right is a list of Instruction models , and at the bottom right is a list of Expression models.

To access help for each instruction or expression, first select the model in one of the lists and then press **Ctrl+F1**.

For further information on how to use the operators, see Use Operators when Scripting, and for further information what the Instruction and expression models do, see Full List of Instruction Models and Full List of Expression Models.

To learn more about using the Dialog Editor follow the links below:

- Use Scripts to Add Power to Your Dialogs
- Special Instructions for Dialog Scripts

- At a Glance: The Dialog Editor (Properties Tab)
- At a Glance: The Dialog Editor (Options tab)
- At a Glance: The Locals tab (Variable/Dialog Editor)
- At a Glance: The Resource tab (Variable/Dialog Editor)
- At a Glance: The Dialog Editor (Layout tab)
- At a Glance: The Used In tab (Variable/Dialog Editor)
- At a Glance: The Notes tab (Variable/Dialog Editor)

At a Glance: The Dialog Editor (Layout Tab)



You can open the **Layout** tab of the **Dialog Editor** from the **Component Manager** or while editing a template by creating a new dialog or opening an existing dialog then clicking on the **Layout** tab.
In the main section A of this tab you can see all the components in the dialog and by dragging and dropping them you can change what order they appear in the interview. You can also arrange up to three components side by side so they show up on the same row in the interview.

To learn more about using the Dialog Editor follow the link below:

- Change the Layout of Variables in a Dialog
- At a Glance: The Dialog Editor (Properties Tab)
- At a Glance: The Dialog Editor (Options tab)
- At a Glance: The Dialog Editor (Script tab)
- At a Glance: The Locals tab (Variable/Dialog Editor)
- At a Glance: The Resource tab (Variable/Dialog Editor)
- At a Glance: The Used In tab (Variable/Dialog Editor)
- At a Glance: The Notes tab (Variable/Dialog Editor)

Gather Questions into a Custom Dialog

By default, when HotDocs displays variables in a HotDocs interview, each variable is in a dialog by itself. This dialog is called a default dialog. While this approach is sufficient for some situations, you can also group related variables and present them in a custom dialog. Grouping questions together makes it easier for users to answer interview questions because there are fewer dialogs to navigate through. Additionally, sometimes viewing questions in context makes it easier to understand what information is required.

To add variables or clauses to a custom dialog

- 1. At the HotDocs template, click the **Component Manager** button. The **Component Manager** window appears.
- 2. Select **Dialogs** from the **Components** drop-down list. This limits the list of components to show only the dialogs used in the template.
- 3. Click the **New Component** button. The **Dialog Editor** appears.
- 4. At the **Properties** tab, type a name in the **Dialog name** field.
- 5. Select the components you want from the **Components** list and drag them to the **Contents** field.

Click the **Components** drop-down button and select a component type to limit the list of components you are viewing. Ye use a component you haven't created yet, select a component type and click the **New Component** button.

- 6. Optionally, to change the order variables are asked in the dialog, select a variable and drag it up or down in the list. As you do this, HotDocs displays a horizontal bar showing you where the variable will be placed when you release the mouse.
- 7. Once all the variables are added, you can perform any of the following optional tasks:

То	Do This
Specify a prompt for the dialog that is different from the dialog name	Type the desired prompt in the Title field. The title is what will be used in the interview outline and in the dialog title bar during the interview.
	Dialog titles are useful when the dialog name you have assigned wouldn't make sense to the user.
	To make the title bold, italic, or underlined, insert the opening dot code for the style you need. Do not include a closing dot code.
Add other items to a dialog, such as text, graphics, links, or design elements	Click on the Create New Dialog Element icon and drag it to the Contents list. The Dialog Element Editor opens where you can choose to add text, lines or spaces, a link that runs a computation, a link that launches an application, a hyperlink, or a graphic.
Choose how you want to display a dialog if it's repeated	Click the Style drop-down button and choose a repeated display option: Repeated Series or Spreadsheet .
	To assign a label to the entire repeated series of dialogs, enter the label in the Label field.
Customize how the dialog displays variable questions and control how the dialog functions	Click the Options tab and make your selections.
Make the contents of your dialog dynamic based on answers a user enters	Click the Script tab and create a script.
Provide helpful information about the dialog for the user	Click the Resource tab and enter the resource text. The information you enter here appears in the resource pane during the interview.
Have two or more variables appear side by side in the dialog	Click the Layout tab and drag variables to the line you want.

View a list of all components that use this dialog	Click the Used In tab. (See View Relationship Between the Current Component and Other Components.)
Enter notes about the component, such as an explanation about why the component was created or how it should function in the interview	Click the Notes tab and enter your comments.
See how your dialog looks and works without closing the Dialog Editor	Click Test. A test assembly window appears, showing the current dialog. If you want to make changes to the dialog, make your changes in the Dialog Editor (without closing the assembly window) and click Update . When you are satisfied with your changes, close the test assembly window and click OK .

Once your dialog is created, you can rearrange components within the dialog by dragging them up or down in the **Contents** list. You can remove any component by dragging it back to the **Components** list. To edit a component, double-click it at either the **Contents** field or the **Components** list.

By default, HotDocs asks variables and dialogs in the order it "reads" them in the template (top-tobottom, left-to-right). To control when a dialog appears in the interview, you must use an ASK instruction at that point in the template.

When you add variables to a dialog, the variables and the dialog become linked and used variables are grayed out in the **Components** list of the **Dialog Editor**. To use the same variable or clause in two or more dialogs, you must clear the **Link variables to this dialog** option (on the **Options** tab) or the variable is removed from the first dialog.

If you don't link variables to a dialog, the variables will appear in their own default dialogs in the interview. If you assign titles to the variables, that title will be used in the interview outline and in the dialog title bar. Otherwise, the variable prompt or name will be used.

You can control the width of answer fields in a dialog.

To save your changes to a dialog without closing the Dialog Editor, click **Save**.

To move components between the **Components** and **Contents** list using the keyboard, use the **Left** and **Right** arrow keys. To move components within the **Contents** list, press **Alt+Up Arrow** or **Alt+Down Arrow**.

Edit a Custom Dialog

Once you have created a dialog, you can use Component Manager to make changes to it at any time.

To edit a dialog

- 1. At the HotDocs template, click the **Component Manager** button. The **Component Manager** window appears.
- 2. Select **Dialogs** from the **Components** drop-down list. This limits the list of components to show only the dialogs used in the component file.
- 3. Select the dialog you want to edit and click the **dialog Edit Component** button. The **Dialog Editor** appears, showing the contents and properties of the dialog.
- 4. Make changes to the dialog. (See Change a Dialog's Options for ideas.)

To edit a dialog without using Component Manager, make sure your cursor isn't in a variable field. Then, click the **definite Component** button in the HotDocs toolbar. Select **Dialog** and click the **Component** drop-down button. Choose the dialog you want to edit from the list.

Change a Dialog's Options

You can make several changes at the **Options** tab of a custom dialog:

- Put True/False variables, clauses, or inserted dialogs into single-selection or multiple-selection groups
- Decide where variable prompts appear in relation to answer fields
- Make sure variables grouped in a dialog are asked only if the dialog is specifically ASKed
- Make sure a dialog always appears, no matter how the user navigates through an interview
- Specify an answer source to be used with the dialog

To change dialog options

- 1. Open a dialog to edit.
- 2. Click the **Options** tab. The window shows several custom options.
- 3. Perform any of the following tasks:

То	Do This	
Place groups of True/False variables together, clause components together, or child dialogs together so users can choose only one option	From the Selection grouping drop-down list, choose Select One. When HotDocs processes a dialog that has a selection grouping, it marks all options as <i>false</i> , which tells HotDocs those variables are answered. (Of course, the user can change this by selecting an option.) The user viewing the dialog, however, sees no visual cue that these options are false (and therefore answered), so they might not understand why the answered status changes in the interview outline. To avoid confusion, always select None of the Above , which adds a <i>None of the Above</i> option to the list. This way, the dialog can appear answered, even if the user does not select an option.	
Place groups of True/False variables together, clause components together.	From the Selection grouping drop-down list, choose Select All That Apply.	
or child dialogs together so users can choose from several options	marks all options as <i>false</i> , which tells HotDocs those variables are answered. (Of course, the user can change this by selecting an option.) The user viewing the dialog, however, sees no visual cue that these options are false (and therefore answered), so they might not understand why the answered status changes in the interview outline. To avoid confusion, always select None of the Above , which adds a <i>None of the Above</i> option to the list. This way, the dialog can appear answered, even if the user does not select an option.	
Add a None of the Above option to a	Select None of the Above.	
single-selection or multiple-selection group	Providing a <i>None of the Above</i> option for your users gives them the chance to answer a question without selecting any of the options you have presented.	
Choose the visual relationship you want	Select an option from the Prompt position drop-down list:	
between variable prompts and answer fields	 Above places the prompt above the answer field. Left places the prompt to the left of the answer field. The length of the prompt text can be from 12 to 50 units, with one unit being equal to the width of the character <i>5</i>, as specified in the Maximum units field. If the prompt length exceeds the maximum units setting, the prompt will wrap to the next line. Automatic places the prompt according to the values you specify in the Maximum units field. Prompts that are shorter 	

Prompts that are longer are placed above the answer field.

To align	prompts to	the right,	select I	Right-align	prompts.
re angri		the ngin,	501000		P. C. P. C.

When you place variables side-by-side in a dialog, prompts will always appear above the answer field.

Enter the new prompt in the **Prompt to use when displayed as child dialog** box.

The prompt you specify here appears in the dialog only. The dialog's name or title is still used in the interview outline and in the dialog itself.

Specifying a prompt lets you provide more instructive text for accessing and then answering questions in the child dialog.)

Enter a prompt for a

the parent dialog

(By default, child dialogs appear on

child dialog button on

parent dialogs as links with the child dialog's name as the prompt.

Keep child dialogs from appearing as buttons on the parent dialog

Keep the dialog from being asked if none of the variables in the dialog are active (or used by the document)

Make sure the dialog is asked—even if it has already been answered—when a user clicks the Next Unanswered button

Control whether dialogs are asked automatically

Cause variables in the dialog to be asked

Deselect **Show buttons for child dialogs**. HotDocs removes any representation of the child dialog from the parent dialog. The only reference to the dialog will be in the interview outline, and users must manually click on the dialog to answer questions within it.

Select Prevent this dialog from being asked when irrelevant.

This option is only available when the component file property **Automatically disable irrelevant interviews and dialogs** has been turned off.

Select Always stop at this dialog when moving to next unanswered.

This option is also useful when you have a dialog that contains dialog element text only. By selecting this option, you can ensure that users navigating the interview using the **Next Unanswered** button will still see this dialog.

Clear Ask automatically.

Clear **Link variables to this dialog**. If this option is cleared and the dialog is not specified in an ASK instruction, Hotdocs asks the variable questions individually.

separately, rather than together in a dialog	
Use one variable in two or more dialogs	Clear Link variables to this dialog.
Identify an answer source file from which the user can choose answers	Either type the name of an answer source file in the Answer source field, or click the drop-down button and select an existing answer source.
	Click the Map Variables button to link an existing answer source to a dialog where the variable names are not the same.
	You can also link an answer source to a Time Matters Record or an Outlook Contacts list.
Display all repeats even if graved or hidden	Clear Trim endmost iterations whose answers are grayed or hidden .
	This option is only available on repeated dialogs.

Add Resource Information to a Variable or Dialog

Sometimes including additional information about questions users are answering during an interview can make it easier for them to enter the correct answer. To provide this extra information, you can assign resources either to variables or to dialogs. Users can view this information by viewing the **Resource** pane during the interview.

By default, resources are displayed as plain text. However, you can use any of the following other formats or programs to create and display resources. (Click the associated link for instructions on creating the resource type.)

- WinHelp
- HTML Help
- Folio infobase
- Custom program
- Web page

To add a plain text resource to a variable or dialog

- 1. Open the variable or dialog to which you want to assign resources. (See Edit a Variable or Edit a Custom Dialog.)
- 2. Click the **Resource** tab.
- 3. Click the **Resource type** drop-down button and select **Plain Text**.
- 4. Type the information in the **Text** field.
- 5. Optionally, to use variables in the resource or make text in the resource conditional, enter the variable name, or enter an IF instruction or expression. (See Use Variables and Scripts in Prompts, Dialog Element Text, and Resources.)
- 6. Optionally, select **Pop-up** to display the information in a What's This? help-style window, rather than a typical window. (The pop-up window only appears when the user clicks the **Resource** button next to the answer field.)

During assembly, HotDocs displays the resource text either in the resource pane or in a text-only pop-up window, depending on your selection. (Pop-up windows stay open only until the user clicks somewhere else.)

You cannot use angle brackets with text between them in a plain-text resource (for example, **<Your Name>**). Instead, you must use the codes **<**; and **>**; to insert the brackets (for example, **<Your Name>**). (You can use brackets, however, if you put a space immediately after the opening angle bracket and immediately before the closing angle bracket, like this **< Your Name >**.)

When editing Multiple Choice variables, you can assign resource text to the entire variable or to individual options. At the **Multiple Choice Variable Editor**, click the **Resource** tab and make your selection at the **Resource for** drop-down list.

Change the Layout of Variables in a Dialog

You can change the appearance of dialogs, by, for example, including placing variables side by side in the dialog and changing the width of variables already placed side by side. Changes can only be made in dialogs that have the **Regular** and **Repeated Series** styles applied. You cannot change the layout of **Spreadsheet** dialogs.

To place variables side by side on a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor.
- 2. Click the **Layout** tab. The window changes to show icons representing each component in the dialog.
- 3. Click a component icon and drag it to another location in the window.

When you do this, HotDocs displays either a vertical or a horizontal bar that indicates where the variable will be placed when you release the mouse button. A vertical bar shows that the variable will be placed to the right or left of variables already on that line. A horizontal bar shows that the variable will be placed above or below that line. You can place up to three variables on the same line.

 Optionally, click **Test**. HotDocs displays the variables in a test assembly window. (You can leave this window open and return to the **Dialog Editor** to make additional changes, which are immediately updated in the window.)

If you place only two components on the same line, you can make one of the components wider than the other by right-clicking the component and selecting **Wide** from the shortcut menu. To further adjust the width of answer fields, at the **Variable Editor**, click the **Advanced** tab and make your changes.

To edit a component or remove it from the dialog, right-click the component and select **Edit** or **Remove** from the shortcut menu.

When two or more components appear on the same line, the prompts appear above the answer field. You cannot specify a different prompt alignment.

Add Notes to Components

As you create components, you may want to include notes about the component. For example, maybe you need to include information about restrictions you applied to a Number variable, or maybe you need to explain the purpose of the variable or dialog or how it should be used in the template. To do this, you can add a note to the component. You add notes at the **Notes** tab of the component editor. Notes can be viewed by anyone editing the component.

You can add a comment to a variable field or instruction field so it can be visible by anyone editing the template text. See Add a Comment to a Variable or Instruction Field for details.

To add development notes to a component editor

- 1. Edit the component. (See Edit a Variable or Edit a Custom Dialog.) The component editor appears.
- 2. Click the **Notes** tab. The view changes to show a multi-line text field.
- 3. Enter the notes you want associated with the component.

View Relationship Between the Current Component and Other Components

When editing a component (such as a variable or a dialog), you can view a list of other components in the template that refer to it. You can view this list at the **Used In** tab of the component editor.

By default, when you first click the **Used In** tab, HotDocs displays only the dialog or database component to which the variable is linked, as well as any other dialogs in which the variable is used. To generate a complete list of component references (including other components that refer to the current component), you can click the **List All Components** button. If you want a complete list to be displayed every time you click the **Used In** tab, you must select **Always list all components when viewing this tab.** (Depending on the complexity of your template, you may experience a slight delay each time you view this list, which is why displaying it is optional.)

To view component relationships

- 1. At the template or at Component Manager, select the component and edit it. (See Edit a Variable.) The **Component Editor** appears.
- 2. Click the **Used In** tab. The window changes to show a box listing the dialog(s) (and database components) in which the component is used.
- Optionally, to view a list of other components that refer to this component, click List All Components. HotDocs creates a list of all components that refer to this particular component. (Depending on the complexity of your template, this may take several seconds.)
- 4. Optionally, to always view this list of component cross-references each time you view the **Used In** tab (regardless of the component you are editing), select **Always list all components when viewing this tab**.

The dialog to which the variable is linked is marked with a red arrow. Dialogs that simply use the component are marked with a plain arrow. See Use the Same Variable or Clause in Two or More Dialogs.

If you make changes to other components in the component file while you have the component editor open, you can click **List All Components** again to regenerate the list of used components.

Use the Same Variable or Clause in Two or More Dialogs

When you add a variable to a dialog, HotDocs creates a link between the two so that when HotDocs processes the variable during the interview, it knows to display the dialog to which the variable is linked. However, this linking limits your ability to use a variable in more than one dialog. To use a variable in two different dialogs, one of the dialogs must not be linked to its variables.

One reason you may want to include variables in two or more dialogs is so you can have two different versions of a dialog, only one of which should be displayed, based on answers a user enters. Otherwise, when you add the variable to the second dialog, it is *automatically* removed from the first dialog. This is

because a variable can be linked to only one dialog. You can, however, specify an option that allows the variables and clauses in that dialog to be used in another dialog.

To use the same variable or clause in two or more dialogs

- Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor. (At the Components list of the Dialog Editor, variables and clauses that are already linked to other dialogs are grayed out.)
- 2. Click the **Options** tab. The window changes to show several custom options.
- 3. Clear Link variables to this dialog.

If you clear this option for a dialog, you must specifically ASK the dialog in the template or else it will not be displayed during the interview. (See Control Whether Dialogs are Asked Automatically.)

Control Whether Dialogs are Asked Automatically

You can control whether a dialog is asked automatically during an interview. You can do this by selecting an option at the **Options** tab of the **Dialog Editor**.

Dialogs that are *not* asked automatically must be asked using an ASK instruction. (See Control When Your Dialogs Appear.)

By default, HotDocs asks a dialog automatically: 1) when the dialog has not been asked before in the interview, and 2) when a variable to which the dialog is linked is asked automatically. Also, a dialog will be asked automatically when it is used in a REPEAT instruction. (See How Variables and Dialogs Are Asked for details on how variables and dialogs are asked.)

One example of why clearing this option may be useful is when you've created a list of answers by merging two different lists together (see WHILE EXPRESSION; END WHILE). When doing this, you often need to merge the new list into the document using a third repeated dialog. Since you've already gathered all the applicable answers, you don't want this repeated dialog to be asked.

To specify an option that controls when a dialog is asked

- 1. Edit a custom dialog. (See Edit a Custom Dialog.) The **Dialog Editor** appears.
- 2. Click the **Options** tab. The window changes to show several custom options.
- 3. Clear Ask automatically.

This option replaces the need for using ASK NONE / ASK ALL instructions. However, instructions for using them are still available.

Inserting Dialog Elements in Dialogs

At a Glance: The Dialog Element Editor

Properties Options Use	ed In Notes	
Dialog element <u>n</u> ame:		_ 8
		Generate name automatically
Style		
Iext	O Script link	© Image
🔘 <u>H</u> orizontal divider	👩 🔘 Application link	
Vertical spacing	🔵 <u>W</u> eb link	
Display text:		
		*
0	0	
		*
	OK	Canaral Saura

You can open the **Dialog Editor** from the **Component Manager** or while editing a template. From there you can choose to add a new dialog element by clicking the **Selement Dialog Element** button. This opens the **Dialog Element Editor**. You can also access this window by double clicking on an existing dialog element at the **Component Manager**.

In the first text field A you can enter a name for your new dialog element or you can check the box B to the right of this to have HotDocs generate a name automatically.

Below this is a set of check boxes C where you can choose what style of dialog element you need to make. You can choose from:

- **Text:** Allows you to include additional information on the dialog to help the user answer the questions correctly.
- **Horizontal divider:** Allows you to insert a horizontal line on the dialog to separate questions in the dialog. If needed, you can also include text with the separator to better explain what's required in the following section.

- **Vertical spacing:** Allows you to place empty space in the dialog to help you better define "sections" of a dialog.
- **Script link:** Allows you to place a link on the dialog that, when clicked, executes a computation script. This may be useful if you want users to perform a computation or other type of command and have the result immediately take effect.
- **Application link:** Allows you to place a link on the dialog that, when clicked, launches an application. This may be useful if your users need to access other programs as they complete the interview.
- Web link: Allows you to place a link on the dialog that, when clicked, opens a Web page.
- Image: Allows you to place an image or picture on the dialog.

Depending on what you choose the section below **D** will show different options. If you select **Text** then you will see a text field where you can enter the text you wish to display.

stification: Left 🔻
1

If you select **Horizontal divider** (above) then you can choose to enter a caption to go with the divider, the percentage of the dialog font you would like the caption font to be and the justification of the cation on the horizontal divider.

Spacing: 100 🚔 % of dialog font

If you select **Vertical spacing** (above) then you can choose the size of the spacing by selecting a percentage of the dialog font.

Computation:	- 📔
Link text:	
	*
	*

If you select **Script link** (above) then you can enter the name of a computation variable in the first text field or use the **New Component** button next to it to create a new computation variable. In the second text field you can enter the text you wish to use as a link.

E <u>x</u> ecutable file:	
Link text:	
	*
	.*

If you select **Application link** (above) then in the first field you can type in the file path for your application or use the **Browse** button to navigate to it. In the second text field you can enter the text you wish to use as a link.

URL:	
Link text:	
	*
	.*

If you select **Web link** (above) then in the first field you can type in the URL you wish to use or click the **Browse** button to navigate to it. In the second text field you can enter the text you wish to use as a link.

Image <u>fi</u> le:			
Position			
Ieft	© <u>C</u> enter	© <u>R</u> ight	

If you select **Image** (above) then in the first field you can choose the image file you wish to use by either entering the file path, choosing from the drop down list of recently used images or clicking the **Browse** button to the right of the field. Below this you can choose one of the three position options for the image.

More dialog element options are available in the other tabs:

- Options: Choose whether to display text in All views of the dialog, Spreadsheet view only, or Edit Row view only when using spreadsheet dialogs.
- **Used In:** See a list of other components that use this dialog element. (see At a Glance: The Used In tab (Component/Dialog Editor) for more information
- **Notes:** Make development notes about the dialog element. (see At a Glance: The Notes tab (Component/Dialog Editor) for more information.

To learn more about using the Dialog Element Editor follow the links below:

- Add Text to Your Dialogs
- Add the Ability to Launch an Application from a Dialog

- Add a Link to a Dialog That Runs a Computation
- Add a Hyperlink to a Dialog
- Add Spacing and Separator Lines to a Dialog
- Add a Graphic File to a Dialog
- Tips on Naming Your Variables

Add Text to Your Dialogs

You can add text to a dialog to guide the user in answering questions. Dialog text can be inserted above or below variables or other items in the dialog.

To add text to a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor.
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Solution Dialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.
- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the **Style** group, ensure **Text** is selected.
- 5. In the **Display text** field, enter the text you want to appear in the dialog.
- 6. Optionally, to assign font properties to the prompt text (such as bold or underline), enter a formatting dot code.
- 7. Optionally, if the dialog is repeated as a spreadsheet, click the **Options** tab and choose how you want the spreadsheet to appear in the interview:
 - **Spreadsheet view only** displays the text on the dialog only when users view the spreadsheet and not when they edit individual rows in the spreadsheet.
 - **Edit Row view only** displays the text on the dialog only when users edit individual rows of the spreadsheet. No text will appear when they are viewing the spreadsheet.
- 8. Move the dialog element within the dialog by dragging it up or down in the **Contents** list.

To edit the dialog element, double-click it in the **Contents** list. To remove the element, drag it back to the **Components** list.

You can use a dialog to display just text and no variables. For example, you may want to include a dialog that contains only instructions for completing the interview.

You can customize dialog element text based on users' answers by including variables and other scripting instructions in the element text.

Add the Ability to Launch an Application from a Dialog

At times, you may want your users to launch a separate application while answering questions in a specific dialog. You can add an option to a dialog that lets users either click a hyperlink or click a button to start the application.

For example, perhaps your users may need to calculate a number before entering their answer in an interview. You can provide a link on the dialog that launches the Windows Calculator so they can calculate their answer.

To place the link on a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor.
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Pialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.
- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the **Style** group, select **Application link**.
- 5. In the **Executable file** field, click the **application**'s .EXE file.
- 6. In the **Link text** field, enter the text you want to use as the link.
- 7. Click the **Options** tab and select how you'd like the link to appear on the dialog, based on the following information:
 - In the **Display as** group, select whether the link should be a **Button** or **Hyperlink**. (If you select **Button**, you can specify a custom button size in the **Button size** group.)
 - If you select **Hyperlink**, select whether the link should include text, an image, or both in the **Display using** group. (Enter the name of the image file in the **Image file** field if you select **Image**.)
 - In the **Display in** group, select **Desktop interviews**. (Application links are not allowed in HotDocs Server interviews.)
- 8. Click **OK**. The **Dialog Element Editor** is closed.
- 9. Drag the component up or down in the **Contents** list, depending on where you want the hyperlink placed on the dialog.

When specifying the executable file, if you include just the program's file name and not a folder path, HotDocs looks for the executable file in the same folder as the template. If it doesn't find the file there, it looks in the HotDocs program folder. Finally, if the file is still not found, HotDocs uses information from the Windows Path Environment variable to search for it.

You may include command-line options in the file path of the executable, for example, "*C*:*Windows**Notepad.exe*" *c*:*Temp**MyTextFile.txt*.

Add a Link to a Dialog That Runs a Computation

You can add a link that processes a Computation variable to a HotDocs dialog. This may be useful if you want users to execute a calculation or other type of command and have the result immediately take effect.

To add a button to a dialog that runs a Computation variable

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor .
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Pialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.
- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the **Style** group, select **Script link**.
- 5. In the **Computation** field, enter the name of the Computation variable you want to run. (Click the **New Component** button to create or edit the computation.)
- 6. In the **Link text** field, enter the text you want to use as the link.
- 7. Click the **Options** tab and select how you'd like the link to appear on the dialog, based on the following information:
 - In the **Display as** group, select whether the link should be a **Button** or **Hyperlink**. (If you select **Button**, you can specify a custom button size in the **Button** size group.)
 - If you select **Hyperlink**, select whether the link should include text, an image, or both in the **Display using** group. (Enter the name of the image file in the **Image file field** if you select **Image**.)

Any images displayed in a browser-based (HotDocs Server) interview must be located in the same folder as the template. Also, bitmap (.BMP) files are not supported in browser-based interviews. Additionally, Silverlight does not support all possible color depths that are included in the PNG specification. Specifically, gray scale and 64-bit true color are not supported in Silverlight.

- In the **Display in** group, select whether the link should appear when the dialog is displayed in **Desktop interviews**, **Server interviews**, or both.
- 8. Click **OK**. The **Dialog Element Editor** is closed.
- 9. Drag the component up or down in the **Contents** list, depending on where you want the hyperlink placed on the dialog.

Add a Hyperlink to a Dialog

You can include a hyperlink on a dialog that can open a URL in a Web browser. This may be useful if you need users to refer to a specific Web page when answering questions in a dialog.

You may want to add a hyperlink to a dialog to let HotDocs Server users retrieve answers from a database. For example, the site administrator creates a Web page that presents information from a database, allowing users to select the appropriate data. Once the user selected the data, the Web page closes and the answers from the database fill the fields in the interview dialog. If you create such a Web page, it will only be accessible in HotDocs Server.

To add a hyperlink to a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the **Dialog Editor**.
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Pialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.
- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the **Style** group, select **Web link**.
- In the URL field, enter the complete Web address for the page you want the hyperlink to open. (Make sure the URL is entered correctly. Include http:// or https:// or ftp://, depending on the type of URL it is.)
- 6. In the Link text field, enter the text you want to display as the hyperlink.
- 7. Click the **Options** tab and select how you'd like the hyperlink to appear on the dialog, based on the following information:
 - In the **Display as** group, select whether the link should be a **Button** or **Hyperlink**. (If you select **Button**, you can specify a custom button size in the **Button size** group.)
 - If you select **Hyperlink**, select whether the link should include text, an image, or both in the **Display using** group. (Enter the name of the image file in the **Image file** field if you select **Image**.)

Any images displayed in a browser-based (HotDocs Server) interview must be located in the same folder as the template. Also, bitmap (.BMP) files do not work in browser-based interviews. Additionally, Silverlight does not support all possible color depths that are included in the PNG specification, specifically, gray scale and 64-bit true color.

• In the **Display in** group, select whether the link should appear when the dialog is displayed in **Desktop interviews**, **Server interviews**, or both.

8. Optionally, to control the appearance of the browser window when it opens, enter the JavaScript parameters in the **Window.open features string (HotDocs Server only)** field.

This value corresponds to the third parameter of the JavaScript window.open method (sFeatures). For example, to open the resource in a window that is 200 pixels high, 400 pixels wide, includes the status bar, and does not include the toolbar, menu bar, and location bar, use this string:

height=200,width=400,status=yes,toolbar=no,menubar=no,location=no

- 9. Click **OK**. The **Dialog Element Editor** is closed.
- 10. Drag the component up or down in the **Contents** list, depending on where you want the hyperlink placed on the dialog.

Rather than create a hyperlink that opens a Web browser, you can create an e-mail link. To do this, include **mailto:** in the URL, followed by an e-mail address, like this: **mailto:publications@hotdocs.com**.

If your hyperlink is designed to allow HotDocs Server users to access a database via a Web page, you must select **Server interviews** in the **Display in** group. Note that the hyperlink will not work in a desktop interview.

Add Spacing and Separator Lines to a Dialog

You can add white space and/or separator lines to better organize a dialog.

You can control how much empty space appears, and you can include a caption with each separator. For example, if you have two groups of questions in a dialog—one about plaintiffs and the other about defendants in a case—you can add space to separate each group, or you can add a separator line with the text *Plaintiff Information* and *Defendant Information* on each separator line. When adding empty space to a dialog, you can control how much space will be used.

To add white space to a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor.
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Pialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.
- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the Style group, select Vertical spacing.

- 5. In the **Spacing** field, enter a percentage. (For example, if you want the space to be equal to the space taken by text in the dialog, enter **100** in the **Spacing** field. If you want it to be larger or smaller, enter the appropriate number in the field.)
- 6. Click **OK**. The **Dialog Element Editor** is closed.
- 7. Drag the component up or down in the **Contents** list, depending on where you want the white space placed on the dialog.

To add lines to a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the Dialog Editor.
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Dialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.
- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the **Style** group, select **Horizontal divider**.
- 5. Optionally, enter a short description or phrase in the **Caption** field.
 - In the **Font size** field, enter a font size for the caption, based on the percentage of the rest of the dialog's text size. (For example, if you want the caption to be the same size as the rest of the text in the dialog, enter **100** in this field.)
 - Click the **Justification** drop-down button and select to which side of the dialog the caption is aligned.
- 6. Click **OK**. The **Dialog Element Editor** is closed.
- 7. Drag the component up or down in the **Contents** list to the desired location.

To format the caption text (for example, to make the text bold), highlight the text, right-click, and then choose a formatting option from the **Character Format** shortcut menu.

Add a Graphic File to a Dialog

You can include images in your dialogs to help users understand interview questions or to display your company logo or other icon. The types of files you can add include .BMP, .JPG, .GIF, and .PNG.

Images with transparency may not appear correctly in dialogs.

To add an image to a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the **Dialog Editor**.
- 2. At the **Properties** tab of the **Dialog Editor**, click the **Dialog Element** button and drag it to the **Contents** list. The **Dialog Element Editor** appears.

- 3. Enter a name for the component in the **Dialog element name** field, or select **Generate name automatically**, which allows HotDocs to name the component.
- 4. In the **Style** group, select **Image**.
- 5. In the **Image file** field, click the **Rowse** button and locate the image file.

Any images displayed in a browser-based (HotDocs Server) interview must be in the same folder as the template. Also, .BMP files do not work in browser-based interviews. Additionally, Silverlight does not support all possible color depths that are included in the PNG specification, specifically, gray scale and 64-bit true color.

- 6. Optionally, select an alignment option: Left, Center, and Right.
- 7. Click **OK**. The **Dialog Element Editor** is closed.
- 8. Drag the component up or down in the **Contents** list, depending on where you want the graphic placed on the dialog.

Using Scripts to Make Dialogs Dynamic

Use Scripts to Add Power to Your Dialogs

You can create scripts for your dialogs that allow you to hide or dim variable questions, require certain questions be answered before the user can proceed to the next dialog, and assign predetermined values to variables. You can also use scripts to make variables within a dialog conditional, just as they are in the underlying template text.

Before creating a dialog script, see Special Instructions for Dialog Scripts as well as Understand the HotDocs Scripting Language. Dialog scripts should only change the appearance of variables in a dialog—they should not be used to compute answers.

To create a script for a dialog

- 1. Open a dialog for editing. (See Edit a Custom Dialog.) HotDocs displays the **Dialog Editor.**
- 2. Click the **Script** tab. The window changes to show scripting options.
- 3. Enter your dialog script. (See Use the Script Editor and Special Instructions for Dialog Scripts.)

Once you have written your script, you can make sure it works correctly by testing the dialog. The dialog appears just as it will during regular assembly. (See Test a Custom Dialog.)

At the **Script** tab, the **Components** list shows the components (including dialog element components) in the order they are used in the dialog. This makes it easier to see exactly which components you can use in the script.

Special Instructions for Dialog Scripts

Dialog scripts should only change the appearance of variables in a dialog — they should not be used to compute answers. A dialog script may be run any number of times, with no guarantee that an answer will only be computed once.

The following is a brief explanation of instructions that can be used in dialogs. In each of the following examples, either replace the placeholder **NUM** with a number value, or replace **VAR** with a variable name:

Instruction	Description
CONCEAL VAR	This instruction, which you use in a dialog script, keeps variables from appearing in the Select From Answer Source dialog box of an answer source.
DEFAULT VAR TO VALUE	This instruction suggests a value for a variable if the variable is unanswered.
ERRORTEXT	This instruction allows you to create an error message that can appear during the interview.
GRAY ALL; GRAY VAR; UNGRAY ALL; UNGRAY VAR	These instructions, which are used in a dialog script, control whether components in a dialog appear grayed or ungrayed, depending on answers a user enters. GRAY ALL dims all components in the dialog, while UNGRAY ALL makes all of the components active again. Likewise, GRAY VAR dims a single component, and UNGRAY VAR enables the component again.
HIDE ALL; HIDE VAR; SHOW ALL; SHOW VAR	These instructions, which are used in a dialog script, control whether the user is able to see variables in a dialog. The HIDE ALL instruction hides all variables in the dialog, while SHOW ALL reveals the variables again. Likewise, HIDE VAR hides a single variable, and SHOW VAR reveals the variable.
LIMIT NUM	The LIMIT instruction limits the number of times a dialog can be repeated. It is placed in the script of the dialog that must be limited to a specific number of repetitions.
OMIT VAR	The OMIT VAR instruction, which you use in a dialog script, keeps variables from appearing in the Edit Answer Source dialog box of an answer source.
REQUIRE ALL; REQUIRE VAR	The REQUIRE instruction requires users to answer questions in a dialog before they can advance to the next dialog in the interview.

SET VAR TO VALUE

This instruction lets you specify a given value for a variable's answer automatically, rather than allow the user to specify an answer. With the SET instruction, you can transfer names and other values from one variable to another.

Assigning Answer Sources to Dialogs

Suggest an Answer Source for Dialogs

You can suggest an answer source (a list of answers for the user to choose from) for a dialog. Users open the list when answering questions in an interview and pick an existing set of answers instead of entering them manually. For example, if you have a large number of attorneys in your firm. Rather than enter a specific attorney's information each time you need to assemble a document, you can enter this information once and then retrieve it on a document-by-document basis.

An answer source file contains sets of answers for one or more variables in a dialog. For example, a dialog may require information about an attorney, such as a name, address, and telephone number. If there are multiple attorneys in a firm, all of this information for each attorney could be saved in an answer source. During an interview, the user opens the answer source, selects the appropriate attorney, and that information will be merged into the document.

You must group all of the variables you want answered by a selected record in the same dialog. If your template contains variables that must be answered by different records, you can group all the variables to be answered by one record together in one dialog, all the questions for another record in another dialog, and so on.

If you are integrating HotDocs with another program you can also specify a DLL file as an answer source that will link the dialog to the third-party program. (Contact your HotDocs sales representative for more information about using the HotDocs API.)

To create an answer source

1. Edit or create a dialog that contains the variables you want to associate with an answer source file.

Remember that variables in each dialog can be answered by only one record. If you have variables that must be answered by a different answer source, place those variables in a different dialog.

- 2. Click the **Options** tab. The window changes to show several custom options.
- 3. To select an existing answer source file, click the **Answer source** drop-down button, or create an answer source by typing a file name with a three-letter extension, such as .ANX or .HPL, in the

Answer source field. (To use an existing answer source, click the **Open** button and then search for the file.)

- 4. If you are linking an existing answer source to a dialog, and the variable names used in the answer source do not match the variable names in the dialog, click **Map Variables** to associate the variables so they can share information.
- 5. Click **OK** to close the **Dialog Editor**.

Once you have created an answer source, a **Select** button appears on the dialog during the interview. The user clicks this button to select an answer or enter new answers, and the answers are saved to the answer source file you specified.

Be careful when using an answer source with a dialog other than the one for which it was created. Each variable in the answer source must have a corresponding variable in every dialog with which it is used. Otherwise, if records are added or deleted, the records are added or deleted incorrectly and existing records will be damaged. You can write a script that uses the HIDE instruction to hide variables so they won't be visible in the dialog. If you don't want them to be visible in the answer source record, use either the CONCEAL or OMIT instruction in the same script.

When associating an answer source with a repeated child dialog, the child dialog cannot be repeated as a spreadsheet on the parent. If you do this, the **Select** button will not appear in the interview.

Tips for Working with Answer Sources

Keep the following tips in mind when creating and using answer sources for dialogs:

- Since answer source files are saved to the default *Answers* folder, be careful giving them an .ANX extension, or you may not be able to tell them apart from your regular answer files. It's best to choose a file name extension such as .HPL ("HotDocs pick list"), so you can distinguish them from other types of files.
- If an answer source file is not found in the default *Answers* folder, HotDocs looks for it in the default template folder next. If you want to store the file somewhere else, enter a folder path along with the file name in the **Answer source** field.
- Answer sources are automatically saved as you enter answers in them.
- When you view the **Answer source** drop-down list, you may see several options listed. One option, **CURRENT ANSWER FILE**, allows you to link variables in the current dialog to variables used in a repeated dialog. Other options you may see, including external programs such as the Corel Address Book, allow you to link dialogs directly to an application and associate variable names to use the information stored in the program to answer variables. (Only programs you have installed that support this type of integration will be listed.)

At a Glance: The Variable Mapping dialog box

d HotDocs Variable Mapping		? 🔀
Map variables in: CURRENT ANSWER FILE		
🗛 ariable Name	🚯 уре 🕜 Мар То	
Agreement Date	Date	-
Company Representative	Multi	<u> </u>
Signature Date	Date	•
1	(OK Cancel

You can open the **Dialog Editor** from the **Component Manager** or while editing a template. If you click the **Options** tab, and the **Map Variables** button at the bottom right you will see the **Variable Mapping** dialog.

There are three columns in the table displayed. The first column \boxed{A} shows the variable names, the second column \boxed{B} shows the type of variables. In the third column \boxed{C} you can choose the variable you wish to map to using the drop-down lists.

To learn more about using the Variable Mapping dialog box follow the links below:

- Link a Dialog to a List Gathered by a Repeated Dialog
- Link an Existing Answer Source to Other Dialogs
- Let Users Retrieve Answers from an Outlook Contacts List

• Let Users Retrieve Answers from a Time Matters Record

Link a Dialog to a List Gathered by a Repeated Dialog

At times, you may want to allow users to choose answers for a dialog from a list of answers they've already provided earlier in the interview.

To link a dialog to a list of answers

- 1. Edit the dialog whose answers you want to retrieve from an existing list. (See Edit a Custom Dialog.)
- 2. Click the **Options** tab. The window changes to show several custom options.
- 3. Click the Answer source drop-down button and choose CURRENT ANSWER FILE.
- 4. Click Map Variables. The HotDocs Variable Mapping dialog box appears.
- 5. In the **Map To** column, click the drop-down button corresponding to the first variable and select the variable that should share the same answer. (The drop-down list contains the variables in the current template that are the same type as the variables in the dialog.)

The information in the last step is relevant only if you are working with an internal mapping, or a shared component file. If you use a **command-line option** to create an **answer source** that uses an external component file, you must type the name of the variable from the external component file, since only variables internal to the current template appear in the drop-down list.

Link an Existing Answer Source to Other Dialogs

You can use an answer source with a dialog other than the one it was created for. You must know the answer source file name (and location) as well as the names of the variables referenced in it.

To use an existing answer source

- 1. Edit the dialog you want to link to the answer source. (This dialog can be in any template.) (See Edit a Custom Dialog.)
- 2. Click the **Options** tab. The window changes to show several custom options.

- 3. In the **Answer source** field, enter the answer source file name. (If the answer source is saved in a location other than the HotDocs *Answers* folder or the current template's folder, enter a full folder path along with the file name.)
- 4. If the variable names in the current dialog match those in the answer source, click **OK**. If not, continue this process.
- If the variable names in the current dialog don't match those in the answer source, click Map Variables. The HotDocs Variable Mapping dialog box appears. The variable names in the current dialog are listed in the Variable Name column.
- 6. In the **Map To** column, click the drop-down button corresponding to the first variable whose answer you want to come from the answer source. (The drop-down list contains the variables in the answer source that are the same type as the variables in the dialog.)
- 7. From the list, select the variable you want to map to the variable in the current dialog.
- 8. Continue selecting corresponding variables for each variable in the dialog you want to map.

Let Users Retrieve Answers from an Outlook Contacts List

This feature is supported in Microsoft Outlook 2000 and later.

You can link dialogs in your template to a Microsoft Outlook Contacts list. This allows your users to retrieve information they already know from Outlook, rather than typing it in manually.

To do this, you must first specify Outlook as an answer source for a specific dialog. Then you must map variables in the dialog to fields in the Contacts list. Like most variable mappings, data types must match. For example, you cannot map a Text variable to a date field, and vice versa.

To map variables in a dialog to fields in Outlook

- 1. Edit the dialog you want to link to the *Contacts* list. (This dialog can be in any template.) (See Edit a Custom Dialog.)
- 2. Click the **Options** tab. The window changes to show several custom options.
- 3. Click the **Answer source** drop-down button and choose **Outlook**.
- 4. Click **Map Variables.** The **HotDocs Variable Mapping** dialog box appears. The **Variable Name** column lists the variables in the current component file.
- 5. In the **Map To** column, click the drop-down button and choose the Outlook field that corresponds to the variable.
- 6. Repeat Step 5 for each variable you want to map.

Now, when users view this dialog during the interview, a **Select** button will appear on the dialog. When they click the button, HotDocs will display the Contacts list in Outlook where users can select the name of a contact and have the appropriate information merged into answer fields in the interview.

Let Users Retrieve Answers from a Time Matters Record

To link your dialog to a Time Matters answer source, you must be using Time Matters 7.0 or later. Additionally, you must select the **Activate Answer Source Integration** option when you enable the HotDocs link in Time Matters. (See the Time Matters documentation for complete instructions on enabling this link.)

You can link dialogs in your template to a Time Matters Contacts or Matters record. This allows your users to retrieve information they already know from Time Matters, rather than entering it manually.

To do this, you must first choose whether to link to a Time Matters Contacts record or a Matters record as the answer source. You then must map variables in the dialog to fields in the record. Like most variable mappings, data types must match. For example, you cannot map a Text variable to a date field, and vice versa.

When linking to Time Matters, you can also designate whether the user has write-back capabilities. This means that if the user changes an answer once it has been selected from Time Matters and entered in the interview, HotDocs can either always save the answer back to Time Matters, or it can allow the user to choose whether the answer should be saved back. If changes should never be reflected in the original Time Matters record, you can choose to disallow write-back.

To map variables in a dialog to a Time Matters record

- 1. Edit the dialog you want to link to the Time Matters record. (This dialog can be in any template.) (See Edit a Custom Dialog.)
- 2. Click the **Options** tab. The window changes to show several custom options.
- 3. Click the **Answer source** drop-down button and choose **Time Matters Contact** or **Time Matters Matter**.
- 4. Click **Map Variables**. The HotDocs **Variable Mapping** dialog box appears. The **Variable Name** column lists the variables in the current component file.
- 5. In the **Map To** column, click the drop-down button and choose the Time Matters field that corresponds to the variable.
- 6. Optionally, click the **Write Back** drop-down button and choose the option you need:
 - **Never** keeps the answer from being written back to Time Matters if it is changed.
 - **Always** always writes the answer back to Time Matters if it is changed. The user will not be notified or prompted of the change.
 - **Prompt** displays a dialog box that lets the user decide if the changed answer should be written back to Time Matters.
- 7. Repeat steps 5 and 6 for each variable you want to map.

Now, when users view this dialog during the interview, a **Select** button will appear on the dialog. When they click the button, HotDocs will display the Time Matters records list where they can select the specific record and have the appropriate information merged into answer fields in the interview. If users change an answer, the write-back options you selected will take effect when they save the answer field.

Inserting Dialogs into other Dialogs

Insert Dialogs Into Dialogs

You can insert dialogs into other dialogs. For example, you could create a dialog that collects basic client information and then, if the client is married, add that dialog to the client dialog to collect spouse information. Dialogs inserted into other dialogs are called *child dialogs*.

During an interview, a child dialog appears on its parent dialog as a button followed by the name or title of the dialog. When the user clicks the button, the inserted dialog appears and the interview outline expands to show the inserted dialog.

If you are inserting multiple child dialogs in a parent dialog and you specify a **Selection grouping** style, HotDocs places a check box or option button next to each inserted dialog. When this check box or option button is selected, HotDocs places the child dialog in the interview outline so that its contents can be answered. Otherwise, HotDocs will treat the child dialog as if it doesn't exist—even if there are required variables in it. In the interview, the user can either select the check box or option button and then click the child dialog icon, or the user can click the child dialog icon, which automatically selects the check box or option and then displays the child dialog.

You can insert as many levels of regular dialogs as you want. If you are nesting repeated dialogs, however, you can nest up to 3 levels of repeated child dialogs. See Create a List Within a List for details.

To insert one dialog in another

- 1. Create the parent dialog as well as the child dialog. (See Gather Questions into a Custom Dialog.)
- 2. Once both dialogs are created, edit the parent dialog. (See Edit a Custom Dialog.)
- 3. At the **Properties** tab, click the **Components** drop-down button and select **Dialogs** from the list. HotDocs limits the contents of the list to just dialogs.
- 4. Drag the dialog you want to insert to the Contents list.
- 5. Optionally, click the **Options** tab and select a grouping option from the **Selection grouping** drop-down list. (See Group Child Dialogs in a Parent Dialog.)

By default, child dialogs appear on parent dialogs as links with the child dialog's name as the prompt. If you want to provide more instructive text for accessing and then answering questions in the child dialog, enter a prompt in the **Prompt to use when displayed as child dialog** field.

(The prompt you specify here will be used in the dialog only—the dialog's name or title will still be used in the interview outline and in the dialog itself.)

Group Child Dialogs in a Parent Dialog

When you add two or more child dialogs to a parent dialog, you can group the child dialogs so users can more easily specify which dialog they want to answer. When you group child dialogs, HotDocs places a check box or option button in front of the child dialog icon. When this check box or option button is selected, HotDocs places the child dialog in the interview outline so that its contents can be answered. Otherwise, HotDocs will treat the child dialog as if it doesn't exist—even if there are required variables in it. In the interview, users can either select the check box or option button and then click the child dialog icon, which automatically selects the check box or option and then displays the child dialog.

If child dialogs are not grouped, each child dialog will appear in the interview outline, regardless of whether the dialog is optional to answer. If the user does not answer every question in every dialog, HotDocs will report in the *End of Interview* dialog that there are unanswered questions.

To group child dialogs in a dialog

- 1. Create the parent dialog as well as the dialogs you want to insert. (See Gather Questions into a Custom Dialog.)
- 2. Once your dialogs are created, edit the parent dialog and add the child dialogs to it. (See Edit a Custom Dialog and Insert Dialogs Into Dialogs.)
- 3. While editing the parent dialog, click the **Options** tab. The window changes to show options for working with the dialog.
- 4. Click the Selection grouping drop-down button and select Select One or Select All That Apply.

When child dialogs are grouped, you can use a SET instruction to automatically select a child dialog for answering. Doing this will automatically add the correct child dialog to the interview outline. You can also test whether a child dialog has been selected for answering by using a True/False expression.

Using ASK Instructions to Control Dialog Order

At a Glance: The ASK Field dialog box

ASK Field		? 🗙
Field type	(Data <u>b</u> ase	
Dialog:	6	▼ 👔
Show <u>A</u> dvanced		OK Cancel

You can open the **ASK Field** dialog box while editing a template by clicking the **EASK Field** button on the HotDocs Ribbon.

To begin with you can choose one of the two check boxes A at the top of the dialog. Select **ASK Dialog** to insert an ASK instruction that displays a specific dialog or select **ASK Database** to insert an ASK instruction that displays a database table from which the user can select one or more records.

You can then enter a name for your new dialog or database component in the field **b**elow or choose an existing component from the drop-down list. Click the **dEdit Component** button the right to display the Dialog Editor so you can make changes to the dialog or database component you are asking.

To learn more about using the ASK Field dialog box follow the links below:

- Control When Your Dialogs Appear
- Put ASK Instructions in a Computation Variable
- Use ASK Instructions to Control the Interview

Control When Your Dialogs Appear

When HotDocs creates an interview for the user, it reads through the template and displays dialogs based on the order it encounters variables and instructions in the template. However, if you want your dialogs to appear in a different order—for instance, if you want a certain dialog to appear first, even though variables that prompt the dialog to be asked appear at the end of the template—you can use an ASK instruction to force HotDocs to display the dialog.

To insert an ASK instruction

1. Position the cursor at the point in the template where you want the dialog to appear (for example, at the top of the template).

- 2. Click the **ASK Field** button. The **ASK Field** dialog box appears.
- 4. Click **OK**. The ASK instruction is inserted in the template.

You can place several ASK instructions in a single computation script to control the order *all* dialogs are asked in the interview. See Put ASK Instructions in a Computation Variable and Define a Custom Interview.

Sometimes you want to restrict a dialog so it only appears in the interview when it is specifically asked (using an ASK instruction). To do this, clear **Link variables to this dialog** (at the **Options** tab). If a dialog that has this option cleared is not specifically asked, variables in the dialog will be asked individually.

In Microsoft Word, you can also insert ASK instructions either by clicking on the **HotDocs** dropdown menu in the HotDocs toolbar and choosing **ASK Field**, or by right-clicking in the template and choosing **ASK Field** from the shortcut menu.

Ask automatically has replaced the need for using ASK NONE / ASK ALL instructions. However, they will still be available for this release.

Put ASK Instructions in a Computation Variable

When you insert an ASK instruction directly in a template, the instruction can ask only one dialog at a time. If you want to ask several dialogs in a specific order, you must insert individual ASK instructions for each dialog. Some template developers like this approach to asking dialogs. Others, however, prefer to group ASK instructions together by creating one computation that asks all the dialogs, and then inserting just the Computation variable into the template. In either case, during assembly, the dialogs appear in the interview outline in the order of the ASK instructions.

To put ASK instructions in a Computation variable

- 1. Create the dialogs you want to ask. (See Gather Questions into a Custom Dialog.)
- 2. At the template, position your cursor where you want to insert the Computation variable that asks the series of dialogs.
- 3. Create a Computation variable. (See Customize a Computation Variable.)
- 4. At the **Computation Editor**, drag **ASK DIALOG** from the **Instruction models** list to the **Script** field.

- 5. Replace the **DIALOG** placeholder with the name of a dialog you want asked. (See Use the Script Editor.)
- 6. Repeat Steps 4 and 5 for each ASK instruction. Your computation script may look something like this:
 - ASK Attorney Information
 - ASK Court Information
 - ASK Client Information
- 7. When you are finished, click OK.

Use ASK Instructions to Control the Interview

HotDocs 5 users: If your HotDocs 5 templates use ASK UNANSWERED or ASK DEFAULT instructions, those instructions will be treated as ASK ALL in HotDocs 11. Command-line options that once controlled these are no longer functional.

In addition to using an ASK instruction to ask dialogs or variables, an ASK instruction can control whether variables and dialogs required to assemble a document are asked during assembly. Changing ASK instructions lets you work with answers behind the scenes—assigning values without displaying the variables to the users. Specifically, ASK NONE keeps HotDocs from asking variables or dialogs that would otherwise be asked automatically.

Ask automatically has replaced the need for using ASK NONE / ASK ALL instructions. However, they will still be available for this release.

See How Variables and Dialogs Are Asked for details on how HotDocs determines when to ask variables and dialogs.

Using ASK instructions is particularly useful if you need to repeat a dialog but you don't want the dialog to be asked during the interview.

You can insert an ASK instruction directly in a template or place it in a Computation variable and insert the variable in the template.

To change the ASK mode in a text template

1. Place your cursor at the point in the template where you want HotDocs to control what gets asked.

2. Insert an ASK instruction (see Control When Your Dialogs Appear). Once the instruction is inserted, replace the dialog name with one of the following keywords: **ALL** or **NONE**.

To change the ASK mode in a computation script

- 1. Open a Computation variable for editing. (See Customize a Computation Variable.) The **Computation Editor** appears.
- 2. Drag **ASK DIALOG** from the **Instruction models** list into the **Script** field.
- 3. Replace **DIALOG** with one of the following keywords: **ALL** or **NONE**.

If you change the ASK mode in a computation, you should always change it back to what it was originally (**ALL** or **NONE**).

Defining Interviews and Interview Templates

HotDocs Interviews Overview

This document provides an overview of HotDocs Interviews:

Interviews Defined

Once you automate a template, HotDocs can generate an interview from that template. A HotDocs interview is a series of dialogs, each containing one or more questions which HotDocs generates from the variables you add to your template. The template's user must answer the questions in the interview before HotDocs can assemble a custom document. Once a person completes the interview, HotDocs assembles the final customized document from the template by merging the interview answers into the template text. You can view the interview by clicking the Interview tab of the assembly window. The left pane of the assembly window displays an outline of the interview. As the user answers questions, the outline tracks the user's progress through the interview.

Choose an Interview Type

When you assemble a document from a template, HotDocs displays either a default interview or a custom interview to gather the data needed to assemble the final document:

- Default Interview: HotDocs generates a default interview by asking the variables it encounters in the template. When you link a variable to a dialog, HotDocs asks the dialog. For most templates, this default interview is sufficient.
- Custom Interview: If you need control over the order questions are asked in the interview, you can create a custom interview component. A custom interview is defined by a computation variable script in which you use a series of ASK instructions to ask the variables and dialogs in your template. A custom interview can also contain other scripting, such as IF instructions and REPEAT instructions.

HotDocs interviews are dynamic, meaning that each time the user answers a question HotDocs updates the entire interview to reflect any changes caused by that answer. In more complex templates this update may take longer than expected as HotDocs must process each field. Using a custom interview you create a script that asks these components and processes only once. This can improve assembly speed considerably.

Understand the Interview Generation Workflow

In generating an interview from a template, HotDocs follows the following workflow:

1. HotDocs first checks to see if you created a custom interview for the template

- a. If you created a custom interview, HotDocs inspects the interview's computation variable for the presentation order of the variables or dialogs in the interview.
- 2. If there is no custom interview, HotDocs generates a default interview, containing the basic dialogs necessary to present the template variables as questions to the end user.
 - a. HotDocs generates the default interview by scanning through the template, starting at the top left corner and finishing at the bottom-right corner, just as a person would read through the template.
 - b. When HotDocs locates a variable in the template, it presents the variable as a question in the interview.

If you linked a variable to a dialog, HotDocs displays the dialog in the interview.

c. As it reads through the template, HotDocs also processes and executes any instructions you added to the template (for example, IF, REPEAT, and INSERT instructions).

d. If you want to decrease the number of dialogs the user has to click through you can choose to combine default dialogs.

To reduce the amount of dialogs generated in a default interview HotDocs combines variables into as few default dialogs as possible, if you would prefer to have one variable per dialog you can turn off this feature at Component File Properties

Complex templates require more script processing, which can slow interview interactions. You can use a custom interview to minimize the amount of script processing and increase interview responsiveness.

Implicit Asking Versus Explicit Asking

When HotDocs encounters a variable as it scans through the template, it uses one of two means to determine whether to add the variable to the interview:

• **Implicit Asking**—HotDocs checks to see if it has already added that variable in the interview; if not, HotDocs adds the variable to the interview. HotDocs only adds a variable again if instructed to do so by an ASK instruction in the Interview Computation variable.

If you have linked the variable to a dialog, HotDocs adds all variables on that dialog to the interview and considers the dialogs' variables "asked" as well; HotDocs does not add to the interview again.

• **Explicit Asking**—An ASK instruction within the template; this causes HotDocs to add the asked variable or dialog to the interview, regardless of whether HotDocs has included it before.

Use ASK instructions carefully to avoid duplicating variables in your interview.

Previous versions of HotDocs used the ASK NONE instruction to stop HotDocs from building interviews using implicit asking. This instruction has been known to behave
unpredictably in desktop HotDocs and differently in browser interviews. Therefore, we now recommend using the Ask Automatically feature to control the interview.

Have HotDocs Disable Irrelevant Variables in Interviews

When creating dialogs for an interview, you frequently need variables within a dialog to be dynamic. For example, you may want to disable some questions in a dialog until a user answers a different question in the dialog a specific way. You can either write a script to disable a dialog, or you can have HotDocs automatically hide or disable variables, based on whether they are used in the document.

For example, say you have a dialog that asks (using True/False variables) whether a client has any preexisting health conditions. If the client does, he or she must enter an explanation for each health condition. You can set variable preferences that keep the explanations from being asked unless the corresponding True/False questions are answered affirmatively. Then, if a dialog contains only inactive variables, you can create a script to keep HotDocs from asking the dialog from in the interview.

When you start an assembly, HotDocs processes the templates to determine (based on the interview's answers) which variables are referred to in you instructions. If a variable is unused it will be grayed or hidden. Once HotDocs has checked for unused variables, HotDocs does another pass through the template to generate the interview.

Interview Templates Defined

In addition to creating interview components to use with existing HotDocs templates, you can create an interview template for gathering common information (such as court, attorney, or client information); then save the answers as an answer file for use in assembling other documents.

An interview template is a component file that contains the variables for which you need to get answers. Users can select an interview template for assembly just as they can any other template. Unlike Text and Form templates, however, HotDocs can only use an interview template to generate an answer file, not to assemble a completed document.

A common use for an interview template is to create an answer source file. For example, you can create an interview template that gathers a list of information about attorneys in a law firm and have that information saved in an answer source file. You can then link a specific dialog in your template to that answer source so that when users view it, they can select answers from that list rather than enter them manually. You can use command-line options to lock the answer file so users cannot edit it.

Generating a Default Interview

For less complex templates, you can have HotDocs generate a basic interview so you don't have to create dialogs yourself.

To generate a default interview for a template

- 1. Open Component Manager.
- 2. Click the **Component File Properties** button.
- 3. Click the **Interview** tab.
- 4. Select Generate default interview.

When HotDocs assembles a document from the template, HotDocs asks the questions in the interview based on the order of the corresponding variables in the template.

You can reduce the number of dialogs in an interview by selecting **Combine default dialogs** on the **Component File Properties** window.

If you want to determine the appearance of one or two dialogs in an interview yourself, you can insert ASK instructions at the place in the template where you want your questions asked. If you want to control more aspects of your interview, you can use a custom interview script.

Combine Default Dialogs

To decrease the amount of dialogs in the interview you can use the Combine Default Dialogs option to have HotDocs ask all interview questions on one dialog.

In the 11.1 release, this feature is valid only for templates published to the desktop environment.

To generate combine default dialogs in an interview

- 1. Open Component Manager.
- 2. Click the **Component File Properties** button.
- 3. Click the **Interview** tab.
- 4. Select Combine Default Dialogs.

When HotDocs assembles a document from the this template, HotDocs asks the questions in the interview in one dialog where it arranges the questions based on the order of the corresponding variables in the template. If you also have a dialog component in your component file, HotDocs asks all the questions until it encounters the first variable used in that dialog. HotDocs will then ask the dialog, followed by all the remaining questions in the template.

If you want to determine the appearance of one or two dialogs in an interview yourself, you can insert ASK instructions at the place in the template where you want your questions asked. If you want to control more aspects of your interview, you can use a custom interview script.

Improve Interview Speed Using a Custom Interview

Once you automate a template, you can use that template to generate an interview that displays the questions the user must answer in order to assemble a custom document based on that template.

It is important to understand how HotDocs generates both a default and a custom interview.

Generate a Default Interview

With default interviews, HotDocs first scans the template and extracts all variables and instructions from the template. From this scan, HotDocs produces an "extracted script."

The example below shows a short template, followed by the corresponding extracted script:

(Template text)

Last Will of «Client Name»

I, «Client Name», being of sound mind, do make this my last will and testament. «IF Client Marital Status = "Married"»My «Client Spouse Gender: wife/husband»'s name is «Spouse Name». «Client Spouse Gender: she/he» is my personal representative. «ELSE IF Client Marital Status = "Never Married"»I have never been married. «END IF»

(Extracted script)

«Client Name»
«Client Name»
«IF Client Marital Status = "Married"»
«Client Spouse Gender: wife/husband»
«Spouse Name»
«Client Spouse Gender: She/He»
«ELSE IF Client Marital Status = "Never Married"»
«END IF»

Once HotDocs creates an extracted script, HotDocs processes the script and generates an interview, displaying any variables in the script as questions in the interview. (When a variable is linked to a dialog, HotDocs displays the dialog instead.) Additionally, HotDocs processes other instructions (for example, ASK, REPEAT, or IF instructions) included in the extracted script. While the extracted script may refer to a particular variable or instruction multiple times, once a user has answered the question corresponding to the variable or instruction, HotDocs does not ask that question again.

For most templates, allowing HotDocs to generate the interview this way is sufficient. In fact, HotDocs sets all templates, by default, to use this method.

Generate a Custom Interview

For complicated templates with lots of scripting, HotDocs' default interview generation can produce an interview that responds slowly to user interactions. This is because in default interviews, HotDocs must process every variable and instruction in the extracted script—no matter how many times they repeat, and even if the user has already answered the question or the instruction has already been executed. Depending on the complexity of the script, this may cause a delay when moving between answers in the dialog or between dialogs in the interview.

To avoid such issues, you can create a custom interview that contains only the required scripting and variables. You create this script using a computation variable. The following example shows a custom interview script that might be created for the template already shown above:

(Custom interview script)

ASK Client Information IF Client Marital Status = "Married" ASK Spouse Name END IF

Once you write the custom interview script, you must select a component file property to tell HotDocs to use your custom interview script instead of the default script.

Defining a Custom Interview Component

If you need control over the order HotDocs asks questions in an interview, you can create a custom interview component.

There are two parts to using a custom interview in your template:

- Create the custom interview component that contains all of your scripting.
- Specify a component file property that tells HotDocs to use the component when generating the interview.

To create a custom interview component

- 1. After automating your template, open Component Manager.
- 2. From the Components drop-down list, select **Computation Variables** and click the **New Component** button.
- 3. Type a name in the **Variable name** field. (The component can use any name (including INTERVIEW) as long as it's specified in the **Interview** tab of the **Component File Properties** dialog box)
- 4. Using a series of instructions, specify how you want HotDocs to ask the variables in the template, based on the logic you use in the template. For example, you can create a series of ASK instructions that ask the dialogs, as well as use IF instructions to make variables in the template conditional upon users' answers.
- 5. Click **OK** when you have finished.

Once you have created the custom interview, you must specify a component file property that tells HotDocs to use the computation when it generates the interview.

Tell HotDocs to use the custom interview component when generating the interview

- 1. With **Component Manager** still open, click **Component File Properties**.
- 2. Click the Interview tab and select Use custom interview.
- 3. Specify the name of the Computation variable in the **Interview component** field.

Now, HotDocs will use this computation to generate the interview.

If you are pointing several templates to a shared component file, but you want each template to use its own custom interview, at the **Interview** tab, clear **Use Interview properties stored in the shared component file**, select **Use custom interview**, and then specify the name of the interview computation in the **Interview component** field.

In a text template, if you select both **Generate default interview** and **Use custom interview**, HotDocs always uses the custom interview. That means that HotDocs only asks variables contained in the custom interview. For a form template, if you select both options, HotDocs presents the custom interview first and then includes any variables not in the custom interview, but contained in the default interview, at the end of the interview outline.

Disabling Irrelevant Variables in Interviews

You can have HotDocs hide or disable variables, based on whether the user is going to include them in the document.

There are three steps to automatically disabling unused variables:

- Specifying at the variable level whether variables should be grayed or hidden if they're not needed in the document.
- Specifying at the dialog level whether dialogs that contain only irrelevant variables should be left out of the interview entirely.
- Selecting a component file property that automatically disables irrelevant variables in the interview.

To mark how HotDocs should treat variables if they're not needed in the document

- 1. Edit the variable.
- 2. At the **Advanced** tab of the **Variable Editor**, click the **When irrelevant** drop-down button and choose an option:
 - **Default** causes the variable to use the default setting, which you define at the Component File Properties window.
 - **Gray** causes the variable to appear grayed out if the user does not include it in the document. (This is the default option for all variables.)
 - **Hide** causes HotDocs to hide the variable if the user does not include it in the document. If the user changes the answer file and the variable is now needed, HotDocs shows the variable in the dialog.
 - **Show** causes the variable to always appear in the dialog, regardless of whether it's used or not.

The **When irrelevant** drop-down list is only available when you have selected the component file property **Automatically disable irrelevant interviews and dialogs**.

To automatically exclude dialogs that contain only irrelevant variables

- 1. Edit the dialog.
- 2. Click the **Options** tab.
- 3. Select **Prevent this dialog from being asked when irrelevant**.

This option to **Prevent this dialog from being asked when irrelevant** is only available when you have selected the component file property **Automatically disable irrelevant interviews and dialogs**.

To automatically disable unused variables in the interview

- 1. Open **Component Manager** for the template.
- 2. Click the **Component File Properties** button.
- 3. Click the **Interview** tab.
- 4. Select Automatically disable irrelevant variables and dialogs.
- 5. Click the **Default** drop-down button and choose one of the options from the list:
 - Gray causes HotDocs to gray all variables with their irrelevant status set to Default.
 - Hide causes HotDocs to hide all variables with their irrelevant status set to Default.
 - **Show** causes HotDocs to always ask all variables with their irrelevant status set to **Default**, even if the variable is irrelevant.

HotDocs sets HotDocs Models assembled from markup to disable unused variables. Even disabled, these unused variables are always visible in pop-up dialogs users view when they click a **form field wizard** button.

Creating an Interview Template

You can create an interview template to gather common information (such as court, attorney, or client information), and then save the answers for use in assembling other documents.

If you are using an interview template to generate an answer source for a specific dialog, you may want to assign command-line options to the interview template file properties that lock answer files and save the answers to a specified answer file.

To create an interview template

- 1. At the HotDocs library window, select the folder in which you want to create the interview template.
- 2. Click **New Template**.
- 3. Click the **Type** drop-down button and choose **HotDocs Interview Template (.cmp)**.
- 4. In the File name field type a file name for the new template. If you decide to store the template somewhere other than the default template folder (as seen in the Target folder field), you can also **Browse** to the new destination. HotDocs adds the correct extension to the file name based on the type of text template you selected. HotDocs displays the full path of the currently selected template location in the Target folder field.
- 5. Type a title for the interview template in the Title field (or accept the suggestion HotDocs makes). The title is what identifies the file in the library.
 - a. Optionally, type a description in the **Description** field. Descriptions appear in the **Properties** tab of the library when a user has selected the template.

b. To add an existing component file to the new template, you can select a component from the **Shared component file** drop-down list. If you do so, the component file you choose is now shared between the new template and any other templates using that component file.

c. To use the contents of an existing interview template or component file, in the Initial contents section, you can select **Other File** and then browse to the .cmp file you want.

6. Click **OK.** The **Component Manager** window appears. If you haven't chosen a shared component file, the Component Manager will contain a single computation variable named **INTERVIEW**.

Your interview templates are *not* required to use the Interview computation variable. If you want, you can delete this component and **create a new one**. After creating a new component you need to specify the name of the new interview component at the **Interview** tab of the **Component File Properties** window.)

The interview component you use *must* contain the script that asks the questions in the interview. If you use a shared component file and this component file does not contain an Interview component, you need to make one before HotDocs can assemble a document from the template.

- 8. Select the interview component and click **dit Component**.
- 9. Create the variables you want answered, add them to dialogs, and then create a script that asks the dialogs you want users to see when they assemble this template.

To edit an interview template, select it at the library and click **Edit Template**. Then, select the interview component and click **Edit Component**.

When converting interview templates from previous versions of HotDocs to HotDocs 2009-11 format, you may need to edit the **Component File Properties** for the component file and specify the name of the interview component you are using for the template.

You can use an **existing interview template** as the basis for a new interview template. To do this, select the existing template, click **New Template**, and enter a new file name for the template (but leave the information in the **Other file** field as is).

Using Lists of Answers

Using Computations to Create Lists

Create a REPEAT Instruction Using a Computation Variable

Instead of inserting a REPEAT instruction directly in a template to generate a list of answers, you can use a REPEAT instruction in a Computation variable to create your list of answers. Using a computation enables you to quickly insert a list in more than one location in the template.

A drawback to this approach is that HotDocs applies punctuating, sorting, and filtering wherever you insert the computation—the list appears exactly the same even in different parts of your assembled document. For example, you would be unable to a display a list of cities in a column at the beginning of the template and then as a punctuated sentence later on.

To use a REPEAT instruction in a computation

- 1. At the template, create a Computation variable.
- 2. In the **Script** field, type either empty quotation marks ("") or a zero (**0**) as the starting value for the computation.

When you use a script to merge a list of answers you have gathered using a REPEAT, you must first set the computation to a starting value. You must make this value a "" which represents "nothing" if the RESULT expression resolves to text, or a zero (**0**) if the RESULT expression resolves to a number.)

- 1. Type a REPEAT instruction, followed by the dialog name.
- Type the RESULT expression, followed by the names of the variables whose answers you want to merge. Use the plus (+) operator to add variables, text and space characters (including a hard return), and so forth, to what you want to repeat.
- 5. Type an END REPEAT instruction. An example of a possible script would be:

```
""
REPEAT Tour Information
RESULT + Name of City + ", " + Venue + "
"
END REPEAT
```

Punctuate a List Using a Computation Variable

When you use a computation to generate a repeated list, you can use the FORMAT instruction to punctuate the list in sentence style.

To punctuate a repeated list in a computation

- 1. Create a Computation variable that contains your REPEAT instruction. (See Create a REPEAT Instruction Using a Computation Variable.)
- Drag FORMAT "LIST_FORMAT" from the Instruction models list into the Script field on a line of its own, after the REPEAT instruction but before the RESULT expression.
- Replace the LIST_FORMAT placeholder with a punctuation format (such as "A, B, and C"). An example would be:

```
""
REPEAT Tour Information
FORMAT "a; b; and c"
RESULT + Name of City + ", " + Venue
END REPEAT
```

Sort a List Using a Computation Variable

If you are using a Computation variable to create a list of answers, you can sort answers in the list using the **ASCEND VAR** and **DESCEND VAR** instructions. These instructions sort a list from *A to Z*, and *1 to 9*.

To sort a REPEAT instruction in a Computation variable

- 1. Create a Computation variable that contains your REPEAT instruction. (See Create a REPEAT Instruction Using a Computation Variable.)
- Drag ASCEND VAR or DESCEND VAR from the Instruction models list into the Script field on a line of its own, after the REPEAT instruction but before the RESULT expression.
- 3. Replace the **VAR** placeholder with the repeated variable you want to sort on. An example would be:

....

REPEAT Tour Information ASCEND Name of City RESULT + Name of City + ", " + Venue + "

END REPEAT

When you use a script to merge a list of answers that have been gathered using a REPEAT, you must first set the computation to a starting value. This value must be a **0** if the result will be a number, or it must be "" if the result will be text.

To add a second sort level, insert a second **ASCEND VAR** or **DESCEND VAR** instruction below the first.

Filter a List Using a Computation Variable

If you are using a Computation variable to create a repeated list, you can filter the REPEAT instruction in a Computation variable by using the **FILTER COMPUTATION_VAR** instruction. (See Filter a List of Answers.)

A filter can be as complicated as it needs to be, but it must result in either true or false. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a true or false value—it is not a filter. But the expression YEARS FROM(Child's Birth Date, TODAY) <= 17 can only result in true or false. It can correctly filter all children under the age of 18 from a list.

To filter a REPEAT instruction in a Computation variable

- 1. Create a Computation variable that contains your REPEAT instruction. (See Create a REPEAT Instruction Using a Computation Variable.)
- Once you have added the REPEAT instruction to the script, drag FILTER COMPUTATION_VAR from the Instruction models list into the Script field. (Place this model on a line of its own, after the REPEAT instruction and any punctuation or sorting instructions but before the RESULT expression.)
- 3. Select **Computation Variable** at the **Components** list.
- 4. Click **New Component** to open a second **Computation Editor**.
- 5. Assign a name to the computation.
- 6. Type the filtering instructions in the **Script** field, and click **OK**. HotDocs returns you to the REPEAT computation and the new variable is added to the **Components** list.

7. Drag the new variable onto the COMPUTATION_VAR placeholder and complete the REPEAT instruction. An example of a possible script would be:

```
REPEAT Tour Information
ASCEND Tour City
FILTER New York Appearances
RESULT + Tour City + ", " + Tour State + ", " + Tour Venue + "
"
END REPEAT
```

The *New York Appearances* filter contains the script, Tour State CONTAINS "new york"

When you use a script to merge a list of answers that have been gathered using a REPEAT, you must first set the computation to a starting value. This value must be a **0** if the result will be a number, or it must be "" if the result will be text.

Tips on Using Computations to Create Lists

.....

There are a few things to note when using a computation to create a list of answers.

- In the **Script** field, before the REPEAT instruction, you must set the computation to a starting value. Most often this value will be "nothing"—either two quotation marks with nothing between them ("") if the result of the computation will be text, or a zero (0) if the result will be a number.
- Inside the REPEAT instruction, begin the part that is being repeated (the part with the repeated variable or other value in it) with RESULT followed by a plus character. This RESULT keyword updates the computation as it goes through each repetition. For example:

```
""
REPEAT Decedent Information
RESULT + Decedent Name + ", " + Date of Death + "
"
END REPEAT
```

Also, there are four special models you can use in a computation to get information from a REPEAT instruction.

- COUNT(DIALOG) counts the number of repetitions in a list.
- COUNTER gives you the number of the current repetition.
- SUM(COMPUTATION_VAR) totals a repeating Computation variable.
- SUM(NUM_VAR) totals a repeating Number variable.

Creating Lists within Lists

Introduction: Create Lists Within a List

You can create a sublist of answers by nesting one REPEAT instruction inside another. For example, you could create a list of children inside a list of parents, like this:

Paul Linares
Jack
McKenzie
Abby
Lisa White
Kevin
Logan
Savannah
Gregory Hurley
Randy
Jessica

The list of parents (*Paul Linares, Lisa White, and Gregory Hurley*) is the first repeat level. The sublists of children are the second repeat level. Repeats can be nested up to four levels deep. You can nest repeats directly in a text template or you can nest repeats using a Computation variable in a text or form template.

You can use instructions or expressions with nested REPEAT instructions if you use a Computation variable. And, you can merge a particular answer from a nested repeat list into a text or form document.

Nested repeats cannot be used if you are retrieving answers from a database.

Repeats can not be nested more than four levels deep.

Create a List Within a List

Sometimes you may need to gather a list of answers that must appear within a larger list of answers. For example, perhaps you need a list of your client's real estate holdings. For each holding, you also need property descriptions and improvements. You can create these lists by nesting one REPEAT instruction inside another.

There are two parts to creating a list within a list:

- First, you must select all the template text you want repeated and insert the required REPEAT instruction fields.
- Second, you must edit the properties of the parent (or main) dialog and insert the nested dialog(s) into it.

The first step in creating a list within a list is to REPEAT each level of text in the template. (See Create a REPEAT Instruction to Gather a List of Answers.)

To create a nested repeat

- 1. Create the variables you want repeated and insert them into the template. (See Insert a Variable Field in a Text Template.)
- 2. Gather the variables you want repeated into dialogs. (Make sure you choose a repeat style for each dialog. See Gather Questions into a Custom Dialog.)
- 3. At the template, select all the template text and variables you want repeated (including all repeat levels) and click the **GREPEAT Field** button. The **REPEAT Field** dialog box appears.
- 4. Select the dialog for the first (or top) level of the repeat from the **Dialog** drop-down list and click **OK**. HotDocs inserts the REPEAT instruction in the template. (Even though you have selected all levels of the repeat in the template text, this dialog should only contain the variables from the first level of the repeat.)
- 5. At the template, select the second level of repeated text and variables, as well as any subsequent levels of information (do not select the END REPEAT instruction) and click the **GREPEAT Field** button. The **REPEAT Field** dialog box appears again.
- 6. Select the dialog for the second level of the repeat from the **Dialog** drop-down list and click **OK**.
- 7. Repeat this process for each subsequent list.

The second part in nesting REPEAT instructions is to edit the contents of each dialog and insert each repeated dialog (except the dialog for the first repeat level) into the preceding level's dialog.

- 8. Edit the dialog in the first (or top) level of the repeat. (See Edit a Custom Dialog.) The **Dialog Editor** appears.
- 9. Drag the dialog for the second repeat level from the **Available Components** list into the **Contents** field and click **OK**.
- 10. Repeat this process until each dialog in the nested repeat (except for the first) is inserted into the preceding level's dialog.

Repeats can not be nested more than four levels deep.

By default, a child dialog appears as a button on the parent dialog. You can change the style of a repeated dialog to **Spreadsheet on parent**, which lets users enter the nested list of answers while viewing its parent dialog. (See Display a Child Dialog Directly on the Parent Dialog.)

If you are nesting REPEAT instructions in a form template, the REPEAT instructions must be placed in a Computation variable. See Nest REPEAT Instructions Using a Computation Variable for details.

Display a Child Dialog Directly on the Parent Dialog

By default, when you insert a repeated dialog on its parent dialog, it creates a button which the user must click in order to answer the questions in the dialog. However, HotDocs can display the inserted dialog directly on its parent dialog so users can view—and answer—both dialogs without changing the view. The inserted dialog will appear on its parent dialog as a spreadsheet.

Any dialog elements in the inserted child dialog will not appear on the parent dialog.

The following is an example of a nested repeated dialog. The *Book Information* spreadsheet is inserted in the *Author Information* dialog, and both are displayed simultaneously.

Using Lists of Answers

📄 Editor/Author List - HotDocs Tutorials		
File Edit View Navigate Tools Help		
🗐 🛛 New Answer File 🔹 🗖	🎯 🖪 🗈 🖶 🖪 [2] 🖓	s 🖸 🔞
∠ ∽ ~ & ``````````````````````````````````	&↓ <u>∠</u> =≜ ₩	
Interview Document Preview Ques	stion Summary Answer Summary	
Editor Information	1: Editor Information	
□ 1: Editor Information	1:1: Author Information	
1:1: Author Information	Author Full Name	
	Book Title	Date Completed
	2	
	3	
	4	
	5	
	6	
	7	_
	K Fi <u>r</u> st	Next 🕨 📂 Last 🔰 Finish 🖻
		\$
)	
		NUM

To display a spreadsheet dialog on its parent

- Edit the inserted dialog (or the dialog you want to appear as a spreadsheet on its parent dialog). (See Edit a Custom Dialog.) The **Dialog Editor** appears.
- 2. Click the Style drop-down button and select Spreadsheet on Parent.
- 3. Optionally, enter a number in the **Rows to display** field to control the number of viewable rows that appear in the dialog.
- 4. Click **OK**.

Nest REPEAT Instructions Using a Computation Variable

You can use a Computation variable to nest REPEAT instructions. This allows you to use other instructions or expressions to create more complex repeated dialogs.

For example, perhaps you want to use a computation to generate a list of different departments within a company. Then you can list the names of employees within each department.

To nest REPEAT instructions in a Computation variable

1. Create a Computation variable and enter the script containing the nested repeats in the **Script** field. (See Customize a Computation Variable.) An example of a possible script would be:

```
""
REPEAT Department Information
RESULT + Department Name + " " + "Department" + "
"
REPEAT Employee Information
RESULT + " " + FORMAT( COUNTER , "9" ) + "." + " " + Employee Name +
"
END REPEAT
END REPEAT
```

- 2. Open each dialog for editing and insert each repeated dialog (except for the first) into the dialog preceding it. (See Insert Dialogs Into Dialogs.)
- 3. Select a repeated dialog style for each repeated dialog. (See Choose a Presentation Style for the Repeated Dialog or Display a Child Dialog Directly on the Parent Dialog.)

Repeats can not be nested more than four levels deep.

Use a Particular Answer from a Sublist

Just as you can merge a particular answer from a regular list into the document, you can merge a particular answer from a sublist. To do this, you must include in the brackets the numbers of the answers leading to the answer you want to use, beginning with the first repeat level.

The following graphic shows the hierarchy HotDocs follows when determining which answer to merge into a specific field. For example, if you need to merge the birthday of the third grandchild born to the fourth child of the second parent, you would specify the following information in the variable name field: *«Grandchild Date of Birth[2,4,3]»*.

Then, during the interview, the user provides the answers and HotDocs retrieves the information from the correct levels of the nested repeat. Specifically, the 2 in the first title bar directs HotDocs to the second set

of answers entered for the variables on the repeated dialog *Parent Information*. The 4 in the second title bar directs HotDocs to the fourth set of answers entered for the variables on the repeated dialog *Child Information* for the second parent. The 3 in the third title bar directs HotDocs to the third answer entered for the variable *Grandchild Date of Birth* for the fourth child of the second parent.



To merge a particular answer from a nested repeat

- 1. Determine the number path leading to the answer you want to merge.
- 2. At the template, position the cursor after the variable name, but inside the chevrons.
- 3. Type brackets ([]), and then, within the brackets, type the first number of the path, followed by a comma, then the second number, followed by a comma, and so forth. Don't put spaces after the commas, and don't put a comma after the last number.

As you select specific records from a nested repeat, enter all the necessary numbers in the path. HotDocs always uses a 1 if a necessary number is missing, and it always places the *1* after the number(s) you entered. If you don't enter any numbers, HotDocs uses [1,1,...] as the default number path.

When automating a form template, you must enter the number path in the **Variable** field at the **Field Properties** dialog box. (See Retrieve a Specific Answer from a List.)

Managing Components and Templates

Using Component Manager

At a Glance: The Component Manager



You can open the **Component Manager** directly from the library or from the HotDocs ribbon while editing a template.

The **Toolbar** A in the **Component Manager** runs vertically down the left side of the window. From here you can access the following options:

- **Arrange Windows:** Places the Component Manager window to the left of either the library window, Automator window, or word processor window. This way, you can work in both windows simultaneously.
- Save Component File: Saves any changes you have made while working with Component Manager. (To save changes made in the template, click the **Save Template** button in template window.)
- **Tiew Component File Properties:** Displays the **Component File Properties** dialog box, where you can change settings for the current component file.
- ****Insert a Variable:** Inserts the selected variable into a template at the current cursor position.
- ²Create a New Component: Creates a new component, such as a variable, clause, or dialog.
- **Edit a Component:** Displays the Component Editor, where you can make changes to the selected component.
- **Delete a Component:** Removes the selected component from the component file. If the component is used in the template, you must remove it there, as well.
- *** Rename a Component:** Displays the **Rename Component** dialog box. Renaming a component here changes all references to it in the component file; however, you must manually rename references to the component that appear in the template.
- **Duplicate a Variable:** Creates a new component by copying an existing one.
- Export Computation to Clipboard: When a computation variable is selected you can export the computation to the clipboard as HotDocs readable syntax. (See Import or Export Computations)
- **Import Computation from Clipboard:** Imports a computation from the clipboard and adds it to the component file as a computation variable (See Import or Export Computations)
- **A** Find in Components: Displays the Find in Components dialog box, where you can search for components that contain a specific text string.
- ^{ab}_{ac} **Find and Replace in Components:** Displays the **Find and Replace in Components** dialog box where you can locate and change text strings in your components.
- Spell Check Components: Checks text used in components (such as prompts) for accurate spelling.
- **Print Components:** Prints a list of components. You can choose to print the variable types, prompts, formatting, and any plain text resources that have been assigned.
- **BRestore all:** Brings all open component editors to the front.
- **Section** Minimize all: Minimizes all open component editors to the Windows taskbar.
- **Save and Close:** Saves changes to and closes all open component editors.
- **Weighted Help:** Opens the relevant page of the HotDocs Help File.

At the top right of the window you can see the **Expand** button **B**. You can click this button to display the component file of another template along side this one. This enables you to compare and share components more easily.

You can use the drop down list \bigcirc near the top of the window to filter the type of components displayed in the **Component List** \bigcirc below. You can also use the $2 \downarrow$ **Sort** button to the right of the drop down list to sort the components alphabetically. You can select components from the list then use buttons from the Toolbar \triangle to select options or double click on the component to open the Component Editor. You can also search your components list using the **Find** field \sqsubseteq at the bottom of the window.

To learn more about the component manager follow the links below:

- Open and Close Component Manager
- Open Component Manager at the Template Library
- Use Component Manager to Work with Components
- Create and Edit Multiple Components Simultaneously
- Change Component File Properties
- Delete a Component from the Current Component File
- Copy Components From One File to Another
- Rename Components in a Single Template
- Make a Duplicate Copy of a Variable
- Edit Formats, Merge Text, Dialog Elements, and Patterns

Open and Close Component Manager

You can edit components in a template by using Component Manager.

To open Component Manager

- 1. At the HotDocs template, click the **Component Manager** button. The **Component Manager** window appears.
- Optionally, adjust the Component Manager window to the desired width and click the Arrange Windows button. The Component Manager window appears to the left of the template development window.
- 3. Once Component Manager is opened, you can perform any number of tasks, including:
 - Using Component Manager to work with components.
 - Creating and inserting a variable using Component Manager.
 - Creating and editing multiple components simultaneously.

• Changing component file properties.

To close Component Manager, click the **X** in the upper-right corner of Component Manager.

You can also open Component Manager at the template library. This allows you to create and edit components without actually opening the corresponding template. It also allows you to more easily edit the component file of a clause library or interview template. (See Open Component Manager at the Template Library.)

Open Component Manager at the Template Library

In addition to using Component Manager while you are editing a template (see Open and Close Component Manager), you can use Component Manager to modify the contents of a component file when you are at the HotDocs template library. This may be useful if you need to make changes in the component file, but don't want to open the corresponding template to do it. It is also useful if you need to edit an interview template (see Create an Interview Template), as well as edit clause components in a clause library.

Some changes you make in the component file are not always reflected in the template. For example, if you rename a component at Component Manager without also renaming it in the template, you will receive errors when you assemble the document. Update your template with changes as necessary to avoid such problems.

To open Component Manager at the template library

- 1. At the HotDocs library, select the template or clause library whose component file you want to open.
- 2. Click the **Component Manager** button. The **Component Manager** window appears.
- 3. Make changes to the component file as necessary, such as create new components, edit existing components, and rename components. (See Use Component Manager to Work with Components.)

Use Component Manager to Work with Components

Using Component Manager allows you to work with individual components in a HotDocs template, including copying components between component files, creating new components, and editing existing components.

You can open and close Component Manager as needed, or you can leave it displayed as you work in the template, switching between the two windows. One advantage of leaving Component Manager open is the ability to view all the components in your template at once, and edit them simultaneously as needed.

To work with individual components in a component file

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager window appears.**
- 2. Perform any tasks, as described in the following table:

То	Do This
Sort components in Component Manager either by alphabetical name or by component type	Click the $2 \downarrow$ Sort Components button and then choose either Sort by Name or Sort by Type.
	To sort component lists that appear in the Dialog Editor and Computation Editor , right- click on the list and then choose your sorting option from the shortcut menu.
Display Component Manager side by side with the template development window	Adjust Component Manager to the width you want and click the Arrange Windows button. HotDocs adjusts the size of the template development window so that both windows can be viewed in full.
Save changes you have made to the component file	Click the Gave Components button.
Specify certain assembly and interview preferences for the template and component file	Click the Component File Properties button. (See Change Component File Properties.)
Use Component Manager to insert a variable in the template text	Place your cursor in the template where you want the variable, select the variable from the Components list, and click the **Insert Variable button. (See Create and Insert a Variable Using Component Manager.)
Create a new component, including a variable, dialog, or clause	Click the Components drop-down button, select the type of component you want to create, and click the New Component button.
Make changes to an existing component	Select the component from the component list and click the dit Component button. (You can also double-click the component.)

Remove a component from the component file	Select the component in the component list and click the Delete Component button.
	The component is removed from the component file and any associated dialogs, but references to it in the template and other components (such as scripts, prompts, or dialog text elements) will not be updated. If you've referred to this component in any of these ways, you must manually update these references or your template may not work.
Assign a new name to a component	Select the component(s) and click the ¹ Rename Component button. (See Rename Components in a Single Template.)
Make a duplicate copy of a variable or group of variables	Select the variable(s) and click the Duplicate Variables button. (See Make a Duplicate Copy of a Variable.)
Bring all open Component Editors to the front	Click the Restore All button. HotDocs brings all open Component Editors to the front so you can view and edit them.
Minimize all open Component Editors to the Windows taskbar	Click the Minimize All button. All open Component Editors are then minimized. However, you can still access them by clicking their icon in the Windows taskbar.
Close all open Component Editors	Click the Save and Close All button. All changes made to components are saved and all open Component Editors are closed.

To access the Component Manager toolbar using the keyboard, press F10.

You can open a component file for another template while viewing the current component file. Once opened, you can edit components or assign other properties. To do this, click the **Expand** button, and then select the component file. Once open, edit the component. You can also copy components between the open component files. See **Copy Components From One File** to Another.

For information on searching and replacing text strings in the component file, spell checking components, and printing lists of component properties, see Search Component Files, Print a List of Components, and Spell Check Components.

Create and Edit Multiple Components Simultaneously

Using Component Manager you can create and edit as many components in a given component file as you want, all at the same time. This allows you to compare the properties, advanced options, selection options, and computation scripts of many variables and dialogs. To manage all the open component editing windows, you can either use the **Restore All**, **Minimize All**, and **Save and Close All** buttons in Component Manager; or you can use the Windows taskbar to switch between open windows.

To edit multiple components simultaneously

- 1. Open Component Manager. (See Open and Close Component Manager.)
- 2. At the **Component Manager** window, you can either edit existing components or create new ones:
 - To edit an existing component, select it from the list of components and click the **d** Edit **Component** button. Do this for each component you want to edit. (HotDocs adds the component window to the Windows taskbar so you can more quickly access it.)
 - To create a new component, click the **New Component** button. Do this for each component you want to create.
- 3. Make changes to each variable as necessary, using the following tools to help you manage the open windows:
 - To bring all open Component Editors to the front, click the **Restore All** button.
 - To minimize all open Component Editors, click the \exists Minimize All button.
 - To save and close all open Component Editors, click the **Save and Close All** button. (Any changes you have made to component properties will automatically be saved—you will not be prompted to save them.)
 - To switch between open Component Editor windows, select the component in the Windows taskbar. HotDocs will bring that Component Editor to the front.

If you create new components using Component Manager, they are not immediately inserted into the template—they are only saved in the component file. You must insert the variables into the template. See Insert a Variable Field in a Text Template.

How quickly HotDocs minimizes and restores component editing windows depends on your Windows settings. To restore instantaneously, right-click on the Windows desktop, choose **Properties** from the shortcut menu, click the **Appearance** tab, click **Effects**, and then clear **Use transition effects for menus and tooltips**. (This process may differ, depending on the version of Windows.)

To change a component in several component files, use Template Manager. (See Introduction: Use Template Manager.)

At a	Glance:	The Com	ponent File	Properties	dialog box

mpon		ies - Denio Ling	Joyment Ag	reement.emp		
eneral	Interview As	embly Hidden I	Data (Word)	HotDocs Server	Answer Up	load
emplate	e title:					
	-		A			
emplate	e description:					
			ß			
			Ŭ			
						-
	ant file formatio	C		0		
_ompone	ent file format:	Current Format		0		
emplate	e <u>i</u> dentifier: D	419dbc78-b86e	4cb4-852d-94	164b6bfce44	N	l <u>e</u> w
upplem	ental files .	Additional Ter	unlater (Cothan Addi	Hanal Eilen	
oppiem		Additional Ter	ipiates		uoriai Files	
hared o	component file:			G		
				<u>O</u> K	C	ancel

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar.

In the first text field A you can edit the title of the template, this title is used to identify the template in the template library file list, and in the second text field B you can edit the template description which will appear in the **Properties** tab of the library.

Underneath the two text fields are two drop-down menus. Using the first menu you can change the component file format, this can be used to make the component file compatible with earlier versions of HotDocs (see Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs). To keep the component file current with the version of HotDocs you have installed, choose **Current Format**.

Since the release of HotDocs 11.1, HotDocs no longer supports converting component files to pre-2009/10 formats.

Below this you can see the specific Template ID D associated with this template and use the **New** button to the right to generate a different ID.

If you have additional templates or files that need to be published with this template you can add these using the next two buttons. Clicking the first one copens the **Additional Templates** dialog box where you can add or remove other templates attached to this one. Clicking the second one copens the **Other Additional Files** dialog box where you can add or remove any other types of file you would like to attach to this template, for example you could attach a PDF guide or an image file. To learn more about using these dialogs see At a Glance: Additional Templates and Files (Component File Properties).

If you choose to use an earlier format of HotDocs be careful not to use new features that aren't supported in the older versions of HotDocs or you may experience errors during assembly.

In the drop-down list **G** at the bottom of the dialog you can choose to use the component file from another template, the menu will show you a list of available files.

Before you point a template to a shared component file, make sure you first close the template and then edit the component file from the template library. Shared and pointed component files must also be stored in the same folder.

There are further options available in the Interview, Assembly, Hidden Data, HotDocs Server and Answer Upload tabs.

To learn more about setting the component file properties follow the links below:

- Define a Custom Interview
- Change Component File Properties
- Specify Whether Component File Properties are Shared
- Make Templates Stop Sharing Component Files
- Specify a Word Template for Storing Post-Assembly Macros
- Work with Variables in Headers, Footers, Footnotes, and Text Boxes

At a Glance: Additional Templates and Additional Files (Component File Properties)

Managing Components and Templates

🔒 Additional Templates	Dther Additional Files
When publishing this template, always include the following additional templates:	When publishing this template, always include the following additional files:
Α	
	Add Remove OK Cancel

To access these features

- 1. On right side of the HotDocs Developer toolbar, click **Component Manager**.
- 2. Near the top of the Component Manager window, click **Component File Properties**.
- 3. On the **Genera**l tab, next to Supplemental files, click either **Additional Templates** or **Other Additional Files**.

Both of these dialogs behave in the same way but are used to attach different file types. The main portion of these dialog boxes is the files list A where you can see any templates or files currently attached to the template

You can use the **Add** and **Remove** buttons **b** to edit the additional file lists. Clicking the **Add** button will allow you to browse for a template or file to add while clicking on the **Remove** button with a list item highlighted will remove that item and it will no longer be published with the template.

For more information about the Component File Properties tabs see the links below:

- At a Glance: The Component File Properties Dialog Box
- At a Glance: Interview Tab (Component File Properties)
- At a Glance: Assembly Tab (Component File Properties)
- At a Glance: Hidden Data Tab (Component File Properties)
- At a Glance: HotDocs Server Tab (Component File Properties)
- At a Glance: Answer Upload Tab (Component File Properties)

At a Glance: Interview Tab (Component File Properties)

orementarian	Interview	Assembly	Hidden Data (Word)	HotDocs Server	Answer U	pload
🗸 Gen	erate default	t interview				
Use	custom inter	view				
Intervie	aw componen	it:		B		-
Auto	omatically dis	able irreleva	nt variables and dialog	s Default:	Gray	~
Hide	e interview ou	utline				
🔄 Hide	End of Inter	view <mark>dia</mark> log				
Gen	erate default	t titles for di	alogs			
Com	bine default	dialogs				
Resource	ce button lab	el:		D		
				Ŭ		

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **Interview** tab at the top of the dialog box to access the options for HotDocs interviews.

At the top of the dialog, use the two check boxes A to tell HotDocs to generate a default interview or to use the custom interview you define. If you select **Generate default interview**. HotDocs then asks the variables and processes the instructions in the template in the order it "reads" them. If you link a variable to a dialog, the dialog is displayed instead. Select **Use custom interview;** then from the drop-down list select the interview component you want to use. If you want to edit the interview component, or if you haven't selected a component from the drop-down list, click the **Edit Component** button to the right to edit the component or to create a new one.

You can set further interview options using the following five check boxes C:

• Automatically disable irrelevant variables and dialogs

Select this option to automatically disable unused variables in the interview. Use the **Default** drop-down list to the right to select the default style of irrelevant variables:

When this option is selected you can change the irrelevant variable style of individual variables by editing the variable, then at the Advanced tab select a style from the **When irrelevant** drop-down list.

- Gray causes HotDocs to gray any irrelevant variables.
- Hide causes HotDocs to hide all irrelevant variables.
- Show causes HotDocs to ask all variables, even if the variable is irrelevant.

• Hide the interview outline

Select this option to hide the interview outline and force the user to navigate the interview by only using the **Previous** and **Next** buttons.

• Hide the end of interview dialog

Select this option to hide the end of interview dialog. Then, when the user clicks **Next** at the last dialog in the interview, depending on which End of Interview action the user defined HotDocs either sends the assembled document to the word processor or displays the **Document** tab. .

• Generate default titles for dialogs

Select this option to have HotDocs generate default titles for dialogs. The way HotDocs generates default dialog titles is by seeing if the first item in a dialog is dialog element text. If so, HotDocs displays that text as the dialog title in the interview outline. If not, HotDocs displays the prompt from the first variable and appends the word "*etc*."

• Combine default dialogs

HotDocs presents the end user with "default" dialogs containing variables that you have not (yet) explicitly linked to a dialog component. This property controls whether those default dialogs contain only a single variable each, or whether HotDocs combines all adjacent, unlinked variables into single default dialog. When a variable is linked to a dialog component, HotDocs no longer presents it in a default dialog.

In the 11.1 release, this feature is valid only for templates published to the desktop environment.

Type the text you want to use as a **Resource button label**, that the user sees when they hover their mouse cursor over the resource button during the interview, in text field **D**.

You can set further component file options in the General, Assembly, Hidden Data, HotDocs Server and Answer Upload tabs.

At a Glance: Assembly Tab (Component File Properties)

General	Interview	Assembly	Hidden D	ata (Word)	HotDocs Server	Answer Uplo	ad
Product	title:			A			
Do n	ot use answ variable nam	er files ies in summa	aries 🔽	Mark answe Enable Edit	ers in assembled do Answer at Docum	ocuments ent Preview ta	ь
Mov	e to the Typ	eHere" boo	kmark 📃	Update tabl (Desktop or	e of contents, ref ly)	erences, fields	, etc.
Unansw	ered variabl	e placeholde	r: Defaul	t	0		•
Post-as	sembly macr	o file (Word	only):	D			
Maximur	m WHILE iter	ations: 10	0	Maximu	im processing stac	k depth: 100)

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **Assembly** tab at the top of the dialog box to access the options for document assembly.

In the text field A at the top of the page you can enter a product title that will appear in the assembly window title bar. If you leave this field empty, HotDocs will use the text *HotDocs Developer 11*.

Below that are a set of six check boxes B you can tick to set more assembly options:

- **Do not use answer files** Tick this box to keep users from saving their answers or using an existing answer file for this particular template. Once selected, this option causes the first information-gathering dialog to appear immediately after the template is selected for assembly and at the end of assembly users cannot save their answers.
- Mark answers in assembled documents Tick this check box to let users see their answers highlighted in the assembled Microsoft Word document. When you do this, HotDocs marks merge fields in the Document Preview tab Users can then click the *Highlight Answers* button in the assembly window and answers will be highlighted with a special color.
- Use variable names in summaries Tick this box to have HotDocs use variable names instead of prompts when generating Question and Answer Summaries.
- Enable Edit Answer at Document Preview tab To let users edit answers while viewing the assembled document in the Document Preview tab (Microsoft Word users only) tick this check box. As users change answers, answers throughout the document (and interview) will be updated to reflect the change.
- **Move to the "TypeHere" bookmark** To position the cursor at a certain place in the document after it has been assembled check this box and create a word processor bookmark in the template. (See Position the Cursor in the Assembled Document.)
- Update table of contents, references, fields, etc Select this option to instruct HotDocs to update the document's table of contents and/or index after assembling the final word processor document. HotDocs updates these references based on any changes applied during document assembly.

The **Update table of contents, references, fields, etc** option on the Assembly tab requires a local installation of Microsoft word. This feature does not work with WordPerfect templates or templates on HotDocs Server.

From the drop-down list you can select the placeholder you would like HotDocs to insert into the assembled document in place of any unanswered questions from the interview.

If you would like to specify a Word document template where you want to store your post-assembly macros click the Rossembly macro file (Word only) field D and locate the template that contains these macros. (See PLAY "MACRO" and Specify a Word Template for Storing Post-Assembly Macros for more information.)

Below this are two number fields **E** where you can set the **Maximum WHILE iterations** and the **Maximum processing stack depth**.

To prevent HotDocs from infinitely processing a WHILE expression, which will cause HotDocs to stop responding, type a number in the number field on the left. This number represents the number of times dialogs or variables in the template or script can be looped before HotDocs stops it from doing so.

To prevent HotDocs from infinitely recursing (or processing) a computation, which will also cause HotDocs to stop responding, type a number in the right hand number field. This number represents how many

instructions you want HotDocs to allow in the processing stack. When HotDocs reaches this limit, the recursion will stop.

Recursion happens when a computation "calls" itself over and over until the desired result is achieved. For example, you can use a recursive computation to scan a text string, character by character, for a specific value. As HotDocs searches for this value, it adds information to the processing stack. If too much information gets added to this stack, HotDocs may stop responding.

There are further options available in the General, Interview, Hidden Data, HotDocs Server and Answer Upload tabs.

At a Glance: Hidden Data Tab (Component File Properties)

Component File Properties - D	emo Employme	nt Ag	reement.cmp	? 🔀
General Interview Assembly	Hidden Data (W	/ord)	HotDocs Server	Answer Upload
Abocument Data to Remove Trivisible merge ID marks Redundant metafile graph Bocument Properties to Remove Title Subject Author	nics Compa <u>n</u> y Category <u>K</u> eywords	V	nused prope <u>r</u> ty co Hype Stati	odes erlink base istics com <u>p</u> roperties
HatDocs Fields to Remove	Comments			
Answer bookmarks		V E	ditable text <u>b</u> ookm	arks
Document Markup				
Accept all tracked change	s	T	urn o <u>f</u> f change tra	cking
Remove all comments				
Bemove HTML compatibili	ty codes			
Remove unreferenced pa	rts and elements			
Remove saved proofing of	odes			
Remove _GoBack bookma	ark			
		_		
Select All			ОК	Cancel

You can open the **Hidden Data** tab of the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar then selecting the **Hidden Data** tab. The options are spilt into five categories; **Document Data A**, **Document Properties B**, **HotDocs Fields**, **Document Markup**, and **DOCX-specific**.

The first section of options A has three check boxes. You can tick these to have HotDocs remove invisible merge ID marks, redundant metafile graphics, and/or unused property codes.

In the second section of options B there eleven document properties that you can have HotDocs remove:

- Title
- Company
- Hyperlink base
- Subject

- Category
- Statistics
- Author
- Keywords
- Custom Properties
- Manager
- Comments

In the third section of options C you can choose to have HotDocs remove the following fields: answer bookmarks and/or editable text bookmarks.

In the forth section of options **D** you can control the options for Document Markup (for more information on marking up documents see Introduction: Create HotDocs Models). You can choose to accept all tracked changes, remove all comments and/or turn off change tracking.

In the fifth section of options gou can select options specific to DOCX templates. You can choose to have HotDocs remove HTML compatibility codes, unreferenced parts and elements, saved proofing codes, and/or _GoBack bookmarks.

At the bottom of the window there are two buttons **F** that you can use to quickly **Select All** the options or **Clear All** the options in this tab.

There are further options available in the General, Interview, Assembly, HotDocs Server and Answer Upload tabs.

To learn more about removing hidden data follow the links below:

- Remove Hidden Data from Assembled Documents
- Using the Hidden Data Remover dialog box
- Remove Hidden Data from Word Templates

At a Glance: HotDocs Server Tab (Component File Properties)

Managing Components and Templates

eneral	Interview	Assembly	Hidden [Data (Word)	HotDocs Serve	Answer	Upload
V Enab	le template	for use with	HotDocs	Server			
Inter	wiew outline	initially sho	wing		oor to bido <i>l</i> obow	interview	utline
Peso		aitially show	ing	Allow us	ser to hide /show	resource o	
Sinal	e-nage inter	view initially			ser to turn single	-nage inter	view on/
being	g assembled	on a server)				

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **HotDocs Server** tab at the top of the dialog box to access the options for making templates compatible with HotDocs Server.

Tick the first check box A to allow you to browser test your template and publish it with HotDocs Server.

Below are six check boxes B where you can choose to set further options for the HotDocs Server interviews. You can select :

- Interview outline initially showing
- Allow user to hide/show interview outline
- Resource pane initially showing
- Allow user to hide/show resource pane
- Single-page interview initially on
• Update table of contents, references, fields, etc

The **Update table of contents, references, fields, etc** option, on the HotDocs Server tab, does not produce exactly the same results as the desktop option of the same name on the Assembly tab. Unlike in desktop HotDocs, using this feature with server is known to occasionally alter or lose advanced formatting features in the assembled document. When converting a template for use with HotDocs server, we recommend testing all features that previously relied on the desktop version of this option.

• Allow user to turn single-page interview on/off

There are further options available in the General, Interview, Assembly, Hidden Data, and Answer Upload tabs.

To learn more about setting the HotDocs Server component file properties follow the link below:

• Enable Templates for Use With HotDocs Server

At a Glance: Answer Upload Tab (Component File Properties)

Managing Components and Templates

Component File Properties - Demo Employment Agreement.cmp				
General Intervie	ew Assembly Hidden Data (Word) HotDocs Server Answer Upload			
Upload URL:	A]		
Eriendly name:	B]		
C Upload answe	ers without asking			
	<u>OK</u> Cancel]		

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **Answer Upload** tab at the top of the dialog box to access the options for uploading answer files to a Web server after assembly.

In the first text field A enter the URL of the Web page to which you want the answers uploaded and in the second text field enter a friendly name for this URL that you can remember.

To force the answers to always be uploaded without asking the user to confirm tick the check box C below.

There are further options available in the General, Interview, Assembly, Hidden Data, and HotDocs Server tabs.

To learn more about setting the answer upload component file properties follow the link below:

• Specify Options to Upload Answers

Change Component File Properties

You can change certain template and component file properties at the component file level. Component file properties allow you to, among other things:

- Specify default titles and descriptions for the template.
- Define the interview for the template.
- Control which parts of the assembly window (such as the interview outline and *End of Interview* dialog) appear for a given template.
- Enable answer-editing at the **Document Preview** tab.
- Choose which hidden data can be removed from an assembled document.
- Define properties for how the interview will be displayed in a Web browser.

When you change component file properties, those changes usually affect a single template. However, if you are sharing component files (see Use One Component File for Multiple Templates), some properties may be applied to the shared file while others may be applied to the pointed component file. (See Specify Whether Component File Properties are Shared.)

To change component file properties

- 1. Open Component Manager. (See Open and Close Component Manager.)
- 2. Click the **Component File Properties** button. The **Component File Properties** dialog box appears. There are multiple tabs in the Component File Properties dialog box, for a more detailed description of each tab follow the links below:
 - At the **General** tab you are able to edit the template title, template description, the format of the component file and assign a shared component file.
 - At the **Interview** tab you can set options relating to the HotDocs interviews created by this template including using a default or custom interview, hiding the end of interview dialog, combining default dialogs, or generating default titles.
 - At the **Assembly** tab you can decide how HotDocs will assemble the finished document, including setting the processing stack depth, choosing the placeholder for any unanswered variables and setting a product title.
 - At the **Hidden Data** tab you can decide which pieces of hidden data are removed from the document.
 - At the **HotDocs Server** tab you can enable the template to work with HotDocs Server and set options for HotDocs Server interviews.

• At the **Answer Upload** tab you can set the upload URL, friendly name and whether you would like answer files to be uploaded automatically.

Since the release of HotDocs 11.1, HotDocs no longer supports converting component files to pre-2009/10 formats.

Delete a Component from the Current Component File

You can delete a particular reference to a component in the template by highlighting the reference and pressing the **Delete** key. However, to delete a variable from the component file (so that it won't appear in variable lists), you must use Component Manager. You must also use Component Manager to delete formats, patterns, merge text, dialog element text, or other components from the current component file.

Recursion happens when a computation "calls" itself over and over until the desired result is achieved. For example, you can use a recursive computation to scan a text string, character by character, for a specific value. As HotDocs searches for this value, it adds information to the processing stack. If too much information gets added to this stack, HotDocs may stop responding.

To permanently delete a variable, format, pattern, or merge text group

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. From the **Components** list, select the variable, format, pattern, or merge text you want to delete.
- 3. Click the **Delete Components** button and click **Yes** when HotDocs confirms the removal. The component is removed. If a reference to the component still exists in the template, you must remove it or you will receive errors when you assemble the template.

To delete unused components in several templates, use Template Manager. (See Delete Components from Multiple Component Files.)

When you are viewing the component list, you can limit the list of components that are showing by clicking the **Components** drop-down button and selecting the specific component type.

Copy Components From One File to Another

You can copy components from one component file into another. Copying components makes it easy to use the same variables, dialogs, example formats, and so forth in several templates. Once you have copied a variable from one component file into another, you can change its properties to fit that specific template.

When you copy a component such as a dialog or a Computation variable, all of the variables associated with the component are automatically copied as well.

To copy components from one component file to another

- 1. At the template, click the **Component Manager** button. The **Component Manager** window appears.
- 2. Click the **Expand** button in the upper-right corner of Component Manager. The window expands to show a second component list.
- 3. Click the **Open** button. The **Open** dialog box appears. (You can also click the **Other component file** drop-down button and select one of the component files that is saved in the same folder as the current file.)
- 4. Locate the component file you want to display and click **OK**. HotDocs shows a list of components in that file.
- 5. In either the current component file or the other component file, select the components you want to copy.
- 6. Click the **Copy** button (or the **Copy** button). HotDocs copies the components. If you copy a dialog or a Computation variable, any variables referenced by that component will also be copied.
- 7. If a component already exists in the file to which you are copying and contains properties that are different, complete any of the following steps at the **Copy Components** dialog box:
 - Click **Overwrite** to overwrite that specific instance of an existing component with the one you are copying.
 - Click **Don't Overwrite** to not copy a specific component.
 - Click **Rename** to assign a new name to the component you are copying. Both the original and the copied component are saved in the component file.
 - Click **Always Overwrite** to overwrite all existing components with the ones you are copying. Once you select this option, you will not be warned when other duplicate components are found.
 - Click **Never Overwrite** to tell HotDocs to not copy (and overwrite) any existing components with ones you are copying.

Also, at the **Copy Components** dialog box, you can view the components you are copying. To view the component in the file from which you are copying, click the first **View** button. To see the component that exists in the component file into which you are copying, click the second **View** button.

Once you have copied the components, click the **Collapse** button in the upper-right corner of Component Manager to show only the current list of components.

To copy components between several component files at once, you can use Template Manager. (See Copy and Paste Components Across Multiple Component Files.)

Rename Components in a Single Template

If you have a template that contains several components that must be renamed, you should use Template Manager, which changes the name in both the template file and the component file. It also updates any references to the component if it is used by other components. (See Rename Components Across Multiple Component Files.)

You can rename variables, dialogs, example formats, merge text values, dialog elements, and so forth. When you rename a component, it is changed everywhere it is used in the component file. For example, renaming a variable will update all references to it in dialogs, scripts, and prompts. However, you must update references to the component in the template text itself. This may include removing the old variable field and inserting a new variable field.

How you rename components depends on the component type. Main components can be renamed in both the template file and the component file; however supplemental components (such as example formats, patterns, and so forth) can only be renamed using Component Manager.

To rename a component while editing the template

- 1. Place the cursor in the variable field and click the ****Variable Field** button. The **Variable Field** dialog box appears.
- 2. Click the **dit Component** button. The Variable Editor appears.
- 3. In the **Variable name** field, type the new variable name and click **OK**. HotDocs verifies that you want to rename the component.
- 4. Click Yes. The variable component is renamed, and the Variable Field dialog box appears again.
- 5. Click **OK**.
- 6. Remove any other references to the variable in the template and insert the new variable field. (See Insert a Variable Field in a Text Template.)

To rename a single component using Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Select the component from the Components list and click the Rename Component button.
 The Rename Component dialog box appears.
- 3. Type the new component name in the **New name** field and click **Rename**. HotDocs changes the name throughout the component file.

4. In the template, replace old component references with the new component. (See Insert a Variable Field in a Text Template.)

To rename multiple components using Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Select the components from the Components list and click the thRename Component button. The Rename Components dialog box appears.
- 3. In the **Existing Component Names** column, select the components you want to rename.
- 4. Type the new name in the **New Component Names** column.
- 5. Optionally, to create multiple like-named components at once (for example, to rename two variables that use the word *Plaintiff* so the new names instead use *Client*), type the existing word or phrase (i.e. *Plaintiff*) in the **Replace** field, and then type the new word or phase (i.e. *Client*) in the **with** field, and click **Apply**. Where applicable, HotDocs changes all the names in the **New Component Names** column.
- 6. Click **Rename** to rename the components.
- 7. In the template, replace old component references with the new components. (See Insert a Variable Field in a Text Template.)

HotDocs lists all the components that were renamed in the **Renamed Components** list of Component Manager. To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

At a Glance: The Duplicate Variables dialog Box

Dup	licate Variables		? 💌	
0	Existing Variables	Type	New Variables	
Employee Name		Text	Employee Name	
Employee to Complete Trial Period		True/False	Employee to Complete Trial Period	
1	Replace B with	G	in new variable names.	
		0	Duplicate	

After opening the **Component Manage**r from the HotDocs Library and highlighting two or more variables, you can click on the **Duplicate Variable** button on the toolbar to open the **Duplicate Variables** dialog box where you can set the options for duplicating multiple variables.

In the List A you can see the variables you have selected in the **Existing Variables** column. The check-box column indicates which variables will be duplicated. In the **New Variables** column you can see a list of the new variables you will be creating. You must change these variables' names, either by typing each new name in the New Variables column, or by "batch" replacing existing words or phrases in the variable name with the new words. You do this by using the options below.

In the **Replace** field **B** you can enter the word or phrase in the new variable name you want to replace. Then in the **With** field **C** to the right you can enter the new word or phrase in the new variable name you want to use. For example, if you want to duplicate the variables Plaintiff Name, Plaintiff Address, and Plaintiff Gender to create Defendant Name, Defendant Address and Defendant Gender, you can type Plaintiff in the Replace box, and then type Defendant in this box.

You can then click the **Apply** button **D** and replace existing words or phrases in the **New Variables** column with the new word or phrase you want to use. Once replaced, you can click the **Duplicate** button **D** and HotDocs will create the new variables using the names you have suggested.

To learn more about duplicating multiple components in component manager follow the link below:

• Make a Duplicate Copy of a Variable

Make a Duplicate Copy of a Variable

You can copy an existing variable to create a new variable. This may be useful, for example, if you want to create a variable that uses most (if not all) of the same properties of an existing variable, but you want the variable to have a different name.

You must use Component Manager to duplicate variables. You can duplicate one component or many, depending on your needs. You cannot duplicate dialogs, clauses, or database components.

To duplicate a single variable

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manage**r window appears.
- 2. Select the variable you want to copy and click the **Duplicate Variables** button. HotDocs opens the **Variable Editor** and displays a copy of the variable (as shown in the **Variable name** field).
- 3. Change the name and any other properties of the variable and click **OK**. The new variable is added to the component file.

To duplicate multiple variables at once

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Select the variables you want to create copies of and click the **Duplicate Variables** button. The **Duplicate Variables** dialog box appears.
- 3. In the **Existing Variables** column, select the variable(s) you want to duplicate.
- 4. Type the name of the new variable in the New Variables column.
- 5. Optionally, to create multiple like-named components at once (for example, to copy *Defendant*-related components to create *Plaintiff*-related components), type the existing word or phrase (i.e. *Defendant*) in the **Replace** field, and then type the new word or phase (i.e. *Plaintiff*) in the **with** field, and click **Apply**. Where applicable, HotDocs changes all of the names in the **New Variables** column.
- 6. Click **Duplicate** to create the new variables.

HotDocs lists all the variables that were duplicated in the **Duplicated Components** list of Component Manager. To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

Import or Export Computation Variables

You can copy variables between your own component files by using the **Expand** button (See Copy Components From One File to Another) but if you need to pass a computation on to another person, perhaps someone off-site or in another branch you will need to export the computation from the

Component Manager. This creates a piece of HotDocs readable syntax which, when imported back into a Component Manager, will add the whole computation to the component file.

Export a Computation to the Clipboard

- 1. Open Component Manager for the template you wish to copy a computation variable from. (See Open and Close Component Manager.) The **Component Manage**r window appears.
- 2. Identify the Computation you wish to import and click on it once to highlight it.
- 3. Click on the **Export a Computation to the Clipboard** button.
- 4. You can then paste the computation variable syntax into a text file by type ctrl-v or paste by right clicking with the mouse.

Import a Computation from the Clipboard

- 1. Select the entire computation variable syntax and type ctrl-c or select copy by right clicking on the mouse. This puts the information back onto the Clipboard.
- 2. Open Component Manager for the template you wish to add the computation variable to. (See Open and Close Component Manager.) The **Component Manage**r window appears.
- 3. Click the **Import a Computation from the Clipboard** button then you will see the new computation appear in the Components List.

If you copy anything else between exporting and importing or pasting into a text file you will loose the exported computation and have to re-start the process.

Edit Formats, Merge Text, Dialog Elements, and Patterns

You can edit supplemental components such as example formats, dialog elements, merge text, and patterns using Component Manager.

To edit supplemental components

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Click the Components drop-down button and select the type of supplemental component you want to edit. Your options include variable formats (like Text Formats and Number Formats), Merge Text, Dialog Elements, and Text Patterns. The list is filtered to show that kind of component.
- 3. Select the component you want to edit and click *d* **Edit Component**. The component is opened in its component editor.

4. Make any necessary changes.

To create a new example format, merge text group, dialog element text, or text pattern, click the **New Component** button while viewing that specific list of supplemental components at Component Manager.

If you want to assign a name to a variable format—for instance, because a name would be more descriptive than the actual format—type the name in the **Format name** field. This name will be displayed in format lists instead of the format itself.

At a Glance: The Find In Components dialog box

Find in Components		? 💌
All components in list Selected components in list All components in component file		
Find What	B	
2 3 4		
Match case Find whole words or	nly	
 Look in Display text (titles, prompts, etc.) Merge text, patterns, and formats 	✓ <u>R</u> esources ✓ <u>S</u> cripts	✓ Notes ✓ Variable references
		Eind Cancel

After opening the **Component Manager** from the HotDocs Library you can click on the **Hind** button on the toolbar to open the **Find in Components** dialog box where you can set the options for finding components in the component file.

From the **Include** multiple choice list <mark>A</mark> you can select one of the following options: Search all components in the list (list can be filtered by component type prior to clicking the **Find** button), search

selected components only (available if you have highlighted components before clicking the **Find** button, or search all components in the component file.

In the **Find What** field **b**elow you can enter the terms you need to search for. Below that are two check boxes **c** where you can decide if you want HotDocs to match the case of your terms and whether it should find whole words only.

At the bottom of the dialog is the **Look In** check box list **D** where you can set options to limit where HotDocs searches for your terms. You can select as many as you need from the following:

- **Display text (titles, prompts, etc.):** Searches for the text in variable prompts, dialog titles, and dialog element text.
- **Resources:** Searches for the text in variable and dialog resources. (This only searches plain-text resources.)
- Notes: Searches for the text in each component editor's Notes tab.
- **Merge text, patterns, and formats:** Searches for the text in merge text options and text variable patterns. It also searches variable example formats.
- **Scripts:** Searches for the text in literal text strings (which are used in computation and dialog scripts).
- **Variable references:** Searches for variables that are referenced in the components' prompts, merge texts, and scripts.

To learn more about finding components in component manager follow the link below:

• Search Component Files

At a Glance: The Find and Replace Components dialog box

Find and Replace in Components	? 🗙
Include All components in list Selected components in list All components in component file	
Find What 1 2 3 4	Replace With
5 Match <u>c</u> ase Find <u>w</u> hole words only Look in Display text (titles, prompts, etc.)	Confirm replacements
<u>I</u> <u>M</u> erge text, patterns, and formats	Eind Cancel

After opening the **Component Manager** from the HotDocs Library you can click on the description on the toolbar to open the **Find in Components** dialog box where you can set the options for finding components in the component file.

From the **Include** multiple choice list A you can select one of the following options: Search all components in the list (list can be filtered by component type prior to clicking the **Find** button), search selected components only (available if you have highlighted components before clicking the **Find** button, or search all components in the component file.

In the **Find What/ Replace With** field B below you can enter the terms you need to search for and what you would like to replace them with in their relative columns. Below that are three check boxes where you can decide if you want HotDocs to match the case of your terms, whether it should find whole words only, and whether it should have you confirm the replacements.

At the bottom of the dialog is the **Look In** check box list **D** where you can set options to limit where HotDocs searches for your terms. You can select as many as you need from the following:

- **Display text (titles, prompts, etc.):** Searches for and replaces the text string in variable prompts, dialog titles, and dialog element text.
- **Resources:** Searches variable and dialog plain-text resources and replaces the text string with the text you specify.

- **Notes:** Searches for and replaces instances of the text string in each component editor's Notes tab.
- **Merge text, patterns, and formats:** Searches for and replaces instances of the text string in merge text options and text variable patterns. It also searches and replaces text strings in variable example formats.
- **Scripts:** Searches for the text in literal text strings (which are used in computation and dialog scripts) and replaces any text that is found with the text specified.

To learn more about finding components in component manager follow the link below:

• Search Component Files

Search Component Files

Using Component Manager, you can search for literal text strings in component prompts, titles, dialog element text, notes, scripts, patterns, formats, and merge text. If necessary, you can replace any text you find with different text.

To search components for a certain text string

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Click the **H** Find in Components button. The Find in Components dialog box appears.
- From the Include list, select one of the following options: All components in list (searches all of the components currently displayed in the Components list), Selected components in list (searches only the components that are currently selected (highlighted) in the Components list), All components in component file (searches all of the components in the component file, regardless of what is currently displayed (and selected) in the Components list).
- 4. Type the word or string of text in the Find what column. (You can include up to 10 text strings.)
- 5. Optionally, select any other search criteria:
 - Match case searches for a text string that matches the capitalization you have used.
 - Find whole words only excludes instances where the text in the Find column occurs as part of another word.
- 6. Specify which properties of a component you want to search by selecting an option. (See Component Properties for an explanation of each property.)
- 7. Click Find. Once HotDocs is finished searching the components for the specified string, Component Manager displays the Found Components list, which shows only those components that contained a match. This list continues to show the results of your search until you either perform a new search or you close Component Manager. To view the full list of components again, click the Components drop-down button and choose All Components.

To search components for a certain text string and then replace the string with new text

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Click the ^{ab}/_{ac} Find and Replace in Components button. The Find and Replace in Components dialog box appears.
- From the Include list, select one of the following options: All components in list (searches all of the components currently displayed in the Components list), Selected components in list (searches only the components that are currently selected (highlighted) in the Components list), All components in component file (searches all of the components in Component Manager, regardless of what is currently displayed (and selected) in the Components list).
- 4. Type the text string for which you want to search in the **Find what** column. (You can include up to 10 strings.)
- 5. Type the text string with which you want to replace the found text in the **Replace with** column.
- 6. Optionally, select any other search criteria:
 - Match case searches for a text string that matches the capitalization you have used.
 - **Find whole words only** excludes instances where the text in the **Find** column occurs as part of another word.
 - Confirm replacements makes HotDocs ask you before it replaces the text string.
- 7. Specify which properties of a component you want to search by selecting an option. (See Component Properties for an explanation of each property.)
- 8. Click **Find.** If you selected **Confirm replacements**, HotDocs will confirm each replacement. Once HotDocs finishes replacing the components for the specified string, Component Manager displays the **Found Components** list, which shows only those components that were modified. This list continues to show the results of your search and replace until you either perform a new search or you close Component Manager. To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

To search for a specific component based on the component name, select the **Find** check box (located at the bottom of the Component Manager window) and type the text for which you are searching.

At a Glance: The Spell Check Components dialog box



After opening the **Component Manager** from the HotDocs Library you can click on the **Spell Check** button on the toolbar to open the **Spell Check Components** dialog box where you can set the options for spell checking the components in the selected template.

From the **Include** multiple choice list A you can select one of the following options for spell checking: Spell check all components in the list (list can be filtered by component type prior to clicking the **Spell Check** button), spell check selected components only (available if you have highlighted components before clicking the **Spell Check** button, or spell check all components in the component file.

You can then choose as many or as few options as you would like from the **Properties to Check** check box list **B**. Select what you would like HotDocs to check from the following options:

- **Display text (titles, prompts etc):** Checks the spelling of dialog titles and element text, as well as variable prompts and multiple-choice options.
- **Merge text:** Checks the spelling in any merge text options that have been specified for Multiple Choice variables.
- **Plain text resources:** Checks the spelling in any plain-text resources that have been assigned to the selected components.
- **Scripts:** Checks the spelling of any literal text (or text between quotation marks) that is used in a computation script.

To learn more about spell checking components from the component manager follow the link below:

• Spell Check Components

Spell Check Components

Using Component Manager, you can spell check text-based properties of components, such as prompts, titles, notes, and dialog element text.

To spell check components

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manage**r window appears.
- 2. Click the **Spell Check Components** button. The **Spell Check Components** dialog box appears.
- 3. From the **Include** list, select one of the following options: **All components in list**, **Selected components in list**, or **All components in component file**.
- 4. Select the component properties you want to spell check. (See Component Properties.)
- 5. Click **OK**. If HotDocs finds any unrecognized words, it displays the **Check Spelling** dialog box.
- 6. Make your selection, based on information in the following table:

То	Do This
Ignore the current instance of the word and continue spell checking	Click Ignore.
Ignore all instances of the word and continue spell checking	Click Ignore All .
Correct only the current instance of the word and continue spell checking	Select an existing replacement from the Change to list (or type the replacement in the Change to field) and click Change .
Correct all instances of the misspelled word and continue spell checking	Select an existing replacement from the Change to list (or type the replacement in the Change to field) and click Change All .
Add the word to your personal dictionary so that the spelling checker will not question the word again	Click Add .
Display additional spelling	Click Suggest.
word	A deeper search takes longer but produces better possible replacements. If the button is unavailable, HotDocs is searching at the deepest level.
Change your spell checking options, such as which words the spelling checker looks at and how it determines whether a word is a possible replacement	Click Options . (You can also change your options at the HotDocs Options dialog box. See Change Your Spell Checking Options.)

When you are finished spell checking, HotDocs displays a list of components that were modified in the **Changed Components** list. If HotDocs does not find any misspelled words, or if you do not correct any misspellings, this list is empty. To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

At a Glance: The Print Components dialog box



After opening the **Component Manager** from the HotDocs Library you can click on the **Print** button on the toolbar to open the **Print Components** dialog box where you can set the options for printing a list of the components referenced in the selected template.

From the **Include** multiple choice list A you can select one of the following options for printing: Print all components in the list (list can be filtered by component type prior to clicking the **Print** button), print selected components only (available if you have highlighted components before clicking the **Print** button, or print all components in the component file.

You can then choose as many or as few options as you would like from the **Properties to Print** check box list **B**. Select what you would like HotDocs to print from the following options:

- Component types: Prints the component types of the selected component.
- **Titles, prompts, and dialog elements:** Prints any assigned variable prompts, dialog titles, and dialog element text.

- **Scripts:** Prints computation and dialog scripts.
- Plain text resources: Prints any plain-text resources assigned to variables or dialogs.
- **Notes:** Prints any notes associated with the selected components.
- All other component type-specific properties: Prints component-specific data about the variable or dialog.

To learn more about printing component lists from the component manager follow the link below:

• Print a List of Components

Print a List of Components

Using Component Manager, you can print lists of properties for selected components.

To print a list of properties for selected components

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Click the **Print Components** button. The **Print Components** dialog box appears.
- 3. From the **Include** list, select one of the following options:
 - All components in list, which prints all of the components currently displayed in the **Components** list.
 - Selected components in list, which prints only the components that are currently selected (highlighted) in the **Components** list.
 - All components in component file, which prints all of the components in Component Manager, regardless of what is currently displayed (and selected) in the **Components list.**
- 4. Select the properties you want to print. (See Component Properties for an explanation of each property.)
- 5. Click **OK** and specify any printer options.

Using One Component File for Multiple Templates

Be sure to back up your shared component files on a regular basis. If a shared component file becomes damaged, you could lose all of the components for several templates.

When you create a template file, HotDocs automatically creates a component file for that template. The component file stores variables, dialogs, and other components you use for that specific template.

When several related templates use many of the same components, you can create all of those components just once and store them in a shared component file. When you make a change to a shared component, (for example, if you change a prompt or resource text), the change appears in all templates that use that component file.

The way you share the same components with multiple templates is by pointing each template's own component file to the shared component file. Then, as you create and edit components within the template, the template bypasses its own component file (which is now known as the pointed component file or *actual component file*) and stores the information in the shared component file instead.

You must save all templates, pointed component files, and shared component files in the same folder.

It is best practice to point all the templates you want to share a component file to the shared component file *before* you start creating components in any of the templates. That way, HotDocs saves the components directly in the shared component file so they are ready for use when you automate other templates. Otherwise, you must copy those components to the shared file so HotDocs can find them.

Pointing the current component file to the shared file does not automatically copy the components there.

When specifying component file properties, HotDocs uses properties from the shared component file for all templates that point to it (except for those specified at the **General** tab of the **Component File Properties** dialog box). You can, however, use the pointed component file's properties. Additionally, if HotDocs has specified some of the properties you want to use in the shared component file, you can copy them to the pointed component file.

If you are pointing several templates to a shared component file, but you want each template to use its own custom interview, at the **Interview** tab, clear **Use interview properties stored in the shared component file**, select **Use custom interview**, and then specify the name of the interview computation in the **Interview component** text box.

You can unpoint a component file using Component Manager.

You can point new templates to as shared component file as you make them but you can also point pre-existing templates to a shared component file:

To point the current component file to a shared component file

- 1. Make sure you have closed all templates and component editors. (The only window you should leave open is the template library.)
- 2. At the template library, select the template whose component file you want to point to the shared component file.

- 3. Click the **Component Manager** button.
- 4. Click the **Component File Properties** button.
- 5. At the **General** tab, click the **Shared component file** drop-down button and select the shared component file. (This drop-down list shows all the component files listed in the same folder as the actual component file. Remember, you must save pointed and shared component files to the same folder.)
- 6. Optionally, to use the properties of the pointed component file instead of the shared, clear Use properties stored in the shared component file at each respective tab of the Component File Properties dialog box. To copy the properties of the shared component file into the pointed component file, click Copy Shared. (HotDocs will not update any properties you change after choosing to use the pointed component file in the shared file.)
- Click OK. The current template's component file is now pointed and the contents of the shared component file appear in the component list. (At Component Manager, the Component File Properties button changes appearance to indicate that the component file is now pointed.)

If you created variables before you pointed the component file to the shared file, you must copy those variables to the shared file.

HotDocs can display a warning when you edit a template that has a pointed component file. This warning reminds you that changes you make will affect all templates that point to the component file you are editing.

Insert a PLAY Instruction in a Template

To insert a PLAY instruction in a template

- 1. At the template, position the cursor in the template where you want the PLAY instruction.
- 2. If you are using Microsoft Word:
 - Click the **HotDocs** menu in the HotDocs toolbar and choose **Other Field** from the list of options. The **Other Field** dialog box appears.
 - Click the Field type drop-down button and choose PLAY.
 - In the **Macro name** field, enter the name of the macro you want to run.
 - Click **OK**. The instruction is inserted in the template.
- 3. If you are using WordPerfect:
 - Copy an existing variable or instruction field in the template.
 - Replace text between the chevrons (« ») with the PLAY instruction.

See Specify a Word Template for Storing Post-Assembly Macros for more information.

PLAY instructions are executed when you create an actual document from the assembly. This includes sending the document to the word processor, saving the document, or printing a copy of the document. If there are multiple instructions, they are processed in the order they are encountered.

Specify Whether Component File Properties are Shared

When several related templates use many of the same components, you can create all of those components just once and store them in a *shared component file*. The way you share the same components in multiple templates is by *pointing* each template's own component file to the shared component file. Then, as you create and edit components within the template, the template bypasses its own component file (which is now known as the *pointed component file* or *actual component file*) and stores the information in the shared component file instead.

When specifying component file properties (see Change Component File Properties), you can choose which component file will provide the options for the template. By default, templates use the shared component file's properties (except for those properties specified at the **General** tab of the **Component File Properties** dialog box). However, if a certain number of templates in your set require their properties be different from the others, you can choose to use the pointed component file's properties, instead.

For example, perhaps only a few of your pointed templates will be published for use with HotDocs Server. For those specific templates, you can enable them for use with HotDocs Server and set specific properties so they can be used on the Web.

Properties (such as the **Template title** and **Template description**) specified at the **General** tab of the **Component File Properties** dialog box are always stored in the pointed component file.

To choose which component file properties should be used for a pointed template

- 1. Point the template's component file. See Use One Component File for Multiple Templates.
- 2. With Component Manager open and the **Component File Properties** dialog box displayed, click the tab for the properties you want to use from the pointed component file. (For example, to use the pointed file's interview options, click the **Interview** tab.) The view changes to show those specific options.
- 3. Clear Use properties stored in the shared component file.
- 4. Optionally, to copy the properties of the shared component file into the pointed component file, click **Copy Shared**. Any properties specified for this group at the shared component file are copied into the current (or pointed) component file.

The ability to use pointed component file properties for templates also lets you specify custom interviews for each template in your set. Specifically, if you are pointing several templates to a shared component file, but you want each template to use its own custom interview, at the

Interview tab, clear **Use Interview properties stored in the shared component file**, select **Use custom interview**, and then specify the name of the interview computation in the **Interview component** field.

Any properties you change after choosing to use the pointed component file will not be updated in the shared file.

Make Templates Stop Sharing Component Files

You can unpoint a component file if you decide you no longer want to share components between multiple templates. (See Use One Component File for Multiple Templates.)

To unpoint a component file

- 1. Make sure all templates and component editors are closed. (The only thing that should be open is the template library.)
- 2. At the template library, select the template whose component file you want to unpoint.
- 3. Click the **Component Manager** button. The **Component Manager** window appears.
- 4. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 5. At the **General** tab, click the **Shared component file** drop-down button and choose the blank line at the top of the list.
- 6. Click **OK**. The component list changes to show the components used in the original component file.

If you added or created components while the component file was pointed, you must copy those components back to your original component file so HotDocs can find them. See Copy Components From One File to Another for information on how to do this.

Control Whether Users See the End of Interview Dialog

By default, when users complete interviews, the last dialog they see is the *End of Interview* dialog, which is where they choose what to do with the assembled document. In some situations, you may want to control whether users view the assembled document automatically without displaying these options. You can keep this dialog from appearing.

To hide the End of Interview dialog

- 1. Open Component Manager for the template whose *End of Interview* dialog you want to hide. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 3. Click the Interview tab and select Hide End of Interview dialog.

When the *End of Interview* dialog is hidden and the user clicks **Next** at the last dialog in the interview, HotDocs will perform the *End of Interview* action the user specifies at **HotDocs Options**. These options include sending the document to the word processor or Filler, or viewing the assembled document at the **Document Preview** tab. (See Control What Happens When You Finish an Interview.)

Users can choose which options appear in the *End of Interview* dialog by making their selections at **HotDocs Options**. See Customize the End of Interview Dialog.

Position the Cursor in the Assembled Document

You can cause HotDocs to place the cursor at a specified place in the assembled document once the user has completed an interview and sent the assembled document to the word processor. You do this by inserting a "TypeHere" bookmark in your template and then by selecting a component file property.

To insert a TypeHere bookmark

- 1. Create a template or open an existing template for editing. (See Create a New Text Template File or Edit a Template.)
- 2. Place the cursor in the template where you want your cursor to appear when the assembled document is sent to the word processor.
- 3. Use your word processor's functionality to create a bookmark. Name it **TypeHere** and insert it into the text. (See your word processor's help file for information on creating bookmarks.)
- 4. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 5. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 6. Click the **Assembly** tab and select **Move to the "Type Here" bookmark**.
- 7. Click **OK**.

Specify a Word Template for Storing Post-Assembly Macros

You can insert PLAY instructions in templates to play macros after assembly is complete. Such macros are used to update references in the document, remove unused markup coding, apply custom formatting to answers, and so forth.

With a DOTM or DOT template, the macro to be PLAYed can simply be included in the HotDocs template itself. After assembly, HotDocs temporarily attaches the original template to the assembled document so that when the PLAY instruction is processed, Word automatically finds the macro to be PLAYed.

With a DOCX or RTF template, however, you can't store macros in the template. While you can place macros to be PLAYed in a global template saved in Word's *Startup* folder, or add the macros to *Normal.dot*, this can be inconvenient, since you must either provide instructions for the user on how to save the global template to the *Startup* folder or you must instruct them on how to modify *Normal.dot*.

To accommodate this, you can store your post-assembly macros in a Word template, which you can then associate with the HotDocs template. You specify the name of this Word template file at the **Component File Properties** dialog box for the template. This Word template must be stored in the same folder as the HotDocs template that uses it.

To use a post-assembly macro template

- 1. Create a Word DOTM or DOT template and store your macro in it. (See the Microsoft Word documentation for instructions on doing this.)
- 2. Edit the template to which you want to attach the macro. (See Edit a Template.)
- 3. Insert the PLAY MACRO instruction in the template. (See PLAY "MACRO".)
- 4. Open the **Component File Properties** dialog box for the template. (See Change Component File Properties.)
- 5. Click the **Assembly** tab, and, in the **Post-assembly macro file** field, enter the name of the template you created in Step 1. (Remember, the Word macro template must be stored in the same folder as the HotDocs template.)

Post-assembly macros are played whenever the user creates a document from the template, specifically when the user prints the document, saves the document to disk, or sends the document to the word processor.

At a Glance: The Hidden Data Remover dialog box

Hidden Data Remover		? 🗙		
		1 template selected		
A Document Data				
Invisible merge ID mar	ks			
🔲 Redundant metafile gr	ap <u>h</u> ics			
BDocument Properties				
Title	Company	Hyperlink base		
Subject	Category	Statistics		
Author	Keywords	Custom properties		
Manager	Comments			
COocument Markup				
Accept all tracked char	Accept all tracked changes			
Remove all comments	Remove all comments			
DOCX-specific				
Remove HTML compatibility codes				
Remove unreferenced parts and elements				
Remove saved proofing codes				
Remove _GoBack bookmark				
Select <u>A</u> ll Clear	All An	aly <u>z</u> e Remove <u>D</u> ata <u>C</u> ancel		

You can open the **Hidden Data Remover** dialog box directly from the **Tools** menu in the library, after selecting the template you would like to change.

You can use the check boxes in this dialog box to select which pieces of data you would like HotDocs to remove. The options are spilt into four categories; **Document Data** A, **Document Properties** B, **Document Markup** C, and **DOCX-specific** D.

The first section of options A has two check boxes. You can tick these to have HotDocs remove invisible merge ID marks and/or redundant metafile graphics..

In the second section of options B there eleven document properties that you can have HotDocs remove:

- Title
- Company
- Hyperlink base
- Subject
- Category
- Statistics
- Author
- Keywords

- Custom Properties
- Manager
- Comments

In the forth section of options **G** you can control the options for Document Markup (for more information on marking up documents see Introduction: Create HotDocs Models). You can choose to accept all tracked changes, remove all comments and/or turn off change tracking.

In the fifth section of options **D** you can select options specific to DOCX templates. You can choose to have HotDocs remove HTML compatibility codes, unreferenced parts and elements, saved proofing codes, and/or _GoBack bookmarks.

At the bottom of the dialog box gou can choose to **Select All** check boxes, **Clear All** check boxes, **Analyze** the data, or, when you are happy with your selection, **Remove Data**. You can also choose **Cancel** to return to the library without making any changes.

To learn more about removing hidden data follow the links below:

- Remove Hidden Data from Assembled Documents
- Using the Hidden Data Remover dialog box
- Remove Hidden Data from Word Templates

Remove Hidden Data from Assembled Documents

Removing hidden data from an assembled document is supported in Microsoft Word only.

As you work in Microsoft Word templates or documents, some personal information as well as some hidden document properties are stored in the document. This information may compromise the security of the assembled document as well as affect the size of the document file. To keep others from reviewing this information, or to reduce the size of the file, you can choose which of these Word properties should be removed from an assembled document.

Properties you select at the **Hidden Data** tab of the **Component File Properties** dialog box will be removed from the assembled document when users send the document to Microsoft Word.

In addition to removing hidden data from assembled documents, you can also remove hidden data from Word templates you're automating. See Remove Hidden Data from Word Templates for details.

To choose which properties will be removed from an assembled Word document

- 1. Open Component Manager. (See Open and Close Component Manager.)
- 2. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 3. Click the **Hidden Data (Word)** tab. The view changes to show the properties you can remove.
- 4. Select the types of data you want removed from the assembled document. You can remove content as well as properties from the assembled document. Specifically:
 - Options in the **Document Data to Remove** group include marks or other types of data Word uses to manage information in the document:
 - Invisible merge ID marks: Word merges revision ID numbers in the template each time you edit the template text. (Word uses these numbers to improve accuracy when merging or comparing related documents.) Each time you edit the text of a template, more IDs are added to the template and the size of the file increases. Removing these IDs reduces the size of the document.
 - Redundant metafile graphics: When you insert a graphic image in an RTF template, Word merges a Windows Metafile copy of the graphic in the template as well. This means there are two versions of the graphic in the template. Removing the metafile version can significantly reduce the size of the document.
 - Unused property codes: When you apply formatting properties (such as bold, underline, alignment, etc.) to text in a template, these codes are stored in the template. Selecting this property removes any codes that may be left in the assembled document after the section of text to which the property was applied has been removed through the use of IF instructions.
 - Options in the **Document Properties to Remove** group include information stored about the document, including information about who created the document. (Document properties typically appear on the **Properties** dialog box for a given document.)
 - Options in the **HotDocs Fields to Remove** group include the bookmarks HotDocs merges in a document to indicate editable answers and text.
 - Options in the **Document Markup** group include reviewer's comments, such as annotations or change-tracking suggestions.

Now, when the document is assembled and sent to the word processor, the data associated with each option you've selected will be removed from the document.

Using the Hidden Data Remover dialog box

This option is available for Microsoft Word RTF and DOCX templates only.

To access this dialog box: Select a template or multiple templates in the template library. > Choose **Hidden Data Remover** (**Tools** menu).

By default, when working with Word templates, Word stores extra information about changes you make to the template. Sometimes tracking these changes causes the size of the file with which you are working to increase. In some instances, you can keep Word from saving this information. Specifically, you can keep Word from saving random merge ID numbers as well as redundant metafile graphics.

Invisible Merge ID Marks

According to the Microsoft Word 2003 help file, Word uses "randomly generated numbers to help keep track of related documents for comparing and merging. Although these numbers are hidden, they could potentially be used to demonstrate that two documents are related." Each time you edit template text, these merge ID numbers are generated and saved to the template. If you frequently change the contents of a template, the size of the template will increase. If you rarely or never merge or compare templates, you can keep Word from generating these ID numbers.

To keep Word from generating merge ID numbers

- 1. Open Microsoft Word.
- 2. Choose **Options** from the Word **Tools** menu. The **Options** dialog box appears.
- 3. Click the **Security** tab. The view changes to show the different security options you can choose for a template.
- 4. In the **Privacy** options group, clear **Store random number to improve merge accuracy**.

Clearing this option is only applicable in Word 2002 and later.

- Word 2007 users can access this setting by going to the Microsoft Office Button > Word
 Options > Trust Center > Trust Center Settings > Privacy Options > Store random number to improve Combine accuracy.
- Word 2010/13 users can access this setting by going to File > Options > Trust Center > Trust Center Settings > Privacy Options > Store random number to improve Combine accuracy

Redundant Metafile Graphics

According to a Microsoft Knowledge Base article, "if an EMF, a PNG, a GIF, or a JPEG graphic is inserted into a Word document, when the document is saved, two copies of the graphic are saved in the document. Graphics are saved in the applicable EMF, PNG, GIF, or JPEG format and are also converted to WMF (Windows Metafile) format."

To resolve the problem, you can specify a Windows registry setting that keeps Word from saving two copies of the image.

You must be extremely careful when working in the Windows registry. Failure to follow the instructions below exactly could result in your making changes that negatively affect all of the

programs on your computer. You may want to ask your system administrator for help if you are unsure of what you are doing.

To keep Word from saving two copies of graphics in your templates

- 1. Close Word (if it's running.)
- 2. Click **Start > Run**. The **Run** dialog box appears.
- 3. In the **Open** field, type **regedit** and click **OK**. The **Registry Editor** opens.
- Navigate to HKEY_CURRENT_USER > Software > Microsoft > Office > Version > Word > Options. (Replace Version with the version of Word you are using.)
- 5. In the right pane of the window, right-click and choose **New > String Value** from the shortcut menu. A new string is created called **New Value #1**.
- 6. Rename the value to **ExportPictureWithMetafile**. (To do this, you can choose **Rename** from the shortcut menu.)
- 7. Once the name is changed, right-click on the string and choose **Modify**. The **Edit String** dialog box appears.
- 8. In the **Value data** field, enter **0**.
- 9. Click **OK**.

To remove merge ID numbers or extra metafile graphics from existing templates, see Remove Hidden Data from Word Templates for details. To allow this data in templates but remove it from assembled documents, see Remove Hidden Data from Assembled Documents.

Remove Hidden Data from Word Templates

Removing hidden data is supported in Microsoft Word RTF and DOCX templates only.

As you work in Microsoft Word templates, some personal information as well as some hidden document properties are stored in the template. This information may compromise the security of the template as well as affect the size of the template. To keep others from reviewing this information, or to reduce the size of the file, you can choose which of these Word properties should be removed from the template. You can remove this data from a single template or from multiple templates at a time.

You can also remove hidden data from assembled Word documents. See Remove Hidden Data from Assembled Documents for details.

To remove hidden data from Word templates

- 1. Open a template library. (See Open a Library.)
- 2. Select the templates from which you want to remove the data.

- 3. Choose Hidden Data Remover (Tools menu). The Hidden Data Remover dialog box appears.
- 4. Optionally, click **Analyze**. HotDocs analyzes the templates you have selected and reports how much space in the file each option listed in this dialog box is using. (This information appears in parentheses next to each option.)
- 5. In the **Document Data** group, select any of the following options:
 - Invisible merge ID marks: Word merges revision ID numbers in the template each time you edit the template text. (Word uses these numbers to improve accuracy when merging or comparing related documents.) Each time you edit the text of a template, more IDs are added to the template and the size of the file increases. Removing these IDs reduces the size of the template file.
 - **Redundant metafile graphics:** When you insert a graphic image in an RTF template, Word merges a Windows Metafile copy of the graphic in the template as well. This means there are two versions of the graphic in the template. Removing the metafile version can significantly reduce the size of the RTF template.

Once you remove this data from a template, continued editing of the template may reintroduce the extra data back into the template. You can prevent Word from adding this data to the template. Click here for details.

- 6. In the **Document Properties** group, select which properties you want to remove from the template. (This includes information stored in the **Properties** dialog box for a given template. It usually represents information about who created the template.)
- 7. In the **Document Markup** group, select any of the following options:
 - Accept all tracked changes updates the template with any additions or corrections made with Word's Track Changes feature.
 - Turn off change tracking disables the Track Changes option.
 - **Remove all comments** removes comments made in the template using Word's Comment feature.
- 8. Once you've selected which data you want to remove, click **Remove Data**. HotDocs removes the data from the selected templates.

Work with Variables in Headers, Footers, Footnotes, and Text Boxes

Depending on whether you use Word or WordPerfect, how you insert a variable in a header, footer, footnote, or text box is different. The following table explains how to use variables in the various areas of a word processor template.

Microsoft Word

WordPerfect

Headers and footers	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.
Footnotes	Create the variable in Component Manager and drag it into the field. Edit the variable using Component Manager.	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.
Text Boxes	Variables in text boxes are ignored during document assembly.	Create the variable directly in the text box, or create the variable in Component Manager and drag it into the text box. Edit the variable directly in the box or use Component Manager.

Using buttons in the HotDocs Navigation toolbar will have no effect while editing the contents of a header or footer.

When inserting variables or instructions in a WordPerfect header, footer, footnote, or text box, you cannot highlight text and then replace it with the field. You must simply insert the variable by clicking in the text.

HotDocs 5 users: In HotDocs 5, you had to specify a component file property that instructed HotDocs to assemble variables in Word headers and footers. HotDocs 11 automatically assembles these, so the option has been removed from the Using buttons in the HotDocs Navigation toolbar will have no effect while editing the contents of a header or footer.

If a Word template contains headers and footers with variables, you may find your variables being asked "out of order." You can create a custom interview to control the order your variables and dialogs are asked. See **Define a Custom Interview** for details.

Using Template Manager

Introduction: Use Template Manager

You can use Template Manager to manage both templates and component files.

The following is an example of a template library that is being "managed" using Template Manager. A list of templates in the library appears on the left, while a list of components used in the templates appears on the right.

🖶 HotDocs Template Manager 📃 🗖 💌				
📴 토 〒 프 গ্ল 🖻 🖻 🛍 헬 Rena	ame 🔏 Delete 🛛 🔑 Focus 🛛 🗛 🗛	omponents 🔹	i 🖓 🖗	
Template Name / Component File	Component Name /	Component Type	Status	
🕀 🖹 Employment Agreement	Address Line 1	Text Variable	ОК	
🗄 🖹 Employee Personal Data	Address Line 2	Text Variable	ОК	=
🗄 🖹 Editor/Author List	31 Agreement Date	Date Variable	ОК	
🗄 🖺 Life Insurance Application	🚍 Agreement Information	Dialog	OK	
🗄 🖹 Collection Letter	🔙 AL/AK/AZ/AR/CA/CO/CT/DE	Merge Text	OK	
	2 Amount of Coverage	Number Variable	OK	
	2 Annual Salary	Number Variable	OK	
	🗛 Author Full Name	Text Variable	OK	
	🚍 Author Information	Dialog	OK	
	🖁 Authorization	True/False Variable	OK	
	🚍 Beneficiary Designation	Dialog	OK	
	=🗶 Beneficiary Filter 1	Computation Varia	OK	
	=🗶 Beneficiary Filter 2	Computation Varia	OK	
	🛕 Beneficiary Name	Text Variable	OK	
	🗛 Beneficiary Relationship	Text Variable	OK	
	A Beneficiary Telephone Number	Text Variable	OK	
	🚍 Book Information	Dialog	OK	
	🗛 Book Title	Text Variable	OK	
	A City	Text Variable	OK	
	🛄 Client Data	Database	OK	
	A Client Name	Text Variable	ОК	Ŧ
	5 templates 1	24 unique component	ts	

Manage Templates

Template Manager allows you to view a list of items referenced in a library. This view includes templates and any files referenced by the templates, including component files, inserted templates, clauses, clause libraries, and so forth. Template Manager can also report any problems it finds with files you are managing.

You can use Template Manager to convert multiple documents either to HotDocs 11 format or to DOCX. Additionally, you can use Template Manager to rename templates. Using Template Manager to do these tasks can ensure that any changes you make in one file will be reflected in any other files referenced by it.

Manage Components

Template Manager provides a comprehensive overview of component usage in templates, inserted templates, clause libraries, and component files, as well as the dependencies between all these files. You can also copy components to multiple files at once to ensure consistency across templates and component files. Likewise, you can rename components in multiple files. When you rename a component,

the change is also reflected in the corresponding template files. Additionally, you can single out unused components so you can delete them from the component files, which can reduce the number of components with which you must work in a given file. Finally, you can modify the properties of several component files at once.

At a Glance: The Template Manager

🗄 HotDocs Template Manager				
┣ Ţ▼ ± ☎ ┣ ┣ #	🛓 🛉 🖞 Rename 🔏 Delete 🛛 🔑 For	cus All Components	- 😨 🔽	
Template Name / Component File	Component Name	Component Type	Status 🔺	
🕀 🖹 Employment Agreement	A Address Line 1	Text Variable	OK _	
🕀 🖹 Employee Personal Data	A Address Line 2	Text Variable	ОК	
🖶 🖹 Editor/Author List	31 Agreement Date	Date Variable	ОК	
🖶 📴 Life Insurance Application	Agreement Information	Dialog	ок	
🗄 🖹 Collection Letter	▲ AL/AK/AZ/AR/CA/CO/CT/DE	Merge Text	ок	
1	2 Amount of Coverage	Number Variable	ОК	
B	2 Annual Salary	Number Variable	ОК	
	🗛 Author Full Name 🛛 🙂	Text Variable	ОК	
	Author Information	Dialog	ОК	
	Hand Authorization	True/False Variable	ОК	
1	Beneficiary Designation	Dialog	ОК	
	=🗶 Beneficiary Filter 1	Computation Varia	ОК	
	=X Beneficiary Filter 2	Computation Varia	ОК	
	A Beneficiary Name	Text Variable	ОК	
	A Beneficiary Relationship	Text Variable	ОК	
	A Beneficiary Telephone Number	Text Variable	ОК	
	🚍 Book Information	Dialog	ОК	
	A Book Title	Text Variable	ОК	
	▲ City	Text Variable	<u>OK</u>	
Delete selected components from component files that conta 5 templates 125 unique components				

After opening the **Template Manager** from the HotDocs Library toolbar, you can click on the **Each Convert Files** button on the template manager's toolbar.

The Template Manager is split into three main sections. At the top of the window is the **Toolbar** A where you can perform actions by choosing from the following buttons:

- **Open:** Opens a new Template Manager window that shows only the components for the files you've selected in the file list.
- **Expand All:** Expands the list to show all top-level entries in the list with their related files.
- **Collapse All:** Collapses the list to show only the top-level entries in the list.
- **Component Manager:** Opens the Component Manager.

- **Component File Properties:** Opens the **Component File Properties** dialog box so you can modify properties across multiple files at once.
- Convert: Converts the selected files to either the latest version of HotDocs or to DOCX format.
- **Copy:** Copies the components so you can paste them in other files.
- **Paste:** Pastes copied components into other template's component files.
- **Delete:** Deletes the selected component from the files you select.
- **PFocus:** Shows only the components for the selected template or group of templates.
- Component Type drop-down list: Shows only the specific type of component you select.
- **(2) Refresh:** Regenerates the component database and updates the contents of the Template Manager window if you've made any changes.
- **Options:** Changes specific settings for what information Template Manager displays in the components list.
- **Whelp:** Opens the relevant page of the HotDocs Help File.

To the left of the page there is the Template Name/Component File list **B** where you can see a list of all the template files in the library. You can view the component files by expanding.

To the right of the page is the Component List \subseteq where you can see all the components in the library. You can sort the list using the Component Type drop-down list to filter by type or by turning focus on to only show the components that relate to the selected template.

For a more in-depth look at the parts of the Template Manager see Understand Parts of Template Manager Window and Template Manager Toolbar.

To learn more about using the Template Manager follow the links below:

- Introduction: Use Template Manager
- Open and Close Template Manager
- Export Contents of a Template Manager Window to a Spreadsheet
- Convert Multiple Templates to Work with HotDocs 11
- Convert Templates and Clauses to Microsoft DOCX
- Rename Templates Using Template Manager
- Copy and Paste Components Across Multiple Component Files
- Rename Components Across Multiple Component Files
- Delete Components from Multiple Component Files
- Modify Component File Properties Across Multiple Files

Open and Close Template Manager

You can open Template Manager from the HotDocs template library. As Template Manager opens, it scans through the entire template library and creates a database file that lists all of the files in the library as well as all the components used in those files. (This happens even if you select only a single file to explore.) HotDocs uses this database to track changes you make to templates and components so that your changes won't adversely affect other files in the library.

For example, perhaps you want to rename a single template file. When you select it at the template library and launch Template Manager, HotDocs will create a database of all the files in the library, even though you have selected only one. This allows HotDocs to make sure that when you rename the template, any other place where the file is referred to in the library (either in the library itself or in another template), those references are updated with the new file name.

This database file is saved to the same file location as the template library. Do not open this file directly with the intent to modify its contents. Changes you make may have potentially serious consequences. For example, the change may cause HotDocs to make incorrect assumptions about what files and components are being used. This may make you think it's safe to delete something you shouldn't, thus "breaking" templates. Additionally, changes in the database may cause HotDocs to crash if it receives data it is not expecting.

To start Template Manager

- At the HotDocs template library, select the library items you want to work with and click the Template Manager button. A dialog appears telling you that HotDocs is building a component database. (Depending on the number of files in the library, this process may take several minutes.)
 - If it encounters any errors when compiling the database, HotDocs displays an option to view the **Error Report**, which appears in a temporary text file. The report lists the folder path and file names of the files that contained the errors.
 - If it doesn't encounter any errors when compiling the database, it launches Template Manager.
- 2. Once Template Manager appears, you can perform any number of tasks, such as:
 - View the contents of a Template Manager window
 - Work with Template Manager
 - Convert multiple templates to work with HotDocs 11
 - Convert templates and clauses to Microsoft DOCX
 - Rename templates
 - Copy and paste components
 - Rename components
 - Delete components
To close Template Manager, click the **X** in the upper-right corner of the Template Manager window.

Understand Parts of Template Manager Window

When you first start Template Manager, it creates a database of all of the templates and components found in the current library. However, only the templates and components for those files you selected in the library appear in the Template Manager window, which is divided into two panes—the file list and the component list.

Together, the file list and component list show relationships between files and components. For example, when you select a file, Template Manager displays a red marker next to the components used in the selected file. Likewise, when you select a component, HotDocs displays a red marker next to the file or files that use that component either directly or indirectly (for example, you may have an inserted template that uses that component).

As you examine templates and components in Template Manager, you can customize your view of the window. See Template Manager Toolbar for details.

Viewing the File List

The file list shows all of the files you selected at the template library. You can expand each file to see all of the files associated with it. (You can expand the file list to show corresponding component files and inserted templates by clicking the $\mathbf{\overline{F}}$ **Expand** button. (Click the $\mathbf{\overline{f}}$ **Collapse** button to once again minimize the list.) For example, if you expand a template file, you will see the template's component file, as well as any other files that may be referenced in the template. This can include a clause library (and all of its clauses) and any templates referenced in INSERT or ASSEMBLE instructions (which are denoted by the $\mathbf{\widehat{F}}$ icon.)

🛱 HotDocs Template Manager				-
📴 E 👎 🛔 😤 🖻 🗈 🙈 🍟 Rena	ame 🔏 Delete 🔑 Focus 📶 Co	omponents 🗸	2 🔽 🕜	
Template Name / Component File	Component Name /	Component Type	Status	*
🕀 🖹 Employment Agreement	Address Line 1	Text Variable	ОК	
📕 demo employment agreement.cmp	Address Line 2	Text Variable	ок	Ξ
🕀 🖹 Employee Personal Data	31 Agreement Date	Date Variable	ок	
📠 🏜 demo personal data.cmp	Agreement Information	Dialog	ок	
🖶 🖹 Editor/Author List	🔙 AL/AK/AZ/AR/CA/CO/CT/DE	Merge Text	ОК	
🔤 矗 demo editor list.cmp	2 Amount of Coverage	Number Variable	ОК	
🖨 📴 Life Insurance Application	2 Annual Salary	Number Variable	ОК	
🚽 🚠 demo life insurance application.cmp	🗛 Author Full Name	Text Variable	ОК	
🖻 🖹 Collection Letter	🚍 Author Information	Dialog	ОК	
🔒 demo collection letter.cmp	🖁 Authorization	True/False Variable	ОК	
	🚍 Beneficiary Designation	Dialog	ОК	
	=🗶 Beneficiary Filter 1	Computation Varia	ОК	
	=🗶 Beneficiary Filter 2	Computation Varia	ОК	
	🗛 Beneficiary Name	Text Variable	ОК	
	🗛 Beneficiary Relationship	Text Variable	ОК	
	A Beneficiary Telephone Number	Text Variable	ОК	
	🚍 Book Information	Dialog	ОК	
	🗛 Book Title	Text Variable	ОК	
	🗛 City	Text Variable	ОК	
	🛄 Client Data	Database	ОК	
	A Client Name	Text Variable	ОК	Ŧ
c:\\templates\demo employment agreement.rtf	1 of 5 templates selected 1	24 unique components	s	

Where there are problems with a specific file, such as the file is referenced either in the library or a template, but doesn't actually exist on disk, Template Manager overlays a warning on the file icon. (Click here for an explanation of the different template status icons used.)

You can choose to view either file titles or file names. Right-click anywhere in the file list and choose your preference. Additionally, you can copy information about the files you're exploring to another program, such as Notepad or the word processor by choosing **Copy as Text** from the shortcut menu.

While viewing the file list, you can:

- Convert multiple earlier versions of HotDocs templates to HotDocs 11 format
- Convert multiple templates to Microsoft DOCX
- Rename templates using Template Manager
- Modify component file properties across multiple files

Since the release of HotDocs 11.1, HotDocs no longer supports converting component files to pre-2009/10 formats.

Viewing the Component List

The component list shows all the components associated with the files in the file list. This can include components like variables and dialogs, but it can also include supplemental components, such as example formats, merge text, and patterns. Template Manager shows information about the component, such as whether it's used in a template, or whether it's referred to in a template but doesn't exist in the component file. (Click here for an explanation of the different component status icons used.)

You can control what types of components and other information is displayed in the component list. See Set Template Manager Options for details. Additionally, in the component list, you may see multiple listings of the same component. This means that while each component shares the same component name, there is at least one property of the component that is different, such as a prompt. You can right-click the component and select **View** from the shortcut menu to view the properties of the component; however, to edit the component's properties, you must use Component Manager, which you can access by right-clicking the corresponding file and selecting **Component Manager** from the shortcut menu.

While viewing the component list, you can:

- Copy and paste components across multiple component files.
- Rename components across multiple component files.
- Delete components from multiple component files.
- Modify component file properties across multiple files.

To access the Template Manager toolbar using the keyboard, press F10.

Template Manager Toolbar

The following buttons and commands appear on the Template Manager toolbar:

Button or Command	Description
Open Window	Opens a new Template Manager window that shows only the components for the files you've selected in the file list. This makes it easier to copy components between files and helps control the number of files and components you are viewing. You can have as many windows open as necessary.
Expand All Ţ	Expands the list to show all top-level entries in the list with their related files.

Collapse All	Collapses the list to show only the top-level entries in the list.
7	
Component Manager	Opens Component Manager.
A	
Component File Properties	Opens the Component File Properties dialog box so you can modify properties across multiple files at once.
*	
Convert Files	Converts the selected files to either the latest version of HotDocs or to DOCX format.
Copy Components	Copies the components so you can paste them in other files.
E)	
Paste Components	Pastes copied components into other template's component files.
2	
Rename	Renames the selected file or component.
abi	
Delete	Deletes the selected component from the files you select.
_	
Pocus D	Focus again to show <i>all</i> components.)

Components drop-down list	Shows only the specific type of component you select.
Refresh ②	Regenerates the component database and updates the contents of the Template Manager window if you've made any changes.
Options	Changes specific settings for what information Template Manager displays in the components list.
😧 Help:	Opens the relevant page of the HotDocs Help File.

Export Contents of a Template Manager Window to a Spreadsheet

When working with Template Manager, you may want to copy either a list of variables or a list of files to a spreadsheet so you can work with that data directly. You can do this by first opening the spreadsheet application (such as Microsoft Excel) and dragging the data directly from Template Manager into the spreadsheet application.

To export lists of items from Template Manager to a spreadsheet

- 1. Start Template Manager. (See Open and Close Template Manager.)
- 2. Start the spreadsheet application. (See that program's help file for information on doing this.)
- 3. Complete either of the following steps:
 - To copy a list of **components** to the spreadsheet, select the components in the component list and drag them to the rows in the spreadsheet you want to fill. When you release the mouse, the component names appear in one column, while the corresponding component types appears in a second column.
 - To copy a list of **templates** to the spreadsheet, select the templates in the file list and drag them to the rows in the spreadsheet you want to fill. When you release the mouse, the template titles (or file names, if you are showing them) appear in one column, while the corresponding file path appears in a second column.

At a Glance: The Convert Files dialog box

Convert Files	? 💌			
Choose the operation to be performed on the selected file	s:			
Or Convert component files to work with HotDocs 11				
Convert WordPerfect templates and clauses	To format:			
Convert MS Word DOT templates and clauses	MS Word DOCX			
Convert MS Word <u>R</u> TF templates and clauses	O MS Word RTF			
The following files will be converted:				
c:\\documents\hotdocs\templates\demo employment a	c:\\documents\hotdocs\templates\demo employment agreement.cmp			
C References to the above files will be updated if necessary in any files that refer to them.				
After component files are converted, they cannot be used with any version of HotDocs prior to HotDocs 11.				
Conversions cannot be undone, so you should always back up your templates and component files before converting them.				
Are you sure you want to convert these files as indicated above?				
(OK Cancel			

After selecting **Keep existing format**, when prompted by the Component Manager, HotDocs will then only allow format changes to be made at the Component manager and will not enact changes made to the component file in Template Manager.

After opening the **Template Manager** from the HotDocs Library and selecting a template, you can click on the **Convert files** button. From the Convert Files dialog box you can convert one or more templates or component files.

At the top of the dialog are four check boxes \bigcirc you can use these to decide what sort of conversion you would like to run. You can tick the first check box to convert older component files to work with HotDocs 11.

You can use the other three check boxes to select which format your template currently is, and then the two check boxes **B** to the right to select the format you would like to convert it to. You can select

WordPerfect, MS Word DOT, or MS Word RTF templates to convert and from MS Word DOCX and MS Word RTF for the end format.

In the field below C you will see further details about the files to be converted.

If the file you are converting contains INSERTed files, Template Manager will convert both the parent template and any INSERTed templates or clauses.

Template Manager will not convert files used in ASSEMBLE instructions. However, you can convert these files at the main list and any ASSEMBLE instructions that refer to them in other files will be updated.

After a file is converted, Template Manager will search through the contents of the other files in the library to see if any of those files contain INSERT or ASSEMBLE instructions that refer to the template or clauses it just converted. If so, Template Manager updates the file name extension in those instructions so that the instruction correctly refers to the new file.

To learn more about converting files using the template manager follow the links below:

- Convert Multiple Templates to Work with HotDocs 11
- Convert Templates and Clauses to Microsoft DOCX

Convert Multiple Templates to Work with HotDocs 11

Any time you make major changes to templates, component files, or other HotDocs files, it is a good idea to back the files up before making the change. This is especially true when converting component files to work with HotDocs 11.

HotDocs 11 uses a file format for component files that is different from pre-2009 versions of HotDocs. This means that all component files created in these earlier versions must be converted to HotDocs 11 format in order for them to work. Failure to convert the files may result in errors when you attempt to assemble documents using them.

You can either convert component files individually (see Convert a Single Template to a New File Format), or you can convert several at once using Template Manager. When you convert using Template Manager, Template Manager can convert all component files referred to in the template or templates you are converting. For example, if *Template A* contains an INSERT instruction, when you convert *Template A*, any templates (and component files) referred to in the instruction will also be converted.

To convert templates to HotDocs 11 format

1. Back up the templates you want to convert by copying them to a different location.

- 2. Open the library containing the templates you want to convert. (See Create a Library and Add Templates and Other Files to a Library.)
- 3. Click the top folder in the library and launch Template Manager. (See Open and Close Template Manager.) The Template Manager window appears.
- 4. In the file list, select the templates you want convert. (To select all, press Ctrl+A.)
- 5. Click the Convert Files button. The Convert Files dialog box appears.
- 6. The **Convert component files to work with HotDocs 11** option is always selected by default. (This is correct.)
 - Optionally, select one of the options to convert the Word or WordPerfect templates to DOCX. (See Convert Templates and Clauses to Microsoft DOCX for details about this kind of conversion.)
- 7. Click **OK.** HotDocs converts all the selected files (and any dependent files referenced by the templates) to HotDocs 11 format.

After selecting **Keep existing format**, when prompted by the Component Manager, HotDocs will then only allow format changes to be made at the Component manager and will not enact changes made to the component file in Template Manager.

Convert Templates and Clauses to Microsoft DOCX

Any time you make major changes to templates, component files, or other HotDocs files, it is a good idea to back the files up before making the change. This is especially true when converting component files to work with HotDocs 11.

If you change your word processor brand from WordPerfect to Microsoft Word, you can convert your templates and clause libraries to Microsoft Word DOCX using Template Manager.

If you are already using Microsoft Word but your templates or clauses are in RTF format, you should consider converting them to DOCX. This is because HotDocs can now assembly DOCX templates without file conversion, creating less opportunities for errors to appear in your templates. By automating the templates (and clauses) in DOCX, you can reduce the number of steps HotDocs must follow to assemble a document. This improves assembly speed.

Template Manager allows you to convert multiple files to DOCX at once. Doing so ensures that any files dependent on the file you are converting are likewise converted and references to it are updated. Please note the following:

- When you select a template or clause library in the file list to convert, Template Manager will convert that file to the new format.
- If the file you are converting contains INSERTed files, Template Manager will convert both the parent template and any INSERTed templates or clauses.

• Template Manager will *not* convert files used in ASSEMBLE instructions. However, you can convert these files at the main list, and any ASSEMBLE instructions that refer to it in other files will be updated.

After a file is converted, Template Manager will search through the contents of the other files in the library to see if any of those files contain INSERT or ASSEMBLE instructions that refer to the template or clauses it just converted. If so, Template Manager updates the file name extension in those instructions so that the instruction correctly refers to the new file.

When converting templates to a new format, it's a good idea to convert all the files in the library so that you don't create any file name extension inconsistencies. For example, *TemplateA.wpt* and *TemplateB.wpt* both INSERT *Subtemplate1.wpt*. If you were to select only *TemplateA* for conversion, Template Manager would convert it and *Subtemplate1* to RTF. Template Manager would then update the reference to *Subtemplate1* in *TemplateB*, but *TemplateB* would still be in WordPerfect format because you haven't converted it yet. If you tried to assemble *Template B*, you would get errors. (Remember, when inserting one template into another, the file types must match.)

To convert WordPerfect / Word templates or clauses to DOCX

- 1. Back up the templates you want to convert by copying them to a different location.
- 2. Open the library containing the templates you want to convert. (See Create a New Library By Exporting Part of an Existing Library and Add Templates and Other Files to a Library.)
- 3. Click the top folder in the library and launch Template Manager. (See Open and Close Template Manager.) The Template Manager window appears.
- In the file list, select the templates you want to convert and click the Convert button. The Convert Files dialog box appears, with the Convert component files to work with HotDocs 11 option selected by default. (See Convert Multiple Templates to Work with HotDocs 11.)
- 5. Select Convert WordPerfect templates and clauses... to convert your files to WordPerfect.
- 6. Select **Convert MS Word RFT templates and clauses...** to convert your RTF files. HotDocs lists the files that will be converted in the field below these options.
- 7. Click the **Convert to: MS DOCX** check box to the right to convert the selected files to DOCX.
- 8. Click **OK**. Template Manager confirms that you want to convert the files.
- 9. Click Yes to continue.

Template Manager converts the templates (and clauses) to the new format. As it does this, it opens each template file in Word and makes the necessary updates. Once it finishes converting the templates, it refreshes the Template Manager database. When conversion is complete, you are returned to the Template Manager window.

Something to be aware of with DOCX templates is that, Word documents that were originally converted from HTML files will have hidden HTML data stored in the file, once converted to a DOCX template some of that data might not have been stored correctly in the new DOCX

template and you may encounter problems if you ever need to convert that template back to HTML.

At a Glance: The Rename Files dialog box

Rename Files	? 💌
Old file name: demo employment agreement	
New file name: demo employment agreement	
The following related files will be renamed as indicated:	
c:\\documents\hotdocs\templates\demo employment agreement.cmp c:\\documents\hotdocs\templates\demo employment agreement.rtf B	
References to the above files will be updated in these files:	
G	
	Cancel

After opening the **Template Manager** from the HotDocs Library, you can select the template you would like to rename then click on the **PRENAME** button on the toolbar.

In the first text field \boxed{A} enter what you would like to rename the file as. In the field below \boxed{B} you will see a list of related files that will be edited and their new names and in that last field \boxed{C} you will see a list of files containing references to the files in the middle field \boxed{B} . These references will also be updated.

To learn more about renaming files using the template manager follow the link below:

• Rename Templates Using Template Manager

Rename Templates Using Template Manager

Sometimes as you work with complex template sets, you may want to rename templates. However, renaming templates in HotDocs isn't as easy as simply renaming the template and component file using Windows Explorer. References to the template must be updated in the library. References to the file in other templates must also be updated. Additionally, any other dependent files (such as clause libraries and clause archive files) must also be renamed.

Template Manager allows you to rename a template and have all the corresponding files and references to the template in other templates updated as well. For example, if you rename a clause library, both its component file and clause archive file will be renamed. If the clause library is inserted in any templates, the INSERT instruction will be updated with the new file name.

To rename templates using Template Manager

1. Start Template Manager. (See Open and Close Template Manager.) Template Manager opens.

You should select all of the templates in the library before starting Template Manager. This makes it easier to update file names throughout the template set.

- 2. Optionally, right-click anywhere in the file list and choose **Show File Names** from the shortcut menu. This lists templates by file name rather than title, making it easier to identify which files you want to rename.
- 3. Select the file you want to rename from the file list and click ***Rename**. The **Rename Files** dialog box appears.
- 4. Type the new file name in the **New file name** field. Do not include a file name extension.
- 5. As you enter the new file name, Template Manager updates the file list below so you can see which files will be updated.
- 6. Click **Rename**. Template Manager renames all the appropriate files and updates references in the template library. It also searches through other files in the library and updates any INSERT or ASSEMBLE instructions with the new file name information.

If you are renaming a template whose component file is pointed to a shared component file, Template Manager will rename the pointed component file and repoint it to the shared file.

At a Glance: The Copy Components dialog box

iny

After opening the **Template Manager** from the HotDocs Library, select multiple components from the **Component** list then you can click on the **Copy Component** button on the toolbar. This brings up the **Copy Components** dialog box where you can select the correct component file to copy the components (and any components they reference) from.

To learn more about copying components using the template manager follow the links below:

• Copy and Paste Components Across Multiple Component Files

At a Glance: The Paste Components dialog box

- E: \users \holli	icooper.hotdocs\documents\hotdocs\demo lifi	e insurance application.cm
c: \users \holli	icooper.hotdocs\documents\hotdocs\template	es\demo collection letter.ci
C: \users \holli	icooper.hotdocs\documents\hotdocs\template	es\demo editor list.cmp
c: Jusers (holli	icooper.notaocs\aocuments\hotdocs\template	es (demo employment agre
Eller fosers friom	icooper motions (accuments y lottices (template	es querro personal data, cin

After opening the **Template Manager** from the HotDocs Library, and once you have copied files from the component list, you can click on the **Paste Component** button on the toolbar to paste these

components into a different template. This will open the **Paste Components** dialog box where you will be asked to confirm which component file in your library you would like to paste into.

Select the file you would like to paste into by ticking the check box next to its file path in the field.

Below this is a check box you can un-tick if you would prefer HotDocs not warn you when you are overwriting existing components.

To learn more about copying components using the template manager follow the links below:

• Copy and Paste Components Across Multiple Component Files

Copy and Paste Components Across Multiple Component Files

Sometimes you may create a variable or other component that you would like to copy into all the component files in your template library. Rather than open each component file manually and copy the variable to it (see Copy Components From One File to Another), you can use Template Manager to copy a component into several component files at once.

You can copy multiple components at a time. If the component you select is a clause that uses variables, a variable reference (or variable used in another variable's prompt), a dialog, or Computation variable that refers to or requires other variables to function, those variables will also be copied to the new component file.

To copy and paste components between component files

- 1. At Template Manager, select a component or components from the component list and click the **Copy Components** button. The components are saved to the Clipboard.
- If the component is used in two or more component files, Template Manager displays the Copy Components dialog box that lets you choose which component file you want to copy from. Select the correct file and click Copy. The component is saved to the Clipboard.
- 3. From the file list, select the file or files you want to copy the component into.
- 4. Click the **Paste Components** button. The **Paste Components** dialog box appears. The files you have already selected appear in this list with a check mark next to them. The list also includes other component files you are currently working with.
- 5. Select additional component files from the list, clear the selected files, or accept the current selection.
- 6. Optionally, select **Warn when overwriting components that already exist** if you want Template Manager to warn you if the component you are copying already exists in the file.
- 7. Click **Paste**. Template Manager pastes the component into the file or files.

You can also drag components from the component list to selected files in the file list to copy and paste them.

At a Glance: The Rename Components dialog box

Rename Components		? 💌
Old component name:	Agreement Date	
New component name:	Agreement Date	A
The component will be r	enamed and reference	ces will be updated in the following files:
C:\\documents\hotdocs\templates\demo employment agreement.cmp		
C: \ \documents \nd	luocs (lempialles (dem	no employment agreement.rtr
ß		
Clear All	<u>S</u> elect All	Rename Cancel

After opening the **Template Manager** from the HotDocs Library, select a component from the **Component List**, then you can click on the **Preserve** button on the toolbar to rename this component.

In the first text field \underline{A} enter the new name for your chosen component. In the field below \underline{B} you will see a list of the files containing that component and you have the option by unticking the check boxes to deselect files from this list.

At the bottom of the dialog there are two buttons where you can choose to **Clear All** check boxes or **Select All** check boxes.

To learn more about renaming components using the template manager follow the links below:

Rename Components Across Multiple Component Files

Rename Components Across Multiple Component Files

Sometimes as you create complex template sets, you may wish to simultaneously change the name of a component throughout the entire set. You can use Template Manager to select a component, assign a new name, and make the change in multiple component files.

When you use Template Manager to rename components, components are renamed in both the component file and the template file. Additionally, when renaming variables, any references to the variable in prompts or dialog element text will also be renamed.

To rename components using Template Manager

- 1. At Template Manager, select a component from the component list and click ***Rename**. The **Rename Components** dialog box appears.
- 2. Type the new component name in the **New component name** field.
- 3. From the **Following files** field, select the files you want to make the change in. (This field lists all of the files that contain the component or a reference to the component.)

If a component is used in a template and its associated component file, you must select both files in the list before you rename the component. If you select one and not the other, you will create an unresolved reference and you will not be able to assemble a document from the template. (Template Manager selects both files by default.)

4. Click **Rename**. Template Manager renames all selected instances of the component.

Delete Components from Multiple Component Files

As you explore component files, you may discover you have several components that are unused. You can use Template Manager to remove the unused components. If your component file contains a large number of components, removing unused components can make the list much more manageable.

You can identify an unused component by its status, which is shown in the component list.

To remove unused components from one or more component files

- 1. At Template Manager, select the component you want to remove from the component list.
- 2. Click **Delete**. Template Manager confirms that you want to delete the component.
- 3. Click Yes. Template Manager removes the component from the associated component files.

Modify Component File Properties Across Multiple Files

There may be certain component file properties that you want used across multiple templates in your set. You can use Template Manager to specify these properties at once for all of these files.

For example, perhaps you want all the templates in your library to use the same product title. Rather than change this property in each individual component file, you can specify a product title using Template Manager and have it applied to all the selected component files at once.

To modify component file properties in multiple component files

- 1. At Template Manager, select the files you want to modify from the file list and click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 2. Enter or select which properties you want to change. (See Change Component File Properties for specific details of each option.)
- 3. Click **OK**. HotDocs displays a list of component files that will be changed.
- 4. Click **OK**. The changes are made in the selected component files.

Using Component Manager

Using HotDocs Scripting

Scripting Overview

Introduction: Understand the HotDocs Scripting Language

Learning the Language

You can instruct HotDocs to perform certain tasks using the HotDocs scripting language. For example:

- Insert one template into another
- Hide variables in dialogs
- Perform some action based on an answer the user provides
- Add up several dollar amounts
- Find the number of years between two given dates
- Search a user's answer for a certain block of text

The HotDocs scripting language consists of instructions, expressions, operators, and values—such as text, numbers, dates, or answers users enter. To help you learn this language, HotDocs provides you with instruction and expression *models*.

Specifically, an *instruction model* tells HotDocs to perform some sort of function, while an *expression model* retrieves a special value. Most models also include *placeholders*, which you must replace with a value. Possible values include:

- text strings
- number amounts
- other models
- HotDocs variables

A model will not work until all its placeholders are replaced.

In addition to using instructions and expressions, you can use operators to control how the script is processed. Most operators are common mathematical signs, but there are also Boolean operators such as *AND* and *OR*. The operator *AND* means the statement to the left and the statement to the right must both be true. The operator *OR* means either the statement to the left or the statement to the right must be true. The operator () means to perform the operation between the parentheses first.

If your script requires input from another variable, you can create a parameter, an internal variable that accepts input from the variable field. This allows you to create a script that acts like a function, allowing that single script to work against a number of different values. You can also set local variables for your scripts which only exist within that script and do not need to be maintained as regular variable components in the component file. These are useful for things like counters for WHILE loops and flags for IF statements.

Rules for Writing Scripts

Instruction and expression models are designed to help you accurately write scripts. However, there are still certain rules you must follow, particularly when it comes to replacing placeholders with values. Failure to follow these rules may result in syntax errors when HotDocs attempts to process the script during assembly.

When using literal values:

- Decimal numbers less than one must have a zero before the decimal point (0.125 not .125).
- Numbers cannot contain commas (1250 not 1,250).
- Dates must be in the form 3 JUN 1990.
- Text strings must be inside quotation marks.

To help you format these numbers correctly, use the **Enter a Date**, **Enter a Number**, **Enter some Text**, and **Enter True or False** expression models.

When manually typing the script:

- Component names must be entered exactly as they appear in the **Components** list, including capitalization.
- Components and operators must be separated by spaces.
- Operators must be entered exactly as they appear in the **Operators** list, including capitalization.
- Variables used inside a literal text string (or between quotation marks) in a computation must be inside chevrons (« »). If the variable name is not inside chevrons, the variable name—not the variable's value—will be merged as the answer. (For example, "«Employee Name» must wait «Number of Months» months before applying.")

If you want to include a chevron in a HotDocs text template either as a part of your document text or a literal text string, you need to double the opening chevron (but *not* the closing chevron), so that HotDocs does not interpret the chevrons (and any text between them) as a field. The output is a single chevron, as follows: input: ««some text», output: «some text».

- Keywords, such as TRUE, FALSE, AND, END IF, and so forth, must be in all capital letters.
- Only **TRUE** or **FALSE** can be used for a true/false value.

- If you want the computation to return a combination of text, variables, and values, you must join (or concatenate) the data with the Add (+) operator. Otherwise, HotDocs returns only the last piece of text, variable, or value.
- When using the RESULT expression, you must first set the computation to a value. If the computation will return text, set it to a specific value by typing text inside quotation marks at the start of the script. (To set a text computation to an empty value, type nothing between the quotation marks—not even a space.) If the computation will return a number, set it to a value by typing a number at the very beginning of the script. (To set a number computation to nothing, type 0 (zero).)

Writing the Script

There are three places in HotDocs where you can write scripts: The **Computation Editor**, the **Script** tab of the **Dialog Editor**, and the **Expression** box of the **IF Field** dialog box. Each of these places provides you with the tools you need to write the script. Which tools you use, however, depends on personal preference, as well as your skill level.

If you are learning the scripting language, we suggest that you select and drag instructions and expressions from the models lists to the **Script** box. You can also drag operators and components from their respective lists into the **Script** box. If there are placeholders, replace them by dragging components or other models onto them. To correctly format literal values (such as text, numbers, dates, and true/false values) use the **Enter...** expression models.

If you prefer to use the keyboard rather than the mouse, press the **Tab** key to move from one field in the dialog box to the next. Use the arrow keys to select specific components, operators, and models, and use the **Insert** key to bring the selected element into the **Script** box. To replace placeholders in the script, insert your cursor in the placeholder text and then locate the value you want to replace it with in the lists below and press the **Insert** key.

Once you become familiar with the instructions and expressions, you can type your scripts directly in the **Script** or **Expression** box. To ensure you use the correct instruction and expression keywords and component names, you can access lists of these things using keyboard shortcuts and other options available in the script editor. When you do this, HotDocs displays an auto-complete list from which you can choose the instruction or expression you are typing.

Use Operators when Scripting

An operator is a symbol or word that causes an operation such as addition or a comparison to be performed in a computation or expression. Operators are available at the **Operators** list wherever you can create a script or expression. Most operators can be used when working with both number and text values.

There are three types of operators:

- **Comparison operators:** These compare two values of the same type (text, number, date, multiple choice, or true/false). They return values of true or false depending on whether the comparison is true or not.
- **Arithmetic operators:** These calculate new values. Operands used in the script must be the same type.
- **Logical operators:** These return a true/false value based on a logical comparison of their operands, both of which must be true or false values.

The following tables explain how each operator works:

Comparison Operator	Description
=	The two items in the comparison are of equal value. For example:
	Birth Date = 17 Dec 1989
	Employee Name = "Louisa Gehrig"
!=	The two items in the comparison are not of equal value. For example:
	IF Exhibit A != TRUE
	IF Plaintiff Gender != "Male"
<	The first item in the comparison has a lesser value than the second item. For example:
	Account Balance < 9000
	COUNTER < 10
>	The first item in the comparison has a greater value than the second item. For example:
	Dependent Age > 18
<=	The first item in the comparison is less than or equal to the second item. For example:
	Client Age <= 65
	COUNTER <= 2

>=	The first item in the comparison is greater than or equal to the second item. For example:
	Taxed Income >= 75000
CONTAINS	The value of the first item is found in the value of the second item. For example:
	"massachusetts virginia kentucky pennsylvania" CONTAINS State Name
Arithmetic Operator	Description
+	Add the different components of the script together. For example:
	Value 1 + Value 2
	Street Address + ", " + City + ", " + State
-	Subtract the different components of the script from each other. For example:
	Monthly Income - Amount of Owed Child Support
*	Multiply the different components of the script. For example:
	Purchase Price * 0.625
/	Divide the different components of the script. For example:
	Yearly Salary / 12
-	The unary minus operator results in the numeric negation of the operand (which must be a Number value). For example:
	SET Loss Amount TO - Aggregate Amount
%	The unary percent operator results in the operand (which must be a Number value) divided by 100. It is a postfix operator. For example:
	Purchase Price + (Purchase Price * 6.25%)
Logical I Operator	Description

AND	The statement to the left and the statement to the right must both be true. For example:
	IF Client is Married AND Client has Children
OR	The statement to the left or the statement to the right must be true. For example:
	IF Single OR Widowed
NOT	The NOT operator results in the logical negation of the operand (which must be a True/False value). For example:
	IF NOT Client is Married

HotDocs also supports two other unary operators—unary plus (+) and the dollar sign (\$). While both produce numeric results in a script, the results are exactly the same as the operands. Therefore, they should not be used in a script.

The Add (+) operator can also be used to string together (concatenate) two text values.

The final operator, the parentheses (), instructs HotDocs to perform the operation inside the parentheses first.

HotDocs operators are processed in the following order of precedence, from highest to lowest. Operators listed on the same line have the same precedence.



So in other words, when HotDocs is evaluating an expression, first parenthesis are used to determine order of operations, then unary operators, then multiplication and division, then plus and minus, then comparisons, then logical AND, and last logical OR.

Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts

Inserting line breaks and paragraph ends are only relevant if you are merging text into an assembled Word document. In WordPerfect, each of the commands listed below inserts a hard return. (See the WordPerfect helps for an explanation.)

When including literal text strings in computation scripts, you can have HotDocs merge line breaks, paragraph ends, and tabs in the answer. To do this, at the **Script** field, either by manually entering the characters or using a dot code, use the following information:

To insert	Do one of the following
A line break	Press Enter or Shift+Enter . When you do this, HotDocs creates a new line of text in the same paragraph.
	When you use a line break to span a literal text string across two lines in the script editor, the color coding assigned to the text string changes to the default color. This doesn't affect how the computation will be processed, but it may make it more difficult to visually recognize the different portions of your script. To fix the color coding, click the Auto Format button.
	Insert a Line Break dot code («.lb») at the place you want HotDocs to start a new line. (See Insert Characters in Text Strings.)
A paragraph end	Press Ctrl+Enter . When you do this, HotDocs inserts a paragraph mark (\P) and starts a new paragraph of text.
	Insert a Paragraph Mark dot code («.pm») at the place you want HotDocs to start a new paragraph. (See Insert Characters in Text Strings.)
A tab character	Press Ctrl+T .
	If you want the Tab key to insert a tab character (instead of you pressing Ctrl+T), click the Options button and select Tab key inserts a tab in scripts . Now, whenever you press Tab or Shift+Tab , HotDocs will insert a tab in the script instead of taking you to another field in the dialog box.
	Insert a Tab Character dot code («.tc») at the place you want the text to be tabbed. (See Insert Characters in Text Strings.)

When creating multi-line Text variables, you can force HotDocs to merge a paragraph mark (rather than a line break) when the user presses **Enter**. For details, see Customize a Text Variable.

Use the Script Editor

When writing a computation script, there are several tools you can use to make the process easier.

To use the script editor

- 1. Edit the script or expression.
- 2. At the **Script** or **Expression** field, complete any of the following tasks:

То	Do This
Have HotDocs automatically complete keywords, component names, and Multiple Choice options as you type	 Place your cursor in the Script field and press Ctrl+Spacebar. HotDocs displays a list of instruction and expression keywords, constant values (such as month abbreviations), and components. Type a portion of the keyword, component name, or Multiple Choice option for which you are searching. As you type, HotDocs filters the list to show only those keywords that contain the text you have typed. Once selected, press Enter to merge it into your script. To keep seldom-used keywords out of the auto-complete list, click the Options button and clear Include seldom-used keywords in auto-complete list.
Access just a list of components	Press the F5 key, select the component, and press Enter.
Access just a list of keywords	Press Shift+F5, select the keyword, and press Enter.
Display a syntactical hint of how an instruction or expression should be used	Place your cursor inside the keyword and press the F7 key. HotDocs displays a small ToolTip that shows the entire model as well as the type of value it produces, if it's an expression.
Indent matching pairs of IF and REPEAT instructions based on the level of their insertion	Click the Auto Format button.
Undo (or cancel) an action you just performed	Click the Oundo button. HotDocs reverses the change you made.
Redo (or re-implement) an action you just performed	Click the 🔉 Redo button. HotDocs reapplies the change you made.

Cut or copy and paste a selected portion of the script	Select the portion of the script you want to cut or copy, and then click the Cut button or the Copy button. The script is copied to the Clipboard. To paste the script in a new location, insert your cursor at that location and click the Paste button.
Find a specific string of text in the script	Click the A Find button and enter your search text in the Find what field. To find the next instance of the text, click the A Find Next button.
	To find only those instances of text that are complete words, select Find whole words only . To find only those instances that have the same capitalization as the text for which you are searching, select Match case .
Find a specific string of text in the script and replace it with another string of text	Click the defined and Replace button. Enter the search text for which you are searching in the Find what field, and then enter the replacement text in the Replace with field. Once you have entered the required text, click Replace , Replace All , or Find Next .
	To find and replace script text in a specific block of script, select the block of script first and then click the the ab Find and Replace button. At the Find and Replace dialog box, select Replace only in selected text .
Move your cursor to a specific location in the script	Click the Go To button. This displays the Go To dialog box, where you can enter the line number or character position of where you want your cursor to move.
Indent or outdent a block of the script	Select the portion of the script you want to indent or outdent and click the Findent button or the Findent button. (Click repeatedly to increase or decrease the indent.)
Cause HotDocs to ignore a section of the script when it processes it, or insert a comment in the script	Highlight the block of the script you want to comment and click the Comment Block button. This puts two forward slashes in front of each line of the script, which instructs HotDocs to ignore this section. To uncomment it, highlight the text and click the EUncomment Block button.
Match an IF or REPEAT instruction with its END IF or END REPEAT instruction (or vice versa)	Place your cursor inside the instruction, right-click, and select Match IF/REPEAT from the shortcut menu. (You can also press Ctrl+M .)

Highlight an entire IF or REPEAT instruction block (meaning everything between a beginning and ending IF/REPEAT instruction)	Place your cursor inside the instruction, right-click, and select Select IF/REPEAT from the shortcut menu. (You can also press Ctrl+Shift+M .)
View helpful information while using the script editor, including accessing help topics for the different instructions and expressions you can use in your script	Click the HotDocs Help button.
Customize the way the script editor works	Click the Options button and make your changes. (See Change Script Editing Options.)
Insert line breaks and tab characters in a script	See Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts.
Assign formatting characteristics to literal text strings in a script	Insert the corresponding dot code. See one of the following topics for details:
	Change Font Properties of Text
	Insert Characters in Text Strings
	Add Punctuation and Capitalization to Sentences

To access the script editor toolbar using the keyboard, press F10.

To access help for each instruction or expression, first select the model in one of the lists and then press **Ctrl+F1**.

For an explanation of the HotDocs scripting language, see Understand the HotDocs Scripting Language. For details on creating a Computation variable, see Customize a Computation Variable. For a list of instruction and expression models, see Introduction Instruction and Expression Models.

Local Variables

When writing more complex computation scripts, you may need to create temporary or special-purpose variables that are only used or needed in a particular computation or dialog script. Such temporary variables create a maintenance burden since they always show up in component lists and can require

extra explanation for whoever maintains your templates in the future. To make your HotDocs scripting more maintainable, and easier to read, use local variables in these situations.

A **local variable** is somewhat similar to a regular variable, but instead of being usable anywhere in your template, it is only defined in the "local" context of a specific computation or dialog script. Local variables are useful in scripting situations where you would otherwise resort to defining a regular HotDocs variable just to keep track of temporary information that is not meaningful anywhere else but in this one script.

As an example, suppose you're writing a computation to remove spaces that may have been entered as part of an account number. You could do this using a WHILE instruction (link) to loop through each character of the account number, removing spaces as you find them:

```
SET Index TO 1
WHILE Index <= LENGTH( Account Number )
    IF MID( Account Number, Index, 1 ) = " "
        SET Account Number TO FIRST( Account Number, Index -1 )
        + LAST( Account Number, LENGTH( Account Number) - Index)
    ELSE
        INCREMENT Index</pre>
```

END IF

END WHILE

Notice how this script relies heavily on the "Index" number variable. It keeps track of our position as we look through each character in the account number. This is a good example of a variable that should be defined "locally" instead of being defined in the Component Manager (as you would have done in earlier versions of HotDocs). As a local variable, Index is meaningful (and has an answer) only in the context of this script. You can create a local variable called "Index" in another computation as well, but the two would be isolated: setting one computation's local Index to a certain value would have no effect on the other computation's local Index. In fact, local variables do not store their answers in the answer file like regular HotDocs variables. These answers are stored in a special temporary location and are discarded as soon as the computation or script finishes running.

Here is a table that summarizes the differences between traditional HotDocs variables and local variables:

	Regular Variables	Local Variables
Defined in	Component Manager	The Locals tab of a Computation Editor or Dialog Editor window
Usability / Scope	Anywhere: in the template itself, in any script, or in other variables' prompts or resources	Only in the script where it is defined

Naming rules	Name must be unique across all the components in the component file	Name must only be unique within the script where it's defined
Available types	Text, Number, Date, True/False, Multiple Choice	Text, Number, Date, True/False
Shows up in general component lists?	Yes	No
Can be asked on a dialog?	Yes	No
Can be saved in an answer file?	Yes	No
Initial or default value	UNANSWERED	UNANSWERED

You can use regular HotDocs variables (perhaps with "Ask automatically", "Warn when unanswered" and "Save in answer file" all turned off) in every instance where you could use a local variable. However, there are advantages to using local variables when possible, and these advantages become more significant as your template automation projects grow larger and more complex:

- Large or complex component files are easier to maintain when component lists are not cluttered with temporary variables.
- Naming local variables is easier, and scripts using local variables can be easier to read, because the name only has to make sense in the context of the script where it's defined and not in a global component list. In the above example, it is realistic to call the local variable "Index" instead of needing to call it "Account Number Index Temp," for example.
- Reusing components by copying them from one component file to another becomes less complicated because such copy operations no longer need to bring along additional temporary variables.

To Create a Local Variable

- 1. Insert a variable, selecting **Computation** as the variable type.
- 2. At the Variable Field dialog box, click the **Edit Component** button to open the **Computation** Editor.
- 3. Click on the **Locals** tab.
- 4. Create a local variable
 - 1. Clicking the left-hand column of the **Parameter** table and enter a name
 - 2. Click the right-hand column to select the type of variable this will represent.
- 5. Click the **Properties** tab and enter the computation you want to use your local variable in. Drag and drop from the **Parameters and Local Variables** list at the bottom left.

You can tell at a glance whether local variables or parameters have been defined for the current computation or dialog script by looking at the Locals tab. If the Locals tab has an asterisk (*) on it, at least one local variable or parameter has been defined.

To view the defined local variables and drag them into your script as necessary, choose "Parameters and Local Variables" from the Components drop-down in the bottom left of the script editor.

You can define a local variable that has the same name as a regular variable elsewhere in the component file. In such a situation, HotDocs recognizes the local name before looking for the other component. For example, suppose you have a computation variable that defines a local variable called "Temp Number," and there is also a number variable called "Temp Number" defined in Component Manager. If you refer to "Temp Number" within the computation where the local is defined, you will get the local answer associated with it. However, referring to "Temp Number" in the template, or in another script, will retrieve the regular answer from the answer file.

Parameters

When building complex templates or template systems, you may need to duplicate portions of script or logic multiple times. These duplications may be exact or with minor changes. Duplicating script or logic makes templates difficult to maintain because as times goes by and the logic needs refinement, you may introduce errors if you do not update each instance of the logic consistently. To make your HotDocs scripting more maintainable, and often easier to read, encapsulate reused logic in computation variables that use parameters.

A **parameter** is like a local variable for a computation, but instead of initializing the parameter in your script (as you might initialize a local variable using the SET instruction), the initial value of the parameter is copied *into* your computation from wherever the computation was invoked.

For example, this computation determines whether a particular child is under 18 years of age:

Computation name: Decedent Child Under Eighteen

Script:

AGE(Decedent Child Birthdate) < 18

The part of the script that determines whether a child is a minor may need to be duplicated in numerous places. You could define similar computation variables that only differ in the date variable to which they are referring, causing unnecessary extra work and cluttering up your components list, or you could use parameters.

Parameters allow you to write the logic only once:

Computation name: "Under Eighteen"
Parameter: "person" (Date)
Script:
 AGE(person) < 18</pre>

Instead of referring to another date variable ("Decedent Child Birthdate" in the former example), the computation refers to a local parameter. So, whenever this latter computation is referred to, instead of simply referring to it by name, we must also provide a date as "input" to the computation:

Under Eighteen (Decedent Child Birthdate)

The very same computation can be referred to ("called") from elsewhere using different date variables or date expressions:

```
Under Eighteen ( Beneficiary Birthdate )
Under Eighteen ( Beneficiary Birthdate + 5 YEARS )
```

In this way, parameters essentially allow you to write computations that work much like HotDocs' built-in expression models. They allow you to re-use script logic rather than duplicating entire computation variables to perform the same action on different values. This is very much like defining functions within other programming environments.

As a more complex example, let's consider a longer computation that calculates the date of the nearest Monday on or after a given date. It uses a parameter and a local variable.

```
Computation Name: "Next Monday"
Parameter: "origin date" (Date)
Local Variable: "days away" (Number)
Script:
// Find the number of days between origin date and Monday (which is day of week
no. 2)
SET days away TO (2 - DAY OF WEEK( origin date ))
// If origin date is later in the week than Monday, add 7 days to get to next
week
```

IF days away < 0
 SET days away TO days away + 7
END IF
// calculate the date
origin date + days away DAYS</pre>

A computation with parameters can be referred to (called) from any other computation or script. It can also be used in merge fields, either directly in a template or in literal text: «Next Monday(TODAY):04 July 2012» will use the above computation variable (Next Monday) to calculate the next Monday after the current date (TODAY), format it according to the given format example (04 July 2012), and merge the result wherever the field appears (whether in a template, prompt, resource, etc.).

When you add, remove, or change the parameters of a computation, you are changing the way that computation is referred to by its callers. In this way, changing the types or order of a computation's parameters is akin to changing the computation's name: care should be taken to not break things.

For users with a prior knowledge of programming it may be useful to think of HotDocs Parameters as Value Parameters. Therefore any change made in the script does not change any variable that has been passed in when calling the computation.

Computations used as filters on REPEAT instructions are not permitted to have parameters.

To create a computation variable that accepts a parameter:

- 1. Insert a variable, selecting **Computation** as the variable type.
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- Click on the Locals tab and create a parameter by clicking on the left hand column of the Parameter table and entering a name, then the right hand column to select the type of variable this will represent.
- 4. Click on the **Properties** tab and enter the computation that you would like your new parameter to complete. In the computation use the parameter (drag and drop from the **Parameters and Local Variables** list at the bottom left) as the placeholder for the variables you would like to use later.
- 5. Click **OK** on the open dialogs to return to the template.

To use this computation variable from another computation or script,

- 1. Open the **Computation Editor** for the computation from which you wish to use your computation with parameters.
- 2. From the list of **Available Components** in the lower left corner, drag and drop your computation with parameters into the script. HotDocs will automatically generate placeholders for each of the parameters required by the computation.

3. Drag and drop variables or values to replace the placeholders as you would do when replacing placeholders for the built-in expression models.

To use a this computation variable from a field in a template

- 1. Click on the **Variable Field** button on the HotDocs tab and select Computation from the **Type** list.
- 2. At the **Variable** field use the drop-down list to find and click on the parameterized computation. You will now see a table with available parameters, and an empty **Expression** column.
- 3. Click the button to the right of the table and drag the variable you would like to use from the **Variables** list up into the **Expression** field. (You can also manually enter or edit expressions here.)

To test a computation that includes parameters

- 1. Create a computation that includes parameters.
- 2. Click the **Test** button.
- 3. To facilitate testing, HotDocs presents an automatically-generated dialog to gather values for the computation's parameters. Enter a value for each parameter. (Note that this dialog is only part of a test assembly for the computation. In actual usage, the call to the computation would specify the parameter values.)
- 4. If your computation causes subsequent dialogs to be asked, enter values for those additional variables as well.
- 5. Switch to the **Result** tab as necessary to view the result of your computation given the values you entered.
- 6. Switch back to the **Interview** tab to test results given other input.

Just as with regular computations, you can also use the DEBUG instruction to step through a computation line by line and observe how the values of parameters and other variables are affected by your script.

Using Instruction Models

Full List of Instruction Models

Instruction	Description
ADD TEXT TO MULT CHOICE VAR;	These instructions allow you to modify options of a Multiple Choice variable. The CLEAR instruction removes all the current options, and the
CLEAR	ADD instruction adds options to the variable.
MULT_CHOICE_VAR	

ASCEND VAR; DESCEND VAR	The ASCEND instruction sorts lists of answers (gathered using a REPEAT instruction) in alphanumeric order, from 1 to 9, and from A to Z. The DESCEND instruction sorts lists of answers from 9 to 1, and from Z to A.
ASK DIALOG	The ASK DIALOG instruction allows you to control the order in which dialogs appear in an interview. (See Control When Your Dialogs Appear.)
ASK VAR	Sometimes a variable needs to be asked by itself. You can use the ASK VAR instruction so that during the interview, HotDocs displays the variable in its own default dialog.
ASSEMBLE "FILENAME"	You can use the ASSEMBLE instruction to add templates to the assembly queue. Unlike the INSERT instruction, an ASSEMBLE instruction waits until the current document is assembled before starting the next assembly session.
CONCEAL VAR	This instruction, which you use in a dialog script, keeps variables from appearing in the Select From Answer Source dialog box of an answer source.
DEBUG	This instruction steps through the template or script field by field or line by line. It helps you determine why the template you are automating (or the script you are writing) is producing results you don't expect.
DEFAULT VAR TO VALUE	This instruction suggests a value for a variable if the variable is unanswered.
ERASE VAR; ERASE DIALOG	The ERASE instructions let you clear answers in a dialog. Specifically, ERASE VAR clears answers for a specified variable in a dialog, while ERASE DIALOG clears all answers in the dialog. This may be useful when you are using a temporary dialog to store lists of answers from two or more dialogs.
FILTER COMPUTATION_VAR	The FILTER instruction filters out certain entries from a repeated list, based on conditions you specify.
FORMAT "LIST_FORMAT"	The FORMAT "LIST_FORMAT" instruction allows you to create a sentence- style list within a computation. (If you create the REPEAT instruction using a REPEAT field, you can specify the list format by choosing a style from the Format drop-down list. See Punctuate a Sentence-Style List.)
GRAY ALL; GRAY VAR; UNGRAY ALL; UNGRAY VAR	These instructions, which are used in a dialog script, control whether components in a dialog appear grayed or ungrayed, depending on answers a user enters. GRAY ALL dims all components in the dialog, while UNGRAY ALL makes all of the components active again. Likewise, GRAY VAR dims a single component, and UNGRAY VAR enables the component again.
HIDE ALL; HIDE VAR; SHOW ALL; SHOW VAR	These instructions, which are used in a dialog script, control whether the user is able to see variables in a dialog. The HIDE ALL instruction hides all variables in the dialog, while SHOW ALL reveals the variables again. Likewise, HIDE VAR hides a single variable, and SHOW VAR reveals the variable.

IF EXPRESSION; ELSE IF; ELSE; END IF	You can make sections of templates or scripts conditional by using IF instructions. A conditional section will be included only if a condition you specify is <i>true</i> . The ELSE IF instruction allows two or more conditions to be included in an IF instruction. The ELSE instruction establishes a final condition for an IF instruction, specifying that if all preceding conditions are false, the following information should be included. It must be the last item of the IF instruction.
INCREMENT NUM_VAR; DECREMENT NUM_VAR	The INCREMENT and DECREMENT instructions cause HotDocs to increase or decrease a number variable, usually a counter, by the value of <i>1</i> .
INSERT "FILENAME"	This instruction inserts a clause, a clause library, or a template into the document currently being assembled. When HotDocs encounters an INSERT instruction, it immediately processes the instruction and inserts the template, clause, or clause library into the current document. If there are variables to be answered, HotDocs presents them before finishing the interview of the main document.
LANGUAGE CODE	This instruction tells HotDocs to format numbers and dates in a particular language.
LIMIT NUM	The LIMIT instruction limits the number of times a dialog can be repeated. It is placed in the script of the dialog that must be limited to a specific number of repetitions.
OMIT VAR	The OMIT VAR instruction, which you use in a dialog script, keeps variables from appearing in the Edit Answer Source dialog box of an answer source.
PLAY "MACRO"	The PLAY "MACRO" instruction plays a word processor macro after the document is assembled and either sent to the word processor, printed, or saved.
QUIT	The QUIT instruction allows you to close the variable without losing the work you have done. Normally, HotDocs will not save an invalid computation. The only way to exit an invalid computation is to click Cancel , which erases the script. The instruction is usually placed at the beginning of an unfinished or invalid computation.
REPEAT DIALOG; END REPEAT	A REPEAT instruction gathers lists of answers and merges them into a document.
REQUIRE ALL; REQUIRE VAR	The REQUIRE instruction requires users to answer questions in a dialog before they can advance to the next dialog in the interview.
SET VAR TO VALUE	This instruction lets you specify a given value for a variable's answer automatically, rather than allow the user to specify an answer. With the SET instruction, you can transfer names and other values from one variable to another.
WHILE EXPRESSION; END WHILE	The WHILE EXPRESSION instruction allows you to repeatedly process (or loop through) an answer or set of answers until a certain condition is met, such as a certain answer is found or a limit is reached.

ADD TEXT TO MULT_CHOICE_VAR; CLEAR MULT_CHOICE_VAR

Placeholder	Replace With
TEXT	A text value, such as a Text variable or an actual word or name
MULT_CHOICE	Any Multiple Choice variable in the template

These instructions allow you to modify options of a Multiple Choice variable. The CLEAR instruction removes all the current options, and the ADD instruction adds options to the variable.

Using these two models, you can create an entire Multiple Choice variable using answers the user provides.

For example, a template requires the user to first enter a list of all the committee member names. Later, the user can identify which committee member is the chairperson. You can allow the user to identify the chairperson by presenting a Multiple Choice variable that has the names of all committee members as options. The following script would create that Multiple Choice variable:

CLEAR Committee Chairperson REPEAT Committee Members List ADD Member Name TO Committee Chairperson

END REPEAT

In this example, the CLEAR instruction first removes any existing options from the Multiple Choice variable *Committee Chairperson*. Then, the script repeats the *Committee Members List* dialog, gathering the names of each committee member. The ADD instruction then adds each member's name to the Multiple Choice variable. After the committee members have all been entered, you can present the *Committee Chairperson* variable for the user to identify the chairperson.

Additionally, if the Multiple Choice variable you are clearing uses any option prompts, these prompts will also be cleared. When adding new options to the Multiple Choice variable, you can also add new prompts. To do this, use a vertical bar to separate the option from the prompt in the ADD instruction. For example:

CLEAR Marital Status ADD "Single|Client is single" TO Marital Status ADD "Married|Client is married" TO Marital Status ADD "Divorced|Client is divorced" TO Marital Status In this example, the Multiple Choice variable *Marital Status* is cleared. The script then adds a literal text value (denoted by quotation marks) for the new option. The option is immediately followed by a vertical bar, which tells HotDocs to use the text following the bar as a prompt for the option.

Both the CLEAR and ADD instructions must be used in a computation script, which must be processed before the Multiple Choice variable is asked. To add options to a Multiple Choice variable, you should create the variable beforehand and assign a temporary option. Then, as the user provides the answers you want to include as options, the CLEAR instruction removes the temporary option, and the ADD instruction places the user's answers as options in the variable.

Unlike options and prompts, the merge text for a Multiple Choice variable comes from another component—a Merge Text component. This means you cannot create and add merge text to a Multiple Choice variable "on the fly."

ASCEND VAR; DESCEND VAR

Placeholder	Replace With
VAR	A repeated variable

The ASCEND instruction sorts lists of answers (gathered using a REPEAT instruction) in alphanumeric order, from 1 to 9, and from A to Z. The DESCEND instruction sorts lists of answers from 9 to 1, and from Z to A.

For example, the following script would insert a list of clients in alphabetical order, from A to Z. Even though it lists *Client First Name* first, it sorts by *Client Last Name*:

"" REPEAT Client Information
 ASCEND Client Last Name
 RESULT + Client First Name + " " + Client Last Name + "
"
END REPEAT

The first empty set of quotation marks sets the computation value to nothing. Then, the repeated dialog, *Client Information*, asks for each client's first and last name. Next, the ASCEND instruction sorts the list of names by last name. Finally, the RESULT expression tells HotDocs to merge the names in the document.

The ASCEND and DESCEND instructions can only **sort** on a single variable; however, you can sort multiple variables by including multiple ASCEND or DESCEND instructions.
ASK DATABASE



The ASK DATABASE instruction allows you to control when the record selection screen of a database component is shown in an interview. (See Control When Your Record Selection Screens Appear.)

In general, the ASK instruction tells HotDocs to display a record selection screen or a dialog as soon as the instruction is processed. You can insert an ASK instruction directly in the template, or use a Computation variable. In fact, you can control the entire interview by using a series of ASK and other instructions in a single computation. (See ASK DIALOG and Define a Custom Interview for details.)

ASK DIALOG

Placeholder	Replace With
DIALOG	Any dialog in the template

The ASK DIALOG instruction allows you to control the order in which dialogs appear in an interview. (See Control When Your Dialogs Appear.)

ASK instructions are also useful when some interview questions should only be asked in certain situations. In the following example, HotDocs only asks the *Buyer Information* dialog if the user is a first-time buyer:

«IF F	irst	Time	Buyer»
«ASK	Buyer	Info	ormation»
«END	IF»		

In general, the ASK instruction tells HotDocs to display a dialog as soon as the instruction is processed. You can insert an ASK instruction directly in the template, or use a Computation variable to ask several dialogs at once. In fact, you can control the entire interview by using a series of ASK and other instructions in a single computation. (See Define a Custom Interview for details.)

ASK VAR

Placeholder	Replace With
VAR	Any variable in the template

Sometimes a variable needs to be asked by itself. You can use the ASK VAR instruction so that during the interview, HotDocs displays the variable in its own default dialog.

In the following example, an IF expression evaluates if more than a year has passed since the last time the client assembled this document. If so, the *Insurance Company Name* variable is asked:

IF MONTHS FROM(Date Of Previous Filing , TODAY) >= 13

ASK Insurance Company Name

END IF

You can insert the ASK VAR instruction directly in the template or you can use it in a Computation variable. Be aware, however, that if you insert the instruction directly in the template, you can ask only one variable, and any answers the user provides will not be merged into the document at that place.

ASSEMBLE "FILENAME"

Placeholder	Replace With
FILENAME	The name of another HotDocs template

Complete instructions on using the ASSEMBLE instruction can be found in the topic, Start a New Assembly From a Template.

You can use the ASSEMBLE instruction to add templates to the assembly queue. Unlike the INSERT instruction, an ASSEMBLE instruction waits until the current document is assembled before starting the next assembly session.

For example, you may or may not need to include a cover letter with the document you are assembling. If you do, the following script adds the template, *Cover Letter*, to the assembly queue.

«IF Cover Letter Required»

«ASSEMBLE "Cover Letter.docx"» «END IF»

In this script, if the user opts to assemble a cover letter, then the ASSEMBLE instruction is processed and HotDocs assembles the cover letter template after the main document is assembled.

Understand How Templates Are Added to the Assembly Queue

To achieve the best results when using the ASSEMBLE instruction, you should understand how HotDocs adds the templates it finds in ASSEMBLE instructions to the assembly queue—particularly if your template uses a custom interview. There are two different scripts that HotDocs processes during an assembly:

- The *template* script is the body of the template where HotDocs merges the answers to produce the assembled document.
- The *interview* script is the set of instructions that determine what dialogs to include in the interview. If your template uses a custom interview, this is your custom interview computation (e.g., INTERVIEW). Otherwise, the interview script is the same as the template script for a default interview.

Each time HotDocs processes one of these scripts, it first removes from the Assembly Queue any templates that were added by ASSEMBLE instructions in the current template. Then, as HotDocs encounters each ASSEMBLE instruction, it adds that template to the assembly queue. Thus, if you have an ASSEMBLE instruction in your custom interview script, the template in that instruction will be added to the assembly queue whenever HotDocs processes your interview script.

This behavior can cause a problem if the ASSEMBLE instructions in your interview script do not match the instructions in the template script. For example, if HotDocs adds templates to the assembly queue while processing the interview script, those templates could be removed when HotDocs subsequently processes the template script. Likewise, if your template script adds templates to the assembly queue, they could be removed when the interview script is processed.

To ensure that templates are added to, and remain in, the assembly queue, you should include identical ASSEMBLE instructions in both your interview and template scripts. This way, no matter which script HotDocs processes last, it will put the desired templates in the assembly queue. (As noted above, both scripts are identical if you use a default interview, so this information is only important when you use a custom interview in your template.)

You can add command-line options to an ASSEMBLE instruction (for example, ASSEMBLE "subpoena.docx /pr"). If the command-line option includes a file path and name, enclose the path and name in double quotation marks (for example, ASSEMBLE "subpoena.docx /sa /af=""L Chang.anx"""). (Four command line options were designed specifically for use with ASSEMBLE instructions. They are: Suggest Save, Suggest Save New, Save Answers, and Save Answers Prompt. They control the saving of answers after each ASSEMBLE instruction is processed.)

CONCEAL VAR

Placeholder	Replace With
VAR	A variable used in an answer source

This instruction, which you use in a dialog script, keeps variables from appearing in the **Select From Answer Source** dialog box of an answer source.

Answer sources are associated with specific dialogs in a template. When users view the dialog during the interview, they can click a special button, which displays a list of answers that were entered during previous interview sessions. They can either select an existing set of answers or add a new set to the list.

After creating an answer source for one dialog, you can use it with other dialogs, including dialogs in other templates. However, when using an answer source with multiple dialogs, each variable must be represented in both the answer source and in each dialog. If a variable that is referenced in the answer source isn't included in the dialog (or vice-versa), answers in the answer source will get "mixed together" whenever you add, edit, or delete records.

In some situations though, it isn't always practical or relevant to show the user every variable—either in the dialog or in the answer source. To accommodate this, you can use the HIDE, CONCEAL, and OMIT instructions to manipulate these variables in both the dialog and the answer source. Specifically, HIDE keeps a variable from appearing on the dialog, while CONCEAL keeps it from appearing in the answer source. OMIT keeps it from being associated with the answer source at all. Often, you must use a combination of these instructions to achieve your desired result.

For example, in one template, the user must enter both a *Creditor Name* and a *Creditor Address*. However, in a second template that uses the same dialog and answer source, *Creditor Address* isn't needed. You can keep it from appearing in the **Select From Answer Source** dialog box for this template by using the following script:

CONCEAL Creditor Address

Answers for *Creditor Address* are still saved in the answer source, even though they do not appear when the user opens the **Select From Answer Source** dialog box. (You would most likely include the instruction *HIDE Creditor Address* in this script as well, which would keep *Creditor Address* from appearing on the actual interview dialog. To keep the variable from appearing when the user edits a record in the answer source, see OMIT VAR.)

See Suggest an Answer Source for Dialogs for more information.

DEBUG

This instruction steps through the template or script field by field or line by line. It helps you determine why the template you are automating (or the script you are writing) is producing results you don't expect.

For full information on how to use this instruction, please see the following topics:

- Introduction: Debugging Templates
- Insert Debugging Instructions in Templates and Scripts
- Step Through a Template or Script

DEFAULT VAR TO VALUE

Placeholder	Replace With
VAR	Any type of variable in the template
VALUE	A value that corresponds with the variable type

This instruction suggests a value for a variable if the variable is unanswered.

For example, in the following script, the variable *Attorney Name* is defaulted to the literal value of *Sam Jones*:

DEFAULT Attorney Name TO "Sam Jones"

In this example, when HotDocs processes this script, it first determines whether *Attorney Name* has been answered. If it has, the DEFAULT instruction has no effect and HotDocs uses the answer already given. If *Attorney Name* has not been answered, however, HotDocs suggests the answer *Sam Jones*. When this variable appears during the interview, users can accept this answer by moving to the next dialog, or they can enter a different answer in the answer field.

You can also use a DEFAULT instruction to suggest an answer that has already been given in the interview. For example:

DEFAULT Trustee Name TO Client Name

In this example, when HotDocs processes this script, it first determines whether *Trustee Name* has been answered. If it has, the DEFAULT instruction has no effect and HotDocs uses the answer already given. If *Trustee Name* has not been answered, HotDocs then checks to see what answer has been given for *Client Name* and suggests that as the answer. It is important to note, however, that if *Client Name* is unanswered, *Trustee Name* will likewise be unanswered.

Do not use the DEFAULT instruction in the script of a repeating dialog unless the instruction is used in conjunction with a conditional expression (on another variable in the same dialog, at the same REPEAT index) or a **LIMIT** instruction. If you use it by itself in a repeated dialog script, it will always add an unanswered dialog to the interview, which will produce an incorrectly assembled document.

The DEFAULT and SET instructions both assign answers to variables. Click here for an explanation of the differences between the two.

To default two or more options for a Multiple Choice variable, separate each option with a vertical bar (for example, DEFAULT MC Variable TO "Option1|Option2|Option3").

ERASE VAR; ERASE DIALOG

Placeholder	Replace With
VAR	The name of a variable in a dialog you want to clear
DIALOG	The name of the dialog whose contents you want to clear

The ERASE instructions let you clear answers in a dialog. Specifically, ERASE VAR clears answers for a specified variable in a dialog, while ERASE DIALOG clears all answers in the dialog. This may be useful when you are using a temporary dialog to store lists of answers from two or more dialogs.

For example, say you have two repeated dialogs—one containing plaintiff names and another containing defendant names. If you need to generate a single list of all parties in the case, you can combine the two lists into a single repeated dialog. To do this, you would want to erase any existing values from the combined list before populating it with the names from the plaintiff and defendant dialogs.

Using the ERASE instruction in the script keeps the combined list up to date each time the computation script is processed. For example, if the user adds or removes names in the plaintiff or defendant dialogs during the interview, the ERASE instruction will make sure they are properly added or removed when the combined list is regenerated.

The following script demonstrates how to accomplish this:

ERASE Combined List SET Counter TO 0 REPEAT Plaintiff Information INCREMENT Counter

SET Combined Name[Counter] TO Plaintiff Name END REPEAT REPEAT Defendant Information INCREMENT Counter SET Combined Name[Counter] TO Defendant Name END REPEAT

FILTER COMPUTATION_VAR

Placeholder	Replace With
COMPUTATION_VAR	Any Computation variable in the template that results in a true or false value

The FILTER instruction filters out certain entries from a repeated list, based on conditions you specify.

In the following example, the script filters out all corporate entities from a list of vendors:

....

REPEAT Vendor Information

FORMAT "A, B, and C"

FILTER No Corporate Vendors

RESULT + Vendor Name

END REPEAT

In the preceding script, No Corporate Vendors is a Computation variable with the following script:

Vendor Type != "Corporation"

First, the empty quotation marks set the value of the repeat to "nothing." HotDocs then repeats the *Vendor Information DI* dialog. After the user enters all the information, HotDocs processes the responses, filters out all corporate entities, and displays a modified list in the specified format.

You can use the **AND** operator in the computation to filter out entries based on two or more conditions.

A filter can be as complicated as it needs to be, but it must result in either true or false. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a true or false value—it is not a filter. But the expression YEARS FROM(Child's Birth Date, TODAY) <= 17 can only result in true or false. It can correctly filter all children under the age of 18 from a list.

FORMAT "LIST_FORMAT"

Placeholder	Replace With
LIST_FORMAT	An example of the conjunction and punctuation desired, for example, <i>a</i> , <i>b</i> , and <i>c</i> or <i>a</i> ; <i>b</i> ; or <i>c</i> .

The FORMAT "LIST_FORMAT" instruction allows you to create a sentence-style list within a computation. (If you create the REPEAT instruction using a REPEAT field, you can specify the list format by choosing a style from the **Format** drop-down list. See Punctuate a Sentence-Style List.)

For example, perhaps you want the items in your list to appear with the final comma preceding the *and*. Your script would look like this:

"" REPEAT Education Information FORMAT "A, B, and C" RESULT + Degree Designation END REPEAT RESULT

Once a user has provided the list of educational degrees in **Education Information**, the FORMAT instruction ensures that the requisite commas are in place.

FORMAT instructions should be placed immediately after the REPEAT instructions for repeated dialogs, and before other instructions influencing the REPEAT (such as **FILTER** or **ASCEND** and **DESCEND**.

GRAY ALL; GRAY VAR; UNGRAY ALL; UNGRAY VAR

 Placeholder
 Replace With

 VAR
 Any variable used in a dialog

These instructions, which are used in a dialog script, control whether components in a dialog appear grayed or ungrayed, depending on answers a user enters. GRAY ALL dims all components in the dialog, while UNGRAY ALL makes all of the components active again. Likewise, GRAY VAR dims a single component, and UNGRAY VAR enables the component again.

For example, say you want certain variables in a dialog to appear based on the user's family situation. How the user answers the Multiple Choice variable, *Family Status*, controls which questions are asked:

GRAY ALL UNGRAY Family Status IF Family Status = "Married, with children" UNGRAY ALL ELSE IF Family Status = "Married, no children" UNGRAY Spouse Name ELSE IF Family Status = "Separated, with children" UNGRAY Number Of Children END IF

The initial GRAY ALL instruction dims all variables in the dialog, making them inactive. Then the *Family Status* Multiple Choice variable is immediately ungrayed so the user can choose an option. Depending on the answer to this Multiple Choice variable, some or all of the other variables are ungrayed.

In the script above, you could also gray each variable in the dialog (except the Multiple Choice variable); however, graying all of the variables and then ungraying the Multiple Choice variable immediately after saves a lot of repetitive typing.

HIDE ALL; HIDE VAR; SHOW ALL; SHOW VAR

Placeholder	Replace With
VAR	A variable used in an answer source

These instructions, which are used in a dialog script, control whether the user is able to see variables in a dialog. The HIDE ALL instruction hides all variables in the dialog, while SHOW ALL reveals the variables again. Likewise, HIDE VAR hides a single variable, and SHOW VAR reveals the variable.

In the following example, a certain estate planning template may be used for both wills and trusts. The initial dialog uses a Multiple Choice variable called *Document Type* to ask which type of document will be assembled. Then, depending on how the user answers the variable, HotDocs asks either the executor/testator names or the trustee/grantor names:

```
HIDE ALL

SHOW Document Type

IF Document Type = "Will"

SHOW Executor Name

SHOW Testator Name

ELSE IF Document Type = "Trust"

SHOW Trustee Name

SHOW Grantor Name

END IF
```

You should not HIDE or SHOW a variable in a repeated-as-spreadsheet dialog based on another variable in the same dialog. When you use HIDE or SHOW in a spreadsheet dialog, the entire column is shown or hidden. Attempting to do this may produce unexpected results.

IF EXPRESSION; ELSE IF; ELSE; END IF

Replace With		

EXPRESSION A statement that can be evaluated as true or false

You can make sections of templates or scripts conditional by using IF instructions. A conditional section will be included only if a condition you specify is *true*. The ELSE IF instruction allows two or more conditions to be included in an IF instruction. The ELSE instruction establishes a final condition for an IF instruction, specifying that if all preceding conditions are false, the following information should be included. It must be the last item of the IF instruction.

Each IF instruction or expression must end with an END IF instruction. This instruction completes a section of conditional logic. HotDocs automatically creates an END IF paired with each IF instruction. These pairs can be nested, allowing you to test several conditions before applying a single effect.

For example, in the following script, HotDocs uses an IF instruction to insert a paragraph about vacation time—but only if the new employee qualifies for paid vacation:

«IF Vacation Days»In addition, «Employee Name» shall be allowed «Number of Vacation Days:ten» for vacation time.

«Employee Name» shall also receive seven paid holidays, including New Year's Day, Memorial Day, Fourth of July, Labor Day, Thanksgiving (including the day after), and Christmas.«END IF»

More complex situations can also be handled using IF expressions. For example, in the following computation script, a single paragraph in a template may change depending on how close a project is to completion. Using the IF, ELSE IF, and ELSE instructions, the correct paragraph can be inserted:

IF Project Status = "Complete"
 "Upon finishing the project..."
ELSE IF Project Status = "In Process"
 "While working on the project..."

ELSE

"Before working on the project..."

END IF

You can use IF expressions anywhere. Operators such as **AND** and **OR** can link multiple conditions, giving the **user** greater control over the **interview**. However, creating complicated IF instructions directly in the template can make the assembly process sluggish. Consider using Computation variables or INSERT instructions instead.

In past versions of HotDocs, using END by itself instead of the complete END IF instruction was sufficient. Beginning with 11.1, HotDocs no longer processes the shortened version, and now requires the complete instruction.

Please see Introduction: Make Parts of Templates Conditional for further information on using IF instructions and expressions.

INCREMENT NUM_VAR; DECREMENT NUM_VAR

Placeholder Replace With

NUM_VAR A Number variable

The INCREMENT and DECREMENT instructions cause HotDocs to increase or decrease a number variable, usually a counter, by the value of 1.

In the following example, you want to create a list of potential employees. However, you want the list to include only those applicants with four or more years of schooling. To do this, you would use the WHILE instruction to loop through a list of applicants. You would then use the INCREMENT instruction to keep track of which repetition you are on so that the correct information can be merged into a new list.

SET Applicant Count TO 1

SET Prospect Count TO 0

WHILE ANSWERED(Applicant Name[Applicant Count])

IF Applicant Years of Schooling[Applicant Count] >= 4

INCREMENT Prospect Count

SET Prospect Name[Prospect Count] TO Applicant Name[Applicant Count]

END IF

INCREMENT Applicant Count

END WHILE

INSERT "FILENAME"

Placeholder	Replace With
FILENAME	The file name and folder path of another HotDocs template
	Include the folder path if the inserted template resides in a folder that is different from the main template.

The INSERT instruction inserts a clause, a clause library, or a template into the document currently being assembled.

Templates can be inserted from any location, as long as you specify the correct folder path information.

The INSERT instruction differs from the ASSEMBLE instruction in that HotDocs immediately processes the INSERT instruction and inserts the template, clause, or clause library into the current document. If there are variables to be answered, HotDocs presents them before finishing the interview of the main document.

To create a simple INSERT instruction, you can click the **#INSERT Field** button and insert the instruction directly in the template. However, if you want to insert a more complex instruction using conditional logic, you may need to use a Computation variable.

Often an organization's documents include sections that are used frequently (boilerplate language), such as company letterhead or attorney information blocks. Rather than recreate these parts of a document each time you automate a template, you can save just the letterhead or the information block as its own template and then use an INSERT instruction to include it in the templates that require it. For example, let's say you have a specific attorney information block you use in pleadings. You would create a template that contains only the attorney information and then use the INSERT instruction to insert it:

«IF Attorney Information Required» «INSERT "Attorney Information Block.docx"» «END IF»

In text templates, if the inserted template contains unanswered variables, HotDocs asks these variables. After the inserted template is assembled, HotDocs finishes assembling the main template. In contrast, with form templates, HotDocs appends the inserted form template to the main form. Once it finishes assembling the main form, it then assembles the inserted form template.

Inserted text templates may have formatting that differs from the original template. Headers, footers, and margins can often be controlled more easily by using word processor section breaks.

LANGUAGE CODE

Placeholder	Replace With
CODE	Any of the following language codes:

ENG (English)
DEU (German)
DES (Swiss German)
DEA (Austrian German)
FRA (French)
NLD (Dutch)
ESN (Spanish)
ITA (Italian)
PTB (Brazilian Portuguese)

This instruction tells HotDocs to format numbers and dates in a particular language.

For example, the following script allows the template to use Spanish formats:

«LANGUAGE ESN» «Start Contract Date:3 Juno 1990»

Optionally, if your date or number format requires non-U.S. thousands and decimal separators, you can specify which separators you want to use in the LANGUAGE instruction. The first character must be the thousands separator and the second character must be the decimal separator. For example:

«LANGUAGE ".," FRA»

For more detailed information on using foreign language templates, see Create a Foreign Language Template.

LIMIT NUM

Placeholder	Replace With
NUM	A whole number, Number variable, or numeric expression

The LIMIT instruction limits the number of times a dialog can be repeated. It is placed in the script of the dialog that must be limited to a specific number of repetitions.

When setting repeat limits, assign a number value or numeric expression. For example, the dialog, *Daily Schedule Information*, gathers information about the scheduled activities for each day of the work week, so it needs to be limited to five repetitions. The following would be placed in the dialog's script:

LIMIT 5

You can let the **user** change the limit each time a **document** or form is assembled by replacing the NUM placeholder with a Number variable. Make sure the Number variable gets answered before the REPEAT instruction is processed.

To control the number of viewable rows in a spreadsheet (but still allow users to enter as many answers as they need), enter the number in the **Rows to display field**. (Make sure **Spreadsheet** is selected as the **Style**.)

OMIT VAR

Placeholder	Replace With
VAR	A variable used in an answer source

The OMIT VAR instruction, which you use in a dialog script, keeps variables from appearing in the **Edit Answer Source** dialog box of an answer source.

Answer sources are associated with specific dialogs in a template. When users view the dialog during the interview, they can click a special button, which displays a list of answers that were entered during previous interview sessions. They can either select an existing set of answers or add a new set to the list.

After creating an answer source for one dialog, you can use it with other dialogs, including dialogs in other templates. When using an answer source with multiple dialogs, each variable must be represented in both the answer source file and in each dialog. If a variable that is referenced in the answer source file isn't included in the dialog (or vice-versa), answers in the answer source will get "mixed together" whenever you add, edit, or delete records.

In some situations though, it isn't always practical or relevant to show the user every variable—either in the dialog or in the **Select From Answer Source** dialog box. To accommodate this, you can use the HIDE, CONCEAL, and OMIT instructions to manipulate these variables in both the dialog and the answer source. Specifically, HIDE keeps a variable from appearing in the dialog, while CONCEAL keeps it from appearing in the answer source altogether, but still allows you to use it in the dialog. Often, you must use a combination of these instructions to achieve your desired result.

For example, say you have a dialog that shows information about a client, including how much a client owes in payments to the firm. Because information about amounts owed changes, it would not make sense to include it in the answer source. To keep it from appearing in the answer source—both in the **Select From Answer Source** dialog and in the **Edit Answer Source** dialog—you would use the OMIT instruction, like this:

OMIT Amount Owed

The variable would be asked on the dialog, however, so the user could answer it.

See Suggest an Answer Source for Dialogs for more information.

PLAY "MACRO"

Placeholder	Replace With
"MACRO"	A word processor macro that performs a certain function in your template

The PLAY "MACRO" instruction plays a word processor macro after the document is assembled and either sent to the word processor, printed, or saved.

Where you store the macro depends on which word processor you are using:

- For WordPerfect users, the macro can be stored anywhere. The PLAY instruction must include the file name of the macro, and if the macro is stored anywhere other than the default macro folder, the instruction must include a full path to the file as well.
- For Word DOT users, the macro must be stored in the template itself, in *Normal.dot*, or in any global template that is automatically loaded when you start Word.
- For Word RTF or DOCX users, the macro must be stored either in *Normal.dot*, in any global template that is automatically loaded when you start Word, or in a Word template you associate with the template through Component Manager. (See Specify a Word Template for Storing Post-Assembly Macros for details.)

Macros can be helpful in many situations, such as making sure the format of an inserted clause matches the rest of the document. For example, the text in the *Authority Clause* document may be formatted differently than the text in the main document. You could create a macro that can adjust the formatting so that it's uniform:

«INSERT "Authority Clause.docx"»

«PLAY "Standard_Format"»

To insert a PLAY instruction in a template

- 1. At the template, position the cursor in the template where you want the PLAY instruction.
- 2. If you are using Microsoft Word:
 - Click the **HotDocs** menu in the HotDocs toolbar and choose **Other Field** from the list of options. The **Other Field** dialog box appears.
 - Click the Field type drop-down button and choose PLAY.
 - In the Macro name field, enter the name of the macro you want to run.
 - Click **OK**. The instruction is inserted in the template.
- 3. If you are using WordPerfect:
 - Copy an existing variable or instruction field in the template.
 - Replace text between the chevrons (« ») with the PLAY instruction.

See Specify a Word Template for Storing Post-Assembly Macros for more information.

PLAY instructions are executed when you create an actual document from the assembly. This includes sending the document to the word processor, saving the document, or printing a copy of the document. If there are multiple instructions, they are processed in the **order** they are encountered.

QUIT

The QUIT instruction allows you to close the variable without losing the work you have done. Normally, HotDocs will not save an invalid computation. The only way to exit an invalid computation is to click **Cancel**, which erases the script. The instruction is usually placed at the beginning of an unfinished or invalid computation.

For example, perhaps you aren't sure about the specific variable names that need to be included because the variables have not yet been created. Normally, HotDocs won't allow an unfinished script to be saved. The QUIT instruction, though, makes this possible:

QUIT IF Client History = "____" SET ____ TO "Returning" ELSE SET ____ TO "New" END IF Another useful place to include a QUIT instruction is at the end of a computation, which allows you to enter "developer comments" about the computation script. You can also apply a comment block to the section of the script you don't want processed. To do this, select that section of the script and click the **Comment Block** button. (To uncomment the script, click the **Developer Block** button.)

If you use a QUIT instruction in a script, it will cause all the scripting after the instruction to lose its syntax-aware formatting. To restore this formatting once you remove the QUIT instruction, click the **Auto Format** button.

REPEAT DATABASE; END REPEAT

Placeholder	Replace With
DATABASE	A database component

A REPEAT instruction gathers multiple database records and merges them into a document.

For example, the following script will bring up a record selection screen during the interview and allow you to choose multiple records to be inserted into the assembled document:

«REPEAT Client Data»

«END REPEAT»

Each REPEAT instruction must also include an END REPEAT instruction, which tells HotDocs to stop repeating the variables within the instruction.

REPEAT DATABASE instructions cannot be nested.

REPEAT DIALOG; END REPEAT

Placeholder Replace With

DIALOG A dialog with a repeat style specified

A REPEAT instruction gathers lists of answers and merges them into a document.

For example, the following script gathers a list of editors and inserts them into the assembled document:

```
«REPEAT Editor Information»
Editor: «Editor First Name» «Editor Last Name»
«END REPEAT»
```

Each REPEAT instruction must also include an END REPEAT instruction, which tells HotDocs to stop repeating the variables within the instruction.

This next computation script actually contains two repeated dialogs—one nested in another. It produces a list of editors as well as the titles and authors he or she is currently working with:

```
""
REPEAT Editor Information
    RESULT + "Editor: " + Editor First Name + " " + Editor Last Name + "
"
    REPEAT Book Information
        RESULT + "Book Title: " + Book Title + "
" + "Author: " + Author First Name + " " + Author Last Name + "
"
END REPEAT
END REPEAT
```

You can create up to three levels of sublists by nesting REPEAT instructions. (See Introduction: Create Lists Within a List.)

For more information about creating lists of answers, see Introduction: Include Lists in Your Documents.

Repeats can not be nested more than four levels deep.

While editing complicated sequences, you can jump from the END REPEAT instruction to its associated REPEAT instruction, or vice versa, by placing the cursor within the REPEAT or END REPEAT chevrons, then clicking the **Sequences** button.

REQUIRE ALL; REQUIRE VAR



The REQUIRE instruction requires users to answer questions in a dialog before they can advance to the next dialog in the interview.

REQUIRE instructions ensure that important information is not left out of the assembled document. For example, a certain document requires the user to enter the date when a legal filing was first made. Later in the template, this date is used to calculate a deadline for subsequent filings. Many users, however, may not take the time to look up the initial filing date, which creates problems for calculating the deadline. Using the REQUIRE instruction in a dialog script, as shown in this example script, can help resolve this problem:

REQUIRE Initial Filing Date

When a dialog script contains a REQUIRE instruction, a red mark appears on the dialog icon in the interview outline. The question in the dialog is also marked. This mark remains until the required variable is answered. If a user tries to move to another dialog, HotDocs displays an error message, then moves the cursor to the first required answer field in the dialog. Users cannot advance to the next dialog without first providing the required answers.

It may be helpful to include text in the dialog that provides users with information about which fields are mandatory and why. See Add Text to Your Dialogs for details.

You can customize the marks used in the dialog by specifying your options at HotDocs Options. See Customize the Look of the Dialog Pane.

Although current and past versions of HotDocs Developer allow the user to click Finish on an Interview Template than contains unanswered required variables or ERRORTEXT variables which have an incorrect answer in them, this behavior will be changed in the future so your templates should not rely on it.

SET VAR TO VALUE

<u>Placeholder</u>	Replace With
VAR	Any variable
VALUE	A value appropriate for the associated variable

For a more detailed explanation of setting variables to values, see Automatically Assign Answers to a Variable.

This instruction lets you specify a given value for a variable's answer automatically, rather than allow the user to specify an answer. With the SET instruction, you can transfer names and other values from one variable to another.

For example, if the plaintiff is the same as the client, you can use a SET instruction to automatically enter the plaintiff's name as the client's, saving the user from typing the name a second time:

IF Is Plaintiff Also Client SET Client Name TO Plaintiff Name FND TF

The SET Instruction can also be used to set custom error messages in interview dialogs. Used with the ERRORTEXT expression in a dialog script, SET can be used to create conditional error messages to help minimise incorrect answers. To see examples of this method see ERRORTEXT.

Because HotDocs repeatedly processes SET instructions during the course of an interview, you must not let the user specify a different answer for a variable whose value is being SET. When HotDocs updates the interview, the user's answer will be replaced with the value from the SET instruction. To suggest an answer for the user and allow them to change it, use the DEFAULT instruction. (See Differences Between SET and DEFAULT Instructions for an explanation.)

If you are grouping two or more child (inserted) dialogs in a parent dialog, you can SET the child dialog's status to TRUE. This forces the contents of the child dialog to automatically appear in the interview outline. (See Group Child Dialogs in a Parent Dialog for details.)

To set two or more options for a Multiple Choice variable, separate each option with a vertical bar (for example, SET MC Variable TO "Option1|Option2|Option3")

Placeholder	Replace With
EXPRESSION	An expression that results in true or false
	A True/False expression can be as complicated as it needs to be, but it must result in either true or false. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a true or false value—it is not a True/False expression. But the expression

WHILE EXPRESSION; END WHILE

YEARS FROM(Child's Birth Date, TODAY)> 17 can only result in true or false. It is a True/False expression.

The WHILE EXPRESSION instruction allows you to repeatedly process (or loop through) an answer or set of answers until a certain condition is met, such as a certain answer is found or a limit is reached.

Before using the WHILE instruction, you should understand the following:

- The WHILE instruction must be used in a computation or dialog script—it cannot be inserted directly into a template.
- When using the WHILE instruction inside of a REPEAT instruction, the WHILE instruction will not
 affect the underlying COUNTER variable associated with the REPEAT instruction. If you need to
 count something within a WHILE loop, you must create your own temporary counter. Additionally,
 to access repeated variables in a WHILE loop, you must use explicit indexing.
- Unless the instructions inside the WHILE loop include an instruction that increments the temporary counter, the WHILE instruction will repeat until the Maximum WHILE iterations limit is reached. You can specify this property at the Component File Properties dialog box. (See Change Component File Properties.) To avoid problems like this, make sure you increment the temporary counter.

In the following example, you want to create a list of signers in a will. Since the signers may include both beneficiaries and fiduciaries, you want to merge both lists into one. Because some fiduciaries may also be beneficiaries, you will want to remove any duplicate names. To loop through the list of fiduciaries, you will use the WHILE instruction:

SET Signer Count TO 0
REPEAT Beneficiary Information
INCREMENT Signer Count
SET Signer Name[Signer Count] TO Beneficiary Name
END REPEAT
REPEAT
REPEAT Fiduciary Information
SET Lookup TO 1
WHILE Lookup <= Signer Count AND Fiduciary Name != Signer Name[Lookup]
INCREMENT Lookup
END WHILE
IF Lookup > Signer Count
INCREMENT Signer Count

SET Signer Name[Signer Count] TO Fiduciary Name

END IF

END REPEAT

In the first part of this script, the *Beneficiary Information* dialog is repeated, and as answers are entered, their values are set to be used for *Signer Name* (which is the variable that will be repeated to insert all the names of the signers). Then, in the second part of the script, as the *Fiduciary Information* dialog is repeated, HotDocs uses the WHILE expression to test whether the name of the fiduciary is the same as any of the beneficiary names. If it is not, it will likewise be added to the *Signer Information* dialog. (When you insert the REPEAT instruction for the *Signer Information* dialog in the template, clear the **Ask Automatically** option at the **Dialog Editor** (**Options** tab). See Control Whether Dialogs are Asked Automatically for details.)

In the next example, you need to remove unwanted space characters from a user's account number. Here, the WHILE instruction is used to repeat an answer, character by character, so that HotDocs can check to see if there are space characters in the answer. If there are, HotDocs removes them and rewrites the answer.

SET Count Index TO 1
WHILE Count Index <= LENGTH(Account Number)
 IF MID(Account Number, Count Index, 1) = " "
 SET Account Number TO FIRST(Account Number, Count Index -1) +
 LAST(Account Number, LENGTH(Account Number) - Count Index)
 ELSE
 INCREMENT Count Index</pre>

END IF

END WHILE

This script uses a temporary counter (*Count Index*) to keep track of which character in the answer HotDocs is looking at. Any time the answer is repeated and HotDocs finds a space character, it removes it by concatenating the characters before and after the space character. HotDocs then makes sure that the new character it is now examining isn't a space character either. If it is not, HotDocs increments the temporary counter, moves to the next character, and repeats this process.

Using Expression Models

Full List of Expression Models

Expression	Description
Enter a Date	You can use this expression to format your dates correctly as you write computations.
<u>Enter a Number</u>	You can use the Enter a Number expression to format numbers correctly as you write computations. Numbers in computations and expressions must be in numeric form and cannot contain commas. If you enter a number that contains a comma, it will be removed from the value. Decimals must have one digit to the left of the decimal point, even if it is only a zero.
Enter some Text	You can use the Enter some Text expression to format a text string correctly as you write computations. When used in computations and expressions, text strings must be inside quotation marks. This expression adds those quotation marks to the text string.
Enter True or False	You can use this expression to enter a TRUE or FALSE value in a computation or expression script. True/False values must use uppercase letters.
<u>ABSOLUTE VALUE(NUM)</u>	Using the ABSOLUTE VALUE expression, you can find the absolute value of a given number. You can calculate a negative number, but have it appear as a positive number.
<u>AGE(DATE)</u>	The AGE(DATE) expression produces an age, in years, by calculating the number of years between the current date (as determined by your computer's system clock) and a date you provide in the computation script.
<u>ANSWERED(DIALOG)</u>	HotDocs can determine whether a dialog has been answered using the ANSWERED expression. Even if only one variable in the dialog is answered, the expression returns a value of true.
<u>ANSWERED(VAR)</u>	You can use the ANSWERED expression to determine whether a HotDocs variable has been assigned a value. If so, the expression receives the value of <i>true</i> .
<u>CEILING(NUM)</u>	You can use the CEILING function to find the smallest integer that isn't smaller than the Number variable.
<u>COUNT(DIALOG)</u>	You can find out how many sets of answers a user provides for a repeated dialog. A repeated dialog is any dialog used in a REPEAT instruction. This expression produces a number, based on each answered dialog.
<u>COUNT(</u> <u>MULT CHOICE VAR)</u>	This expression counts how many options a user chooses when answering a Multiple Choice variable. The result it produces is a number.
COUNTER	You can use the COUNTER expression to keep track of the current number of repetitions of a repeated dialog. Each time a user clicks the

	Next button at a repeated dialog and provides additional information, the value of COUNTER increases.
<u>DATE - NUM DAYS</u>	You can subtract any number of days from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
<u>DATE - NUM MONTHS</u>	You can subtract a certain number of months from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
<u>DATE - NUM YEARS</u>	You can subtract a certain number of years from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
<u>DATE + NUM DAYS</u>	You can add any number of days to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
DATE + NUM MONTHS	You can add any number of months to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
DATE + NUM YEARS	You can add a certain number of years to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
<u>DATE OF(DAY, MONTH,</u> <u>YEAR)</u>	This expression finds a date value based on day, month, and year values.
DAY OF(DATE)	This expression returns the day portion (1 to 31) of a given date.
DAY OF WEEK(DATE)	This expression determines on which day of the week a specific date falls and converts that value to an integer.
<u>DAYS FROM(START,</u> <u>FINISH)</u>	This expression allows you to find the number of days between two dates.
<u>ERRORTEXT</u>	This instruction allows you to create an error message that can appear during the interview.
EXPONENTIAL(NUM)	An EXPONENTIAL function is a function that quickly accelerates, where the Number variable is the exponent of <i>e</i> (roughly 2.71828), and it can be used in calculating exponential growth.
FIRST(TEXT, NUM)	Using this expression, you can return any number of characters starting with the first character in an answer value.
FLOOR(NUM)	You can use the FLOOR function to find the largest integer that isn't bigger than the Number variable entered.

FORMAT(VALUE, "EXAMPLE")	Sometimes you may need to add a date, number, or true/false value to a text value. You can do this by formatting the date, number, or true/false value as text.
<u>FUTURE VALUE (RATE,</u> <u>TERM, PAYMENT,</u> <u>PRESENT VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.
	If you know the rate, term, payment, present value and type of a savings scheme or loan you can use the FUTURE VALUE function to work out how much the final amount will be after interest.
INTEGER(TEXT)	Sometimes you may have a text value that contains number characters, as in the case of a time of day value. The INTEGER expression allows you to convert those number characters into numeric values so you can perform calculations or compare them with other values.
<u>LAST(TEXT, NUM)</u>	The LAST expression finds and returns a certain number of characters from the end of a text string.
<u>LENGTH(TEXT)</u>	The LENGTH expression counts the number of characters—including spaces and punctuation—in a text value, such as a Text variable.
<u>LOGARITHM(NUM)</u>	The LOGARITHM function will find the common logarithm of a Number variable. You can use it to find the exponent of 10 that would be needed to make the Number variable entered.
<u>MAX(NUM, NUM)</u>	The MAX expression compares two number values and returns the greater of the two.
<u>MID(TEXT, NUM, NUM)</u>	Like the FIRST and LAST expressions, this expression extracts a specified number of characters from within a text string.
<u>MIN(NUM, NUM)</u>	The MIN expression compares two number values and returns the lesser of the two.
MONTH OF(DATE)	This expression returns the month portion of a given date.
<u>MONTHS FROM(DATE,</u> <u>DATE)</u>	The MONTHS FROM expression calculates the number of months between two given dates.
MULT CHOICE=TEXT; MULT CHOICE!=TEXT	The MULT_CHOICE = TEXT expression returns true when the user chooses a Multiple Choice option that is equal to (=) a given text value. If it is not equal (!=), the expression returns false. The MULT_CHOICE != TEXT expression functions in the opposite way—testing instead to see if an answer is not equal to (!=) a given text value.
<u>NATURAL LOGARITHM(</u> <u>NUM)</u>	You can use the NATURAL LOGARITHM function to find the exponent of <i>e</i> (roughly 2.71828) that would be needed to make the Number variable entered.

NOT TRUE FALSE	You can use the NOT TRUE_FALSE expression to find out if a True/False variable is false.
<u>OTHER(</u> <u>MULT CHOICE VAR)</u>	This expression determines whether the user has chosen the <i>Other</i> option of a Multiple Choice variable and, if so, returns the text entered in the <i>Other</i> field. It can also be used to test whether the user has selected the <i>None of the Above</i> option.
<u>PAYMENT (RATE, TERM,</u> <u>PRESENT VALUE, FUTURE</u> <u>VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.
	If you know the rate, term, present value, future value and type of a savings scheme or loan you can use the PAYMENT function to work out how much money needs to be paid in each payment period to reach the full amount
POSITION(TEXT, TEXT)	The POSITION expression finds the position of a certain character or character string in a given text value. It is useful if you need to find a character you know will be in an answer but are not sure where it will appear. It returns a number value, which represents the first character.
<u>POWER(NUM, NUM)</u>	The POWER expression generates a numeric value, based on a given exponent.
<u>PRESENT VALUE (RATE,</u> <u>TERM, PAYMENT, FUTURE</u> <u>VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.
	If you know the rate, term, payment, future value and type of a savings scheme or loan you can use the PRESENT VALUE function to work out how much the investment is worth currently (i.e. how much money you would need to invest over the same amount of payment periods to equal the return)
<u>RATE (TERM, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.
	If you know the term, payment, present value, future value and type of a savings scheme or loan you can use the RATE function to work out what the interest rate is per payment period.
<u>REMAINDER(NUM, NUM</u>)	The REMAINDER expression returns the remainder of a division. If the denominator is a zero, HotDocs generates a divide by zero error.

<u>REPLACE(TEXT, TEXT,</u> <u>TEXT, NUM)</u>	This expression lets you search a string of text for a given character string and replace the results with new text.
<u>RESULT</u>	As you write computations, you often need HotDocs to acknowledge what the result would be at that point in the script. You can update this answer by using the RESULT expression.
ROUND(NUM, NUM)	You can round a number value to a specified number of places.
SELECTION(MULT CHOICE VAR, NUM)	This expression lets you retrieve individual options (answers) selected in a Multiple Choice variable. It returns a text value that corresponds to the defined answer (as designated by the NUM placeholder).
<u>SPACE(TEXT, TEXT)</u>	This expression tests whether the variable is answered. If it is, it merges the answer, followed by a space character. If the variable is unanswered, it merges nothing ("").
<u>SQUARE ROOT(NUM)</u>	Finding the square root of a number means finding an answer that, when multiplied by itself, gives the original number. You can use the SQUARE ROOT function to find the square root of a Number variable.
<u>STRIP(TEXT, TEXT,</u> <u>TRUE FALSE, TRUE FALSE</u>)	This expression removes a specified character or characters from the beginning or end of a text answer. By default, HotDocs removes the characters from both the beginning and the end of the text. If you want to specify just one or the other, you must use the <i>TRUE_FALSE</i> parameters.
<u>SUM(</u> COMPUTATION VAR)	Using the SUM expression, you can add repeated number values.
<u>SUM(NUM VAR)</u>	Using the SUM expression, you can add repeated number values.
<u>TERM (RATE, PAYMENT,</u> <u>PRESENT VALUE, FUTURE</u> <u>VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.
	If you know the rate, payment, present value, future value and type of a savings scheme or loan you can use the TERM function to work out how many payment periods are required to reach the full amount.
TEXT CONTAINS TEXT	The TEXT CONTAINS TEXT expression determines whether the first text value contains the same text as the second value. If it does, it returns the value of <i>true</i> .
TEXT ENDS WITH TEXT	The TEXT ENDS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value ends with the second TEXT value.

TEXT STARTS WITH TEXT	The TEXT STARTS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value begins with the second TEXT value.
TODAY	This expression returns the current date, according to your computer's system clock.
TRIM(TEXT)	You can use the TRIM function to remove any white space characters from the beginning and end of a Text variable.
TRUNCATE(NUM, NUM)	You can truncate a decimal number a specified number of places after a decimal point.
<u>UNANSWERED</u>	This expression removes an assigned value from a variable. It is used most often with the SET VAR TO VALUE instruction.
UNION(MULT_CHOICE, MULT_CHOICE)	This expression creates a single list of all unique options (answers) that have been selected across two Multiple Choice variables.
VALUE(VAR, EXPRESSION)	This expression returns a default value for the variable type if the variable is unanswered. If the variable is answered, the value is the answer the user specifies
<u>YEAR OF(DATE)</u>	You can use this expression model to find the year portion of a given date.
YEARS FROM(DATE, DATE)	This expression calculates the number of years between two given dates.
ZERO(NUM VAR)	This expression returns the value of <i>zero</i> only if a Number variable is unanswered. If the Number variable is answered, the value is the answer the user specifies.

Enter a Date

You can use this expression to format your dates correctly as you write computations.

To use the expression, drag the **Enter a Date** expression into the **Script** or **Expression** field. The **Enter a Date** dialog appears where you can type a date into the field, or you can use the pop-up calendar to select a date. When you click OK, HotDocs inserts the date at the cursor position in the correct format, for example, *3 JUN 1990*.

Enter a Number

You can use the **Enter a Numbe**r expression to format numbers correctly as you write computations. Numbers in computations and expressions must be in numeric form and cannot contain commas. If you enter a number that contains a comma, it will be removed from the value. Decimals must have one digit to the left of the decimal point, even if it is only a zero.

To use the expression, drag the **Enter a Number** expression into the **Script** or **Expression** field. The **Enter a Number** dialog box appears. Enter a number and click **OK**. The correctly formatted number is inserted at the cursor position.

Enter some Text

You can use the **Enter some Text** expression to format a text string correctly as you write computations. When used in computations and expressions, text strings must be inside quotation marks. This expression adds those quotation marks to the text string.

To use the expression, drag the **Enter some Text** expression into the **Script** or **Expression** field. The **Enter some Text** dialog box appears. Enter some text and click **OK**. HotDocs inserts that text—in quotation marks—at the cursor position.

Enter True or False

You can use this expression to enter a TRUE or FALSE value in a computation or expression script. True/False values must use uppercase letters.

To use the expression, drag the **Enter True or False** expression into the **Script** or **Expression** field. The **Enter True or False** dialog box appears. Click either **True** or **False**. The keyword you choose is inserted in all capital letters at the cursor position.

ABSOLUTE VALUE(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value for which you want the absolute value returned. Can be a Number variable or a fixed number value.

Returns a **Number** value

Using the ABSOLUTE VALUE expression, you can find the absolute value of a given number. You can calculate a negative number, but have it appear as a positive number.

For example, you may need to send a notice to a client about an account balance. Not knowing whether it will be a positive or negative balance, you would create the computation *Absolute Value of Final Balance*, which would return a positive expression, regardless. The computation script is:

ABSOLUTE VALUE(Final Balance)

Once you have created the computation, you can use an IF/ELSE expression to merge the desired text with the correct value in the document:

«IF Final Balance < 0»
Your account is \$«Absolute Value of Final Balance» overdrawn.
«ELSE»
You have \$«Absolute Value of Final Balance» in your account.
«END IF»</pre>

AGE(DATE)

Placeholder	Tooltip	Replace With
DATE	d: Date	A date value, which you want to check against the current date. This can be a Date variable or a fixed date value.

Returns a **Number** value

The AGE(DATE) expression produces an age, in years, by calculating the number of years between the current date (as determined by your computer's system clock) and a date you provide in the computation script.

For example, the following script determines the age of the user based on his or her birth date:

AGE(Birth Date)

In the following conditional script, HotDocs determines whether the client is under the age of 18. If so, the *Parent-Guardian Information* dialog is asked. If the client is over the age of 18, no dialogs are asked.

IF AGE(Birth Date) < 18

ASK Parent-Guardian Information

END IF

ANSWERED(DIALOG)

Placeholder	Tooltip	Replace With
DIALOG	d: Dialog	A dialog name

Returns a **True/False** value

HotDocs can determine whether a dialog has been answered using the ANSWERED expression. Even if only one variable in the dialog is answered, the expression returns a value of true.

Let's suppose you have a dialog that gathers information about a decedent (*Decedent Information*). From within this dialog, a user could open an inserted dialog that asks questions about the decedent's assets (*Decedent's Assets*). Later in the template, you could ask additional questions based on whether these dialogs have been answered:

ANSWERED(VAR)

Placeholder	Tooltip	Replace With
VAR	v: Variable	A variable

Returns a **True/False** value

You can use the ANSWERED expression to determine whether a HotDocs variable has been assigned a value. If so, the expression receives the value of *true*.

If you are testing whether a user has ANSWERED a variable, you must make sure the variable is presented to the user using a custom dialog. Using the ANSWERED(VAR) expression alone will not automatically force HotDocs to display the variable for the user.

For example, you may place a variable for the second line of an address (*Client Address 2*) in a custom dialog; however, not all users will provide information for that variable. In the template text, you can surround *Client Address 2* with an IF expression that merges that variable into the document—only if the user answers it:

«Client Name»
«Client Address 1»
«IF ANSWERED(Client Address 2)»
«Client Address 2»
«END IF»
«Client City», «Client State» «Client ZIP»

In this script, the ANSWERED expression is used with an IF expression to insert or remove *Client Address 2*, based on whether the user has provided that information. (Without it, the default unanswered text, ****Client Address 2****, would be inserted in the assembled document.)

Even if a user chooses not to answer a variable that has been tested using the ANSWERED expression, HotDocs will still warn that it is unanswered. If you don't want a warning to appear, clear **Warn when unanswered** at the **Advanced** tab of the Variable Editor. (Control How HotDocs Processes a Variable.)

CEILING(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

You can use the CEILING function to find the smallest integer that isn't smaller than the Number variable, for example:

CEILING(84.2) = 85

COUNT(DIALOG)

Placeholder	Tooltip	Replace With
DIALOG	dlg: Dialog	A dialog name

Returns a **Number** value

You can find out how many sets of answers a user provides for a repeated dialog. A repeated dialog is any dialog used in a REPEAT instruction. This expression produces a number, based on each answered dialog.

The following example determines if there is more than one fiduciary. If there is, certain prefixes and plural abbreviations are added to the fiduciary title so the paragraph is structured correctly.

I appoint «REPEAT Fiduciary Information:a, b, and c»«Fiduciary Name:LIKE THIS»«END REPEAT» as «IF COUNT(Fiduciary Information) > 1»Co-«Fiduciary Title»s«ELSE»«Fiduciary Title»«END IF».

This expression uses the COUNT instruction to determine if *Fiduciary Information* is answered more than once. If it is, the *Co*- prefix is inserted before *Fiduciary Title*, and the plural *s* is inserted at the end of the variable. For example, in the assembled document, the answer may be inserted as *Co-Executors* or *Co-Personal Representatives*.

The difference between COUNT and COUNTER is that COUNT counts the number of repetitions in a list, while COUNTER gives you the number of the current repetition.

COUNT(MULT_CHOICE_VAR)

Placeholder	Tooltip	Replace With
MULT_CHOICE_VAR	m: Multiple Choice Variable	A Multiple Choice variable with the Select option set to All That Apply

Returns a **Number** value

This expression counts how many options a user chooses when answering a Multiple Choice variable. The result it produces is a number.

For example, suppose the user wants to generate a list of cities in which an author plans to make appearances. There might be one—or many—depending on the schedule. HotDocs can merge the correct term—*city* or *cities*—into the document once it knows how many cities were selected from the Multiple Choice variable:

```
IF COUNT( Publicity Tour City ) = 1
    "city"
ELSE IF COUNT( Publicity Tour City ) > 1
    "cities"
END IF
```

COUNTER

Returns a **Number** value

You can use the COUNTER expression to keep track of the current number of repetitions of a repeated dialog. Each time a user clicks the **Next** button at a repeated dialog and provides additional information, the value of COUNTER increases.

For example, a user may want to create a word processor table that contains a numbered list of clients:

«REPEAT Client Information»
«COUNTER». «Client Name»
«END REPEAT»

In the example above, *Client Information* repeats the *Client Name* variable. Each time a user enters a different client, COUNTER is incremented and merged into the assembled document. For example:

- 1. John TeNgaio
- 2. Erica Nees
- 3. Lisa Alvey
- 4. Jonathan Rainwater

HotDocs also uses COUNTER as a way to compare two incrementing number values. For example, perhaps you want to list the last child named in a repeated dialog:

```
REPEAT Children Information
ASCEND Child Birth Date
IF COUNTER = COUNT( Children Information )
RESULT + "The youngest child is " + Child Name
END IF
END REPEAT
```

In this computation script, HotDocs first sets the value of the computation to nothing. It then processes the REPEAT instruction, sorting the children based on their birth dates. It uses COUNTER to determine when the last answer in the dialog is given (by comparing it to the COUNT of the dialog), and then merges the name of the youngest child in the list into the document.

The difference between COUNT and COUNTER is that COUNT counts the number of repetitions in a list, while COUNTER gives you the number of the current repetition.

DATE - NUM DAYS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number value

Returns a **Date** value

You can subtract any number of days from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation calculates the date the books need to arrive at the warehouse before they can be shipped:

Shipping Date - 14 DAYS
DATE - NUM MONTHS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number value

Returns a **Date** value

You can subtract a certain number of months from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

In the following example, HotDocs subtracts four months from the *Shipping Date* and inserts the new date:

Shipping Date - 4 MONTHS

DATE - NUM YEARS

Placeholder	Replace With	
DATE	A date value, such as a Date variable	
NUM	A number value	

Returns a **Date** value

You can subtract a certain number of years from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation subtracts two years from Marriage Date:

Marriage Date - 2 YEARS

DATE + NUM DAYS

Placeholder Replace With

DATEA date value, such as a Date variableNUMA number value

Returns a **Date** value

You can add any number of days to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation script adds 90 days to the Date variable, *Purchase Date*:

Purchase Date + 90 DAYS

DATE + NUM MONTHS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number value

Returns a **Date** value

You can add any number of months to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation determines what the date will be six months from the date the document is assembled:

TODAY + 6 MONTHS

DATE + NUM YEARS

Placeholder	Replace With	
DATE	A date value, such as a Date variable	
NUM	A number value	

Returns a **Date** value

You can add a certain number of years to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

In this script, HotDocs adds 30 years to the date the loan originated:

Loan Origination Date + 30 YEARS

DATE OF(NUM, NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	day: Number	A number value representing the day of the month (from 1 to 31)
NUM	month: Number	A number value representing the month (from 1 to 12)
NUM	year: Number	A number value representing the year (should be four digits)

Returns a **Date** value

This expression finds a date value based on day, month, and year values.

You can use this expression to compare a date the user gives with another date, such as a cut-off date for when an employee had to be hired to qualify for a yearly bonus:

IF Hire Date <= DATE OF (15, 8, YEAR OF(TODAY))</pre>

Employee Name + " qualifies for the annual bonus."

END IF

HotDocs uses an IF instruction to compare *Hire Date* with *August 15* of the current year. If the comparison returns a *true* value, the employee qualifies for the bonus.

Suppose, in this next example, a new employee qualifies for a benefits package on the first day of the second month of employment—regardless of what day the employee was hired during the first month of employment. You can calculate that date with the DATE OF expression:

DATE OF (1, MONTH OF(Hire Date + 1 MONTHS), YEAR OF(Hire Date + 1 MONTHS))

The first parameter in the expression, 1, tells HotDocs to specify the first day of the month. The second parameter identifies the month of the hire date and adds one month. The third parameter determines the year of the hire date (plus one month).

DAY OF(DATE)

Placeholder	Tooltip	Replace With
DATE	d: Date	A date value

Returns a **Number** value

This expression returns the day portion (1 to 31) of a given date.

The following computation is used to determine when a new employee can begin accruing vacation days. If the employee is hired on the first day of the month, he or she immediately begins accruing time off. Otherwise, he or she begins accruing at the beginning of the next month:

```
IF DAY OF ( Hire Date ) = 1
SET Start Accruing Date TO Hire Date
ELSE
SET Start Accruing Date TO DATE OF( 1, MONTH OF( Hire Date + 1 MONTHS ),
YEAR OF( Hire Date + 1 MONTHS ) )
```

END IF

DAY OF WEEK(DATE)



Returns a **Number** value

This expression determines on which day of the week a specific date falls and converts that value to an integer.

HotDocs Developer Help File These integers are as follows: Sunday = 1 Monday = 2 Tuesday = 3 Wednesday = 4 Thursday = 5 Friday = 6 Saturday = 7

For example, perhaps you want to determine whether a payment due date falls on a Saturday or Sunday. If it does, HotDocs moves the payment due date to the following Monday. The following script shows how this works:

```
IF DAY OF WEEK( Payment Date ) = 7
Payment Date + 2 DAYS
ELSE IF DAY OF WEEK( Payment Date ) = 1
Payment Date + 1 DAYS
ELSE
Payment Date
END IF
```

DAYS FROM(DATE, DATE)

Placeholder	Tooltip	Replace With
DATE	start: Date	A date value, such as a Date variable
DATE	finish: Date	A date value, such as a Date variable. (These can be in any order.)

Returns a **Number** value

This expression allows you to find the number of days between two dates.

In the following example, a buyer has 60 days to make a payment on an account balance. If the buyer has miss ed the payment deadline, HotDocs merges a warning into the document:

IF DAYS FROM(Purchase Date, TODAY) > 60

"Your account is past due."

ELSE

"Your account is current. Thank you."

END IF

This example uses an IF/ELSE IF expression to determine the text that must be inserted.

ERRORTEXT

Returns a **Text** value

ERRORTEXT can only be used in a dialog script.

Use the ERRORTEXT expression to create an error message that appears when the error condition you specify occurs during an interview. If a user provides information that triggers the error message, the user must correct the error before HotDocs enables access to the next dialog.

If you make the ERRORTEXT expression conditional, the error message appears both as a popup, and at the top of the dialog, accompanied by a red error symbol. If you do not make the ERRORTEXT expression conditional, as soon as the user clicks the dialog, the error message appears and displays continuously at the top of the dialog. In this case, there is no accompanying popup.

You can make the error conditional by adding an IF instruction to the script. You could, for example, warn users when they type in a forbidden answer.

Say you work for a delivery company that does not deliver to certain areas of the United States, you could use an ERRORTEXT expression to warn customers when they type the name of a state that is outside of your delivery area. Your dialog script might look something like this:

IF state = "Hawaii" Or state = "Alaska"

SET ERRORTEXT TO "Unfortunately we do not deliver to «state»"

END IF

If a customer types either "Hawaii" or "Alaska," the error message appears and they must change their answer before continuing with the interview.

HotDocs blocks the user from continuing to the next dialog, because HotDocs handles an ERRORTEXT expression like a required variable. You can use this feature to generate error messages if the user leaves a necessary question unanswered in an interview.

In a similar scenario to the example given above, suppose your delivery company needs to make sure that customers type the name of a country when filling out their address. Your script, to alert the user when they leave the country field blank, might look something like this:

IF !ANSWERED (Country)

SET ERRORTEXT TO "Country is a required field"

END IF

If a customer does not type the name of a country before trying to move to the next dialog, the error message appears. An Input Error icon also appears in the interview outline until the user provides an acceptable value for this variable.

In your ERRORTEXT message, it may be helpful to include information for users on how they can meet the conditions needed to complete the dialog and continue with the interview.

Although current and past versions of HotDocs Developer allow the user to click Finish on an Interview Template than contains unanswered required variables or ERRORTEXT variables which have an incorrect answer in them, this behavior will be changed in the future so your templates should not rely on it.

EXPONENTIAL(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

An EXPONENTIAL function is a function that quickly accelerates, where the Number variable is the exponent of *e* (roughly 2.71828), and it can be used in calculating exponential growth.

Example:

```
EXPONENTIAL( NUM ) = e NUM
```

EXPONENTIAL(3) = e^{3}

EXPONENTIAL(3) = 20.0855369231877

FIRST(TEXT, NUM)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable, from which the specified number of characters will be returned. This can be a fixed text value, inside quotation marks.
NUM	count: Number	A number value, such as a Number variable or a fixed number value. It specifies the number of characters you want returned.

Returns a **Text** value

Using this expression, you can return any number of characters starting with the first character in an answer value.

The following computation looks at the client's first, middle, and last names and returns only the first character from each of these variables. When merged together, these characters create the client's initials:

```
FIRST( Client First Name, 1 ) + FIRST( Client Middle Name, 1 ) + FIRST( Client
Last Name, 1 )
```

In the following example, the first four characters of a client's last name are merged with a case number to create a file number.

FIRST(Client Last Name, 4) + Case Number

FLOOR(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

You can use the FLOOR function to find the largest integer that isn't bigger than the Number variable entered, for example:

FLOOR(23.7) = 23

FORMAT(VALUE, "EXAMPLE")

Placeholder	Tooltip	Replace With
VALUE	v: Value	Any Number, Date, or True/False variable
"EXAMPLE"	example: Text	A format example (in quotation marks) you want used with the value. Must be in a format HotDocs can recognize.

Returns a **Text** value

Sometimes you may need to add a date, number, or true/false value to a text value. You can do this by formatting the date, number, or true/false value as text.

For example, perhaps you want to create a list of items with their associated monetary values. Because these two values are different in nature, they cannot be added together without first representing the number value as a text value:

"" REPEAT Purchase Information RESULT + Item Name + ", " + FORMAT(Item Amount, "\$9,999.00") + "

...

END REPEAT

In this script, HotDocs first sets the computation value to nothing. Then HotDocs repeats *Purchase information* and then places the answers for both *Item Name* and *Item Amount* (which is formatted to appear as a text value) in the same text string, separated by a comma. If *Purchase Information* is answered more than once, HotDocs manually inserts a hard return (as shown before the END REPEAT) to create a column of amounts.

FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Term	term: Number	A number variable representing the total number of payment periods.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Present Value	[present value: Number]	A number variable representing the current value of the future amount. Optional. Default is <i>0</i> .
(optional)		
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, term, payment, present value and type of a savings scheme or loan you can use the FUTURE VALUE function to work out how much the final amount will be after interest.

For example, if you have an annual rate of 7.5% over 3 years and you pay \$100 a month then you can work out the future value by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate) Term: 36 (amount of months in 3 years)

Payment: -100 Present Value: 0 Type: 0

Future Value = 4,023.1381682

This function works in much same way as the FV(rate, nper, pmt, pv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for FV.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- TERM (Rate, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)

INTEGER(TEXT)

Placeholder	Tooltip	Replace With
TEXT	t: Text	A text value, such as a Text variable

Returns a **Number** value

Sometimes you may have a text value that contains number characters, as in the case of a time of day value. The INTEGER expression allows you to convert those number characters into numeric values so you can perform calculations or compare them with other values.

INTEGER searches the beginning of a text string for number characters and converts those it finds to numeric values. When it encounters a non-number character (such as a letter or punctuation mark) it stops processing the instruction.

For example, if you tried to find the integer of the word *cat*, the INTEGER expression would return *0* (zero) since there are no number characters in *cat*. However, if you used INTEGER on the text value *12:30*, it would return the number value *12* since those characters are numbers. (As explained earlier, it stops processing when it reaches a punctuation mark, which in this case is a colon.)

One of the main uses for the INTEGER expression is to compare time values. In the following computation, HotDocs is attempting to determine if a given time value falls after 5:30 P.M. Because time values are text values, the Text variable, *Call Time*, must first be converted to an integer before it can be used in the comparison:

```
IF Call Time CONTAINS "p"
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
":" ), 2 ) ) /60 ) > 5.5
ELSE
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
":" ), 2 ) ) /60 ) > 17.5
END IF
```

In this script, *Call Time* is a Text variable with a 24-hour or 12-hour time pattern (99:99 or 99:99 A.M.). HotDocs first determines if *Call Time* is in the afternoon (P.M.). If it is, the script uses the INTEGER expression to convert all the digit characters up to the first non-digit character (the colon) into a numeric value. This number represents the hours portion of the total time. Using the MID expression to locate the two digit characters after the colon, it also converts these characters into an integer and divides the value by 60. This number represents the minutes portion of the total time. These two numbers are added together, and if the result is greater than 5.5 (the equivalent of 5:30), the result is *true*. If the result is not greater than 5.5, the result is *false*.

The second portion of the script (after the ELSE expression) performs the same functions on a nonafternoon time value—that is, one that is either in 24-hour format or in the morning (A.M.).

LAST(TEXT, NUM)	
-------------------	--

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable, from which the specified number of characters will be returned. This can be a fixed text value, inside quotation marks.
NUM	count: Number	A number value, such as a Number variable or a fixed number value. It specifies the number of characters you want returned.

Returns a **Text** value

The LAST expression finds and returns a certain number of characters from the end of a text string.

For example, the following text computation returns the last four digits of a Social Security number:

LAST(Social Security Number, 4)

In the next example, the user wants to make the answer to *Item Type* plural. Using the LAST expression, HotDocs checks to see if the last letter in the value is a specific letter. If so, HotDocs inserts the correct plural suffix.

```
IF LAST( Item Type, 1 ) = "s" OR LAST( Item Type, 1 ) = "z"
    Item Type + "es"
ELSE
    Item Type + "s"
END IF
```

This example script does not take into consideration words that end in "y" or "x" or any other letter that would cause yet a different result. It is only shown here in its most basic form to demonstrate how the LAST expression functions.

LENGTH(TEXT)

Placeholder	Tooltip	Replace With
ТЕХТ	t: Text	A text value, such as a Text variable. HotDocs counts the characters in this value and assigns a numeric value.

Returns a Number value

The LENGTH expression counts the number of characters—including spaces and punctuation—in a text value, such as a Text variable.

For example, let's say you want a user to add a descriptive paragraph about the novel he or she has just reviewed. If the description is brief, you'd like to keep it in the same paragraph as the lead-in sentence. However, if the user has much to say about the novel, you would like to start a new paragraph:

```
«IF LENGTH( Plot Description ) <= 150»
The novel's plot description is as follows: «Plot Description»
«ELSE»</pre>
```

The novel's plot description is as follows: «Plot Description» «END IF»

LOGARITHM(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

The LOGARITHM function will find the common logarithm of a Number variable. You can use it to find the exponent of 10 that would be needed to make the Number variable entered, for example:

NUM = 10 LOGARITHM(NUM)

 $1000 = 10^{-3}$

LOGARITHM (1000) = 3

MAX(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n1: Number	A number value, such as a Number variable
NUM	n2: Number	A number value, such as a Number variable (These values can be in any order.)

Returns a **Number** value

The MAX expression compares two number values and returns the greater of the two.

In this example, HotDocs returns the value of either the *Monthly Salary* or the *Monthly Expenses*, depending on which is the greater value:

```
MAX( Monthly Salary, Monthly Expenses )
```

MID(TEXT, NUM, NUM)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable, from which the specified number of characters will be returned. This can be a fixed text value, inside quotation marks.
NUM	start at: Number	A number value, such as a Number variable or a number you type. This number specifies where HotDocs starts "returning" characters.
NUM	count: Number	A number value, such as a Number variable or a number you type. It specifies the number of characters you want returned.

Returns a **Text** value

Like the FIRST and LAST expressions, this expression extracts a specified number of characters from within a text string.

For example, a form template may require that the text variable, *Telephone Number* (with the telephone number pattern), be split into three pre-formatted fields—the area code, the prefix, and the number. In the following example, three different computations would be scripted and inserted into the different fields as follows:

```
MID( Telephone Number, 2, 3 )
MID( Telephone Number, 7, 3 )
LAST( Telephone Number, 4 )
```

The first computation, which you would place in the area code field, tells HotDocs to include three characters, starting with the second character (this takes into consideration the opening parenthesis.)

The second computation (the prefix field) starts at the seventh character (again, taking into account the parentheses and space characters between the area code and the prefix), and inserts the next three characters.

The third expression, which uses the LAST expression, returns the last four digits of the phone number by counting backwards from the last character.

MIN(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n1: Number	A number value, such as a Number variable
NUM	n2: Number	A number value, such as a Number variable. (These values can be in any order.)

Returns a **Number** value

The MIN expression compares two number values and returns the lesser of the two.

In the following number computation, the expression compares the two values, *Shipping Costs* and *Labor Costs*, and returns the lesser of the two values:

```
MIN( Shipping Costs, Labor Costs )
```

MONTH OF(DATE)

Placeholder	Tooltip	Replace With
DATE	d: Date	A date value, such as a Date variable

Returns a **Number** value

This expression returns the month portion of a given date.

For example, in the following script, a new employee has temporary status until the end of his or her third month with the company. HotDocs uses the MONTH OF expression to specify which month that is:

DATE OF(1, MONTH OF (Hire Date + 3 MONTHS), YEAR OF (Hire Date + 3 MONTHS)) - 1 DAYS

In this example, HotDocs finds the first day of the fourth month of employment. HotDocs then subtracts one day to reveal the last day of the third month—either the 28th, 29th, 30th or 31st—depending on the month. It then inserts the new date into the document.

MONTHS FROM(DATE, DATE)

Placeholder	Tooltip	Replace With
DATE	start: Date	A date value, such as a Date variable
DATE	finish: Date	A date value, such as a Date variable. (These values can be in any order.)

Returns a **Number** value

The MONTHS FROM expression calculates the number of months between two given dates.

The following example finds the number of months between the judgment date and today—in months:

MONTHS FROM(Judgment Date, TODAY)

MULT_CHOICE=TEXT; MULT_CHOICE!=TEXT

Placeholder	Replace With
MULT_CHOICE	A Multiple Choice variable
TEXT	A text value that is either equal to (=) or not equal to (!=) one of the options in the given Multiple Choice variable, inside quotation marks.

Returns a **Number** value

The MULT_CHOICE = TEXT expression returns true when the user chooses a Multiple Choice option that is equal to (=) a given text value. If it is not equal (!=), the expression returns false. The MULT_CHOICE != TEXT expression functions in the opposite way—testing instead to see if an answer is not equal to (!=) a given text value.

In the following True/False expression, if the user chooses *Credit Card* as the payment method, HotDocs asks the user for the credit card information:

IF Method of Payment = "Credit Card"
 ASK Credit Card Information
END IF

In the next expression, if *Credit Card* is not chosen as a method of payment, HotDocs inserts a template which can gather alternate payment information about the user:

IF Method of Payment != "Credit Card"
 INSERT "Alt payment method.docx"
END IF

When writing this script, you can use the auto-complete functionality to access your list of Multiple Choice options. Specifically, press **Ctrl+Spacebar** (after you enter the operator) to display a list of the different Multiple Choice options. See **Use the Script Editor** for full details on using auto-complete as you write scripts.

NATURAL LOGARITHM(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

You can use the NATURAL LOGARITHM function to find the exponent of *e* (roughly 2.71828) that would be needed to make the Number variable entered, for example:

```
NUM = e NATURAL LOGARITHM( NUM )
1000 = e <sup>6.90775527898214</sup>
```

NATURAL LOGARITHM(1000) = 6.90775527898214

NOT TRUE_FALSE

Placeholder	Replace With
TRUE_FALSE	A true/false value, such as a variable or expression that results in <i>true</i> or <i>false</i> .

Returns a **Number** value

You can use the NOT TRUE_FALSE expression to find out if a True/False variable is false.

In the following script, HotDocs asks whether the user is a United States citizen. If the user is not, HotDocs asks the user for Visa information:

«IF NOT US Citizen» «ASK Visa Information» «END IF»

OTHER(MULT_CHOICE_VAR)

Placeholder	Tooltip	Replace With
MULT_CHOICE	m: Multiple Choice Variable	A Multiple Choice variable that has either the Other option specified, or the None of the Above option specified

Returns a Text value

This expression determines whether the user has chosen the *Other* option of a Multiple Choice variable and, if so, returns the text entered in the *Other* field. It can also be used to test whether the user has selected the *None of the Above* option.

For example, a user is given a list of lending agents from which to choose. If the user doesn't see the correct name on the list, he or she can select *Other* and specify the correct name. HotDocs then asks for the city in which the lending agent operates, as shown in the following script:

IF Lending Agent = OTHER(Lending Agent)

ASK Lending Agent City

END IF

In this next example, you want to create a list of company representatives. However, if the user doesn't select a company representative, you want the text *No representative selected* merged.

""
IF OTHER (Company Representative) = "None of the Above"
"No representative selected"
ELSE FORMAT (Company Representative, "a, b, and c")
END IF

PAYMENT (Rate, Term, Present Value, Future Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Term	term: Number	A number variable representing the total number of payment periods.
Present Value	present value: Number	A number variable representing the current value of the future amount.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, term, present value, future value and type of a savings scheme or loan you can use the PAYMENT function to work out how much money needs to be paid in each payment period to reach the full amount.

For example, if you have an annual rate of 7.5% over 3 years and you know that the future value is \$4,023.1381682 then you can work out the monthly payment by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate)
Term: 36 (amount of months in 3 years)
Present Value: 0
Future Value: 4,023.1381682
Type: 0

```
Payment = -100
```

This function works in much same way as the PMT(rate, nper, pv, fv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for PMT.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- TERM (Rate, Payment, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

POSITION(TEXT, TEXT, TRUE_FALSE)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable
ТЕХТ	search for: Text	The character or character string for which you want to search
TRUE_FALSE (optional)	[last instance: True/False]	A True/False variable where False indicates HotDocs should start looking for the search value at the left (beginning) of the text and True indicates HotDocs should start from the right (end) Optional Default is <i>EALSE</i>

Returns a **Number** value

The POSITION expression finds the position of a certain character or character string in a given text value. It is useful if you need to find a character you know will be in an answer but are not sure where it will appear. It returns a number value, which represents the first character.

The following script finds the hyphen in the variable, *Case Number*, and returns a number value, representing its numeric position in that given text string.

```
POSITION( Case Number, "-" )
```

":"), 2)) /60) > 17.5

In the next example, the POSITION expression is used as part of a larger computation to test whether a given time falls after 5:30 P.M. POSITION locates the colon (:) in the time value so HotDocs can process the text before and after the colon to find the correct result:

```
IF Call Time CONTAINS "p"
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
":"), 2 ) ) /60 ) > 5.5
ELSE
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
```

END IF

This script first determines if the value of *Call Time* is in the afternoon (P.M.). If it is, the script uses the POSITION expression to locate the first non-digit character (the colon) so the INTEGER expression can convert all of the digit characters leading up to it into a numeric value.

Once identified, this number represents the hours portion of the total time. Using the MID expression to locate the two digit characters after the colon, it also converts these characters into an integer and divides the value by 60. This number represents the minutes portion of the total time. The hours and minutes are added together, and if the result is greater than 5.5 (the equivalent of 5:30), the result is true. If the result is not greater than 5.5, the result is false.

The second portion of the script (after the ELSE expression) performs the same functions on a nonafternoon time value—that is, one that is either in 24-hour format or in the morning (A.M.).

	-	
laceholder	Tooltip	Replace With
IUM	base: Number	A number value, such as a Number variable, to be raised to a power

POWER(NUM, NUM)

Ρ

Ν

NUM

exponent: Number The exponent (or number that indicates the operation of repeated multiplication)

Returns a **Number** value

The POWER expression generates a numeric value, based on a given exponent.

For example, say you want to calculate the future value of an investment:

POWER((1 + Annual Rate of Return), Number of Years) * Amount Invested

HotDocs adds 1 to Annual Rate of Return and then raises it to the power of Number of Years. It then multiplies that number by Amount Invested.

PRESENT VALUE (Rate, Term, Payment, Future Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Term	term: Number	A number variable representing the total number of payment periods.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, term, payment, future value and type of a savings scheme or loan you can use the PRESENT VALUE function to work out how much the investment is worth currently (i.e. how much money you would need to invest over the same amount of payment periods to equal the return)

For example, if you have an annual rate of 7.5% over 3 years, paying \$100 a month and you know that the future value is \$4,023.1381682 then you can work out the present value by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate)
Term: 36 (amount of months in 3 years)
Payment: -100
Future Value: 4,023.1381682
Type: 0

Present Value = 0

This function works in much same way as the PV(rate, nper, pmt, fv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for PV.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- TERM (Rate, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

RATE (Term, Payment, Present Value, Future Value, Type)

Placeholder	Tooltip	Replace With
Term	term: Number	A number variable representing the total number of payment periods.

Payment	payment: Number	A number variable representing the payment made in each payment period.
Present Value	present value: Number	A number variable representing the current value of the future amount.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the term, payment, present value, future value and type of a savings scheme or loan you can use the RATE function to work out what the interest rate is per payment period.

For example, if you are paying \$100 a month over 3 years and you know that the future value is \$4,023.1381682 then you can work out the rate by using the following figures:

Term: 36 (amount of months in 3 years) Payment: -100 Present Value: 0 Future Value: 4,023.1381682 Type: 0

Rate = 0.00625 (7.5% per annum)

This function works in much same way as the RATE(nper, pmt, pv, fv, type, guess) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for **RATE**.

See also:

- TERM (Rate, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

REMAINDER(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	dividend: Number	A number value, such as a Number variable, to be divided (a numerator)
NUM	divisor: Number	A number value, such as a Number variable, by which to divide (a denominator)

Returns a **Number** value

The REMAINDER expression returns the remainder of a division. If the denominator is a zero, HotDocs generates a divide by zero error.

In this basic example, HotDocs divides 10 by 3. The remainder of that division is 1:

REMAINDER(10, 3)

In this next example, a user enters a time value in number format (such as 6 hours). However, using the REMAINDER expression (as well as TRUNCATE and ROUND), HotDocs causes the value to appear in *hours:minutes* format:

```
FORMAT( TRUNCATE ( Number of Hours, 0 ), "9" ) + ":" +
FORMAT( ROUND ( 60 * REMAINDER( Number of Hours, 1 ), 0 ), "09" )
```

This script takes the value of *Number of Hours*, which may have a decimal value, and truncates it to a whole number. Then, using the REMAINDER expression, *Number of Hours* is divided by 1 and the remainder of the division is multiplied by 60 (as in 60 minutes). HotDocs then rounds that value and brings these two values together in a string, separated by a colon. The value is then formatted correctly.

REPLACE(TEXT, TEXT, TEXT, NUM)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	The name of a Text variable or the string of characters you need to search
ТЕХТ	search for: Text	The text for which you are searching, which may include the following special characters:
		\\ backslash character
		\h non-breaking hyphen
		\l line break
		\p paragraph mark
		\s non-breaking space
		\t tab
TEXT	replace with: Text	The text you want to use as a replacement, which may include the following special characters:
		\\ backslash character
		\h non-breaking hyphen
		\l line break
		\p paragraph mark
		\s non-breaking space
		\t tab
NUM	[replace count: Number]	Controls the number of times the character is replaced
(optional)		For example, if no number is specified, all found instances will be replaced; however, if you include a 1 as a parameter, only the first found instance will be replaced. Optional. Default is <i>< all ></i> .

Returns a **Text** value

This expression lets you search a string of text for a given character string and replace the results with new text.

For example, perhaps you want to take the information in an address block (which will most likely appear as separate lines) and display it as a single line, with each "part" separated by a comma. The following script removes all line breaks (\l) from the Text variable, *Multi-Line Address*, replaces them with a comma and space, and then SETs that result to the Text variable *Single-Line Address*:

SET Single-Line Address TO REPLACE(Multi-Line Address, "\1", ", ")

RESULT

Returns a Date, Number, Text or True/False value

As you write computations, you often need HotDocs to acknowledge what the result would be at that point in the script. You can update this answer by using the RESULT expression.

For example, let's suppose you are creating a list of editors. You want to combine *Editor First Name TE* and *Editor Last Name TE* as well as the literal text *Editor*:. You must use RESULT to force HotDocs to acknowledge the result of the computation before you add the next item to the text string:

```
""
REPEAT Editor Information
RESULT + Editor First Name + " " + Editor Last Name + ", Editor" + "
"
END REPEAT
```

In this computation, the RESULT expression returns the value of *Editor First Name* and adds it to the value of *Editor Last Name*. The RESULT expression updates the list each time a new editor is added to the list. If no RESULT expression were used, HotDocs would merge just the first name entered in the list.

ROUND(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable, to be rounded

NUM	[to places: Number]	A number value that indicates the number of places (0-7) to
		the right of the decimal point after which the number will be
(optional)		rounded. Optional. Default is 0.

Returns a **Number** value

You can round a number value to a specified number of places.

The following example looks at the first two digits after the decimal point and rounds the value based on the third digit. (The resulting value is *5.93*.)

ROUND(5.9274, 2)

SELECTION(MULT_CHOICE_VAR, NUM)

Placeholder	Tooltip	Replace With
MULT_CHOICE	m: Multiple Choice Variable	A Multiple Choice variable
NUM	index: Number	A number value that indicates which selected option to return

Returns a **Text** value

This expression lets you retrieve individual options (answers) selected in a Multiple Choice variable. It returns a text value that corresponds to the defined answer (as designated by the NUM placeholder).

In the following example, you want to generate a list of employees that have various different work projects they need to complete. Multiple employees may work on one individual project. Once you have this list, you want to generate a work list report for each employee on the list.

To accomplish this, you first repeat a dialog (*Employee List*) that asks which employees are supposed to work on a given assignment (using the Multiple Choice variable, *Employee Names*). As HotDocs repeats this list, the UNION expression adds each selected, original name from each repetition to a new Multiple Choice variable, *Unique List*.

Once all of the unique answers have been added to *Unique List*, HotDocs then uses the SELECTION expression to retrieve each individual answer from *Unique List*. The result of the script merges these names in a report.

ERASE Unique List

REPEAT Employee List

SET Unique List TO UNION(Unique List, Employee Names)

END REPEAT

ERASE Project Participant

SET Index TO 1

WHILE SELECTION(Unique List, Index) != ""

SET Project Participant[Index] TO SELECTION(Unique List, Index)

INCREMENT Index

END WHILE

SPACE(TEXT, TEXT)

Placeholder	Tooltip	Replace With
ТЕХТ	t: Text	A Text variable or other expression that produces a text result
TEXT (optional)	[append: Text]	A character or text string that can be used in place of the space character. You can include the following special characters in your search:
		\\ backslash character
		\h non-breaking hyphen
		\I line break
		\p paragraph mark
		\s non-breaking space
		\t tab
		Optional. Default is [space].

Returns a **Text** value

This expression tests whether the variable is answered. If it is, it merges the answer, followed by a space character. If the variable is unanswered, it merges nothing ("").

For example, perhaps you need to merge a client's full name. Some clients, however, do not have a middle name. You can create a script that includes this middle name (if it's provided), followed by a space. If no middle name is given, nothing will be merged.

Client First Name + " " + SPACE(Client Middle Name) + Client Last Name

Sometimes you may want to merge a character other than a space. The second optional parameter for this expression allows you to specify what this character should be.

In the following example, the script uses the SPACE expression to determine if each of the variables in the address block are answered. If so, it merges the answer to the variable, followed by a line break character (rather than a space character). This merges each "part" of the address on its own line.

```
SPACE(Address Line 1, "\l") +
SPACE(Address Line 2, "\l") +
SPACE(City, ",") + SPACE(State) + SPACE(Zip Code)
```

SQUARE ROOT(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

Finding the square root of a number means finding an answer that, when multiplied by itself, gives the original number. You can use the SQUARE ROOT function to find the square root of a Number variable, for example:

SQUARE ROOT(64) = 8

STRIP(TEXT, TEXT, TRUE_FALSE, TRUE_FALSE)

Placeholder	Tooltip	Replace With
TEXT	t: Text	The name of a Text variable, or the string of characters you need to search
ТЕХТ	characters: Text	The character or string of characters for which you want to search. You can strip any alphanumeric characters from a text string, including the following special characters: (Make sure you include the backslash.)
		\\ backslash character
		\h non-breaking hyphen
		\l line break
		\p paragraph mark
		\s non-breaking space
		\t tab
TRUE_FALSE (optional)	[at beginning: True/False]	The value of TRUE if characters should be stripped from the beginning of the text. Optional. Default is <i>TRUE</i> .
TRUE_FALSE (optional)	[at end: True/False]	The value of TRUE if characters should be stripped from the end of the text. Optional. Default is <i>TRUE</i> .

Returns a **Text** value

This expression removes a specified character or characters from the beginning or end of a text answer. By default, HotDocs removes the characters from both the beginning and the end of the text. If you want to specify just one or the other, you must use the *TRUE_FALSE* parameters.

For example, perhaps you want to remove punctuation or space characters from the end of an answer (because the punctuation is already included in the document text). The following script will help you accomplish this:

```
SET Product Description TO STRIP(Product Description, " !.,?", FALSE, TRUE)
```

Because the FALSE and TRUE parameters are used, HotDocs strips the characters from the end of the answer (TRUE) and not the beginning (FALSE).

TRIM(TEXT) is equivalent to STRIP(TEXT, "\t\", TRUE, TRUE). See TRIM(TEXT) for more information.

SUM(COMPUTATION_VAR)

Placeholder	Tooltip	Replace With
COMPUTATION_VAR	c: Computation Variable	A repeated Computation variable

Returns a **Number** value

Using the SUM(COMPUTATION_VAR) expression, you can add computation values that have been repeated.

For example, let's say you have a repeated dialog that contains three variables—*Item Name, Item Amount*, and *Item Quantity*. For each line item, you create a fourth variable, a computation called *Total Amount* that multiplies *Item Amount* by the number of items the user purchases (or *Item Quantity*). You can then add all of the *Total Amount* values and receive one sum total:

SUM(Total Amount)

In order for SUM(COMPUTATION_VAR) to work properly in a template, the script that actually calculates the sum (for example, the REPEAT instruction) must be processed before the variable containing the SUM expression is asked.

Also, when using DOCX or RTF templates, headers or footers are always processed before the rest of the template text. This means that if you are using SUM(COMPUTATION_VAR) in a header or footer, the script that calculates the sum must be asked in the header or footer.

SUM(NUM_VAR)

Placeholder	Tooltip	Replace With
NUM_VAR	n: Number	A repeated Number variable

Returns a **Number** value

Using the SUM expression, you can add repeated number values.

In this computation script, HotDocs totals the values of the repeated Number variable Monthly Payment:

SUM(Monthly Payment)

In the next example, HotDocs then takes the total monthly payments the user is making and compares it to the user's monthly income. If the monthly payments are greater than 36 percent of the monthly income, the loan application is rejected:

SUM(Monthly Payment) > (Monthly Income * 0.36)

TERM (Rate	, Payment,	Present	Value,	Future	Value,	Type)
------------	------------	---------	--------	--------	--------	-------

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Present Value	present value: Number	A number variable representing the current value of the future amount.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is 0.

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, payment, present value, future value and type of a savings scheme or loan you can use the TERM function to work out how many payment periods are required to reach the full amount.

For example, if you have an annual rate of 7.5% paying \$100 a month and you know that the future value is \$4,023.1381682 then you can work out how many months you will need to reach that amount by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate) Payment: -100 Present Value: 0 Future Value: 4,023.1381682 Type: 0

Term = 36

This function works in much same way as the NPER(rate, pmt, pv, fv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for NPER.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

TEXT CONTAINS TEXT

Placeholder	Replace With
ТЕХТ	A text value, such as a Text variable
ТЕХТ	A text value, such as a Text variable. Any text you type must be in quotation marks.

Returns a True/False value

The TEXT CONTAINS TEXT expression determines whether the first text value contains the same text as the second value. If it does, it returns the value of *true*.

In this example, some states in the United States are officially recognized as "commonwealth" states. In the following script, HotDocs examines the answer the user provides for the variable *State Name* to see if the user has listed one of these states. If so, HotDocs attaches the correct designation to the merged answer:

IF "massachusetts virginia kentucky pennsylvania" CONTAINS State Name

"The Commonwealth of «State Name»"

ELSE

"The State of «State Name»"

END IF

At first glance, this computation may seem backward. You may think you would test *State Name* to see if it contains any of the commonwealths listed in the text string (for example, IF State Name CONTAINS "massachusetts virginia kentucky pennsylvania"). However, if you used that method, the answer the user assigns to *State Name TE* would have to contain everything between the quotation marks—"massachusetts virginia kentucky pennsylvania"—and you would never produce a *true* statement. Of course, you could test *State Name* against each individual commonwealth (for example, IF State Name CONTAINS "massachusetts" OR IF State Name CONTAINS "virginia" and so forth), but by "switching" the values for the placeholders and placing all of the commonwealths in one text string, you eliminate a lot of repetitive typing.

Also see **TEXT STARTS WITH TEXT** and **TEXT ENDS WITH TEXT** for similar functions.

TEXT ENDS WITH TEXT

Placeholder	Replace With
ТЕХТ	A text value, such as a Text variable
ТЕХТ	A text value, such as a Text variable. Any text you type must be in quotation marks.

Returns a **True/False** value

The TEXT ENDS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value ends with the second TEXT value.
See also TEXT STARTS WITH TEXT.

TEXT STARTS WITH TEXT

Placeholder	Replace With
ТЕХТ	A text value, such as a Text variable
ТЕХТ	A text value, such as a Text variable. Any text you type must be in quotation marks.

Returns a **True/False** value

The TEXT STARTS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value begins with the second TEXT value.

See also TEXT ENDS WITH TEXT.

TODAY

Returns a **Date** value

This expression returns the current date, according to your computer's system clock.

For example, this computation script figures out the number of days between the day the user purchased the item and today's date (or the date the user assembles the document.)

DAYS FROM(Purchase Date, TODAY)

You also can create a Date variable called TODAY. See Create a Date Variable that Inserts the Current Date.

TRIM(TEXT)

Placeholder	Tooltip	Replace With
TEXT	t: Text	A text value, such as a Text variable.

Returns a **Text** value

You can use the TRIM function to remove any white space characters from the beginning and end of a Text variable, for example:

The client, Mary Rogers, has approved...

would be returned as:

The client, Mary Rogers, has approved...

TRIM(TEXT) is equivalent to STRIP(TEXT, " t^{1} , TRUE, TRUE). See STRIP(TEXT, TEXT, TRUE_FALSE, TRUE_FALSE) for more information.

TRUNCATE(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable
NUM	[to places: Number]	A number value specifying the number of places $(0-7)$ to the right of the decimal point at which to truncate the number.
(optional)		Optional. Default is 0.

Returns a **Number** value

You can truncate a decimal number a specified number of places after a decimal point.

For example, the following script truncates the number 5.9375 to include only the first two digits after the decimal point. The truncated value is 5.93:

TRUNCATE(5.9375, 2)

In the next example, however, a Computation variable tests if the value of *Rent Amount* includes cents. The variable is then formatted to eliminate the text *AND NO CENTS* from being merged when the variable contains only a whole number:

```
IF Rent Amount = TRUNCATE( Rent Amount, 0 )
    FORMAT( Rent Amount, "NINE DOLLARS" )
ELSE
    FORMAT( Rent Amount, "NINE DOLLARS AND TWELVE CENTS" )
END IF
```

Specifically, this computation compares the actual value of *Rent Amount* against its truncated value. If the values are equal, HotDocs formats the answer to exclude the text, *AND TWELVE CENTS*. Otherwise it includes the text in the format.

The difference between the TRUNCATE and the ROUND expressions is that TRUNCATE simply "cuts off" a number at a specified digit, while ROUND actually rounds a number up or down, based on where you want the number rounded.

UNANSWERED

This expression removes an assigned value from a variable. It is used most often with the SET VAR TO VALUE instruction:

SET Property Value TO UNANSWERED

If you use this instruction, do not allow the user to change the answer by asking the variable in the interview. Because HotDocs can potentially reprocess the interview several times, the answer the user enters will always be replaced with the UNANSWERED value. Only use this value if you want a variable to be unanswered in the assembled document.

Do not confuse the value of UNANSWERED with that of NOT ANSWERED. UNANSWERED is an actual value assigned to a variable, while NOT ANSWERED (or !ANSWERED) is used to determine whether a value has been assigned.

UNION(MULT_CHOICE, MULT_CHOICE)

Placeholder	Tooltip	Replace With
MULT_CHOICE	m1: Multiple Choice Variable	A Multiple Choice variable
MULT_CHOICE	m2: Multiple Choice Variable	A Multiple Choice variable

Returns a Multiple Choice value

This expression creates a single list of all unique options (answers) that have been selected across two Multiple Choice variables.

In the following example, you want to generate a list of employees that have various different work projects they need to complete. Multiple employees may work on one individual project. Once you have this list, you want to generate a work list report for each employee on the list.

To accomplish this, you first repeat a dialog (*Employee List*) that asks which employees are supposed to work on a given assignment (using the Multiple Choice variable, *Employee Names*). As HotDocs repeats this list, the UNION expression adds each selected, original name from each repetition to a new Multiple Choice variable, *Unique List*.

Once all of the unique answers have been added to *Unique List*, HotDocs then uses the SELECTION expression to retrieve each individual answer from *Unique List*. The result of the script merges these names in a report.

ERASE Unique List REPEAT Employee List SET Unique List TO UNION(Unique List, Employee Names) END REPEAT ERASE Project Participant SET Index TO 1 WHILE SELECTION(Unique List, Index) != "" SET Project Participant[Index] TO SELECTION(Unique List, Index) INCREMENT Index END WHILE

VALUE(VAR, EXPRESSION)

Placeholder	Tooltip	Replace With
VAR	v: Variable	The name of a variable
EXPRESSION (optional)	[default: Value] The specific value you want to assign to the variable if the user leaves it unanswered. Otherwise, HotDocs will use one o the following default values:	
		"" for Text variables
	0 for Number variables	
	1 JAN 1800 for Date variables	
		FALSE for True/False variables
		"" for Multiple Choice variables

Returns a Date, Number, Text or True/False value

This expression returns a default value for the variable type if the variable is unanswered. If the variable is answered, the value is the answer the user specifies.

For example, users will sometimes purposely not answer a question in the interview. However, all variables must be answered or the script will fail. Using VALUE assigns an answer (albeit a default one) so that the script or expression can be processed correctly and still return a value.

In the following example, suppose you need to calculate a sum. However, one of the variables in the calculation, *Sales Tax*, may not be required in order to produce the result. Since the variable may be left unanswered, you can use the VALUE expression to assign a default value of 0 to the variable so that the script can be processed correctly.

The optional *EXPRESSION* placeholder allows you specify the value that should be returned if the user leaves the question unanswered. Otherwise, HotDocs will use the default value, shown in the table, above.

```
Services Cost + Product Cost + VALUE( Sales Tax )
```

YEAR OF(DATE)

Placeholder Tooltip Replace With

DATE *d*: *Date* A date value, such as a Date variable

Returns a **Number** value

You can use this expression model to find the year portion of a given date.

For example, in the following script, a new employee has temporary status until the end of his or her third month with the company. HotDocs uses the YEAR OF expression to specify the year:

DATE OF(1, MONTH OF(Hire Date + 3 MONTHS), YEAR OF(Hire Date + 3 MONTHS)) - 1 DAYS

In this example, HotDocs finds the first day of the fourth month of employment. HotDocs then subtracts one day to reveal the last day of the third month—either the 28th, 29th, 30th or 31st—depending on the month. It then merges the new date into the document.

YEARS FROM(DATE, DATE)

Placeholder	Tooltip	Replace With
DATE	start: Date	A date value, such as a Date variable
DATE	finish: Date	A date value, such as a Date variable (These values can be in any order.)

Returns a **Number** value

This expression calculates the number of years between two given dates.

In the following example, the YEARS FROM expression finds the client's age at the time of the hearing:

YEARS FROM(Hearing Date, Birth Date)

In this next example, the expression determines whether an employee has worked at the company for 25 years or more. If the employee has, HotDocs merges a paragraph into the document with details of a retirement party:

«IF YEARS FROM(Hire Date, Retirement Date) >= 25» We would like to honor you at a retirement party on «Party Date» at «Party Time» at «Party Location».

«END IF»

ZERO(NUM_VAR)

Placeholder	Tooltip	Replace With
NUM	n: Number	A Number variable

Returns a **Number** value

This expression returns the value of *zero* only if a Number variable is unanswered. If the Number variable is answered, the value is the answer the user specifies.

For example, users will sometimes leave a Number variable blank instead of entering *0*. If the Number variable is then used in a Computation variable, because it is unanswered, the computation will not be processed. The ZERO expression ensures the variable is assigned a value so the computation can return a value.

In this example, the ZERO expression is used in the Computation variable *Invoice Total* to return 0 if the Number variable *Shipping Charge* is unanswered.

```
Total Price + ZERO( Shipping Charge )
```

You can suggest default answers for unanswered variables of other types. See VALUE(VAR, EXPRESSION) for details.

Setting Values for Variables

Automatically Assign Answers to a Variable

Variables normally get their values from the answers users enter during an interview, but sometimes you may want to assign an answer to a variable instead of allowing the user to specify the answer.

For example, a document might include the address of the client and, in another place, the address of the client's spouse. Once the client's address has been entered by the user, you could use a SET instruction to automatically fill in the same address for the spouse, since it will be the same.

SET instructions are inserted using a Computation variable, which you can insert in the template where you want the SET instruction to take effect.

When you set a variable to a value, you should clear the **Save in answer file** option for the variable **(Advanced** tab of the variable editor). That way, when HotDocs builds the interview, the answer can be

set without prompting the user to save the answer file. (A changed answer file warning like this could be puzzling to users, especially if users don't add or change any answers in the interview.)

To set a variable to a value

- 1. Create a Computation variable. (See Customize a Computation Variable.)
- 2. From the Instruction models list, drag the SET VAR TO VALUE instruction into the Script field.
- 3. Drag a variable you want to receive the new answer from the **Variables** list onto the **VAR** placeholder.
- 4. Replace the VALUE placeholder with the answer you are assigning to the variable. You can do this by dragging a variable name from the Variables list, by dragging another instruction or expression into the Script field, or by entering an actual value. (See Values in SET Instructions for more details about these different types of values.)
- 5. Click **OK** after the placeholders have been replaced with actual values.

You should never SET a variable's value and then cause the variable to be asked later in the template. If you do this, the value assigned by the SET instruction will always overwrite users' answers. If you want to suggest an answer for users but allow them to change it, use the DEFAULT instruction. (See DEFAULT VAR TO VALUE instruction model and Differences Between SET and DEFAULT Instructions.)

To set two or more options for a Multiple Choice variable, separate each option with a vertical bar (for example, SET MC Variable TO "Option1|Option2|Option3").

If you intend to SET values for all the variables in a repeated dialog, you should make sure the **Trim endmost iterations whose answers are grayed or hidden** option is turned off for that dialog. Otherwise the dialog's COUNT and/or the number of repetitions that show up in the interview outline may be unpredictable.

Values in SET Instructions

You can set a variable to three kinds of values—a fixed value, another variable's value, and a value created by an expression. Be sure the variable and the value are the same type—set a Text variable to a text value, a Number variable to a number value, and so forth.

Fixed Values

A fixed value is a number, a date, a string of text, or *TRUE* or *FALSE* that is typed into the SET instruction. For example:

```
SET Receiving Office TO "Ridgefield"
```

SET Maximum Weight TO 5000 SET Client is Married TO TRUE SET Property Items TO "Furniture|Automobiles"

Be sure to follow the rules for entering fixed values—for example, text must be in quotation marks and numbers cannot include commas. See Understand the HotDocs Scripting Language for an explanation.

Variables

You can set a variable to the value of another variable. That way, the user supplies the answer. For example:

SET Receiving Office TO Customer City

Expressions

You can use an expression to produce the value for a SET instruction. For example:

```
SET Total Weight of Shipment TO (Weight of One Unit * Number of Units) + (Weight of Pallet * (Number of Units / 12))
```

Use SET Instructions to Help Write Computations

Sometimes a SET instruction makes a computation easier to write. When there's a long expression that is used in the computation more than once, set a variable to the expression and then use the variable wherever you need the expression after that in the computation.

For example, you might create a variable like *Total Weight of Shipment* just for use within the computation—it might not be used anywhere else in the template. In the following script, the SET instruction (the first instruction in the script) assigns the value of an entire expression to the variable. You can then use the variable elsewhere in the computation instead of writing the expression again:

```
SET Total Weight of Shipment TO (Weight of One Unit * Number of Units) +
(Weight of Pallet * (Number of Units / 12))
IF Total Weight of Shipment > 10000
    "Peoria Warehouse"
ELSE IF Total Weight of Shipment > 5000
    "South Chicago Warehouse"
```

ELSE IF Total Weight of Shipment > 200

"Chicago Office Warehouse"

ELSE

"Local Office"

END IF

Differences Between SET and DEFAULT Instructions

Both the SET instruction and the DEFAULT instruction assign answers to variables. However, there are differences between the way the two instructions operate in a script. These differences are outlined in the following table:

SET Instruction	DEFAULT Instruction
Sets the value of a variable each time the instruction is processed, even if the variable is already answered.	Sets the value of a variable the first time the instruction is processed and if the variable is unanswered. (If the
You should never SET a variable's value and then cause the	variable is already answered, the
variable to be asked later in the template. If you do this, the	DEFAULT instruction has no effect.)
value assigned by the SET instruction will always overwrite	
the user's answer. If you want to suggest an answer for a	
user but give them the chance to change the answer, use	
the DEFAULT instruction.	
When the instruction is processed, causes HotDocs to	When the instruction is processed,
prompt the user to save the answer file. (To avoid warnings	does not cause HotDocs to prompt
like this, clear the Save in answer file option for the variable (Advanced tab of the Variable Editor).)	the user to save the answer file.

If you intend to SET values for all the variables in a repeated dialog, you should make sure the **Trim endmost iterations whose answers are grayed or hidden** option is turned off for that dialog. Otherwise the dialog's COUNT and/or the number of repetitions that show up in the interview outline may be unpredictable.

At a Glance: The Component Manager



You can open the **Component Manager** directly from the library or from the HotDocs ribbon while editing a template.

The **Toolbar** A in the **Component Manager** runs vertically down the left side of the window. From here you can access the following options:

- **Arrange Windows:** Places the Component Manager window to the left of either the library window, Automator window, or word processor window. This way, you can work in both windows simultaneously.
- Save Component File: Saves any changes you have made while working with Component Manager. (To save changes made in the template, click the **Save Template** button in template window.)

- **TView Component File Properties:** Displays the **Component File Properties** dialog box, where you can change settings for the current component file.
- ****Insert a Variable:** Inserts the selected variable into a template at the current cursor position.
- **Create a New Component:** Creates a new component, such as a variable, clause, or dialog.
- **Edit a Component:** Displays the Component Editor, where you can make changes to the selected component.
- **Delete a Component:** Removes the selected component from the component file. If the component is used in the template, you must remove it there, as well.
- ***Rename a Component:** Displays the **Rename Component** dialog box. Renaming a component here changes all references to it in the component file; however, you must manually rename references to the component that appear in the template.
- **Duplicate a Variable:** Creates a new component by copying an existing one.
- Export Computation to Clipboard: When a computation variable is selected you can export the computation to the clipboard as HotDocs readable syntax. (See Import or Export Computations)
- **Import Computation from Clipboard:** Imports a computation from the clipboard and adds it to the component file as a computation variable (See Import or Export Computations)
- A Find in Components: Displays the Find in Components dialog box, where you can search for components that contain a specific text string.
- ^{ab}_{ac} **Find and Replace in Components:** Displays the **Find and Replace in Components** dialog box where you can locate and change text strings in your components.
- Spell Check Components: Checks text used in components (such as prompts) for accurate spelling.
- **Print Components:** Prints a list of components. You can choose to print the variable types, prompts, formatting, and any plain text resources that have been assigned.
- **BRestore all:** Brings all open component editors to the front.
- **Solution** Minimize all: Minimizes all open component editors to the Windows taskbar.
- **Save and Close:** Saves changes to and closes all open component editors.
- **W** Help: Opens the relevant page of the HotDocs Help File.

At the top right of the window you can see the **Expand** button **B**. You can click this button to display the component file of another template along side this one. This enables you to compare and share components more easily.

You can use the drop down list \bigcirc near the top of the window to filter the type of components displayed in the **Component List** \bigcirc below. You can also use the $2 \downarrow$ **Sort** button to the right of the drop down list to sort the components alphabetically. You can select components from the list then use buttons from the Toolbar \triangle to select options or double click on the component to open the Component Editor. You can also search your components list using the **Find** field \bigcirc at the bottom of the window.

To learn more about the component manager follow the links below:

- Open and Close Component Manager
- Open Component Manager at the Template Library
- Use Component Manager to Work with Components
- Create and Edit Multiple Components Simultaneously
- Change Component File Properties
- Delete a Component from the Current Component File
- Copy Components From One File to Another
- Rename Components in a Single Template
- Make a Duplicate Copy of a Variable
- Edit Formats, Merge Text, Dialog Elements, and Patterns

Open and Close Component Manager

You can edit components in a template by using Component Manager.

To open Component Manager

- 1. At the HotDocs template, click the **Component Manager** button. The **Component Manager** window appears.
- Optionally, adjust the Component Manager window to the desired width and click the Arrange Windows button. The Component Manager window appears to the left of the template development window.
- 3. Once Component Manager is opened, you can perform any number of tasks, including:
 - Using Component Manager to work with components.
 - Creating and inserting a variable using Component Manager.
 - Creating and editing multiple components simultaneously.
 - Changing component file properties.

To close Component Manager, click the **X** in the upper-right corner of Component Manager.

You can also open Component Manager at the template library. This allows you to create and edit components without actually opening the corresponding template. It also allows you to more easily edit the component file of a clause library or interview template. (See Open Component Manager at the Template Library.)

Open Component Manager at the Template Library

In addition to using Component Manager while you are editing a template (see Open and Close Component Manager), you can use Component Manager to modify the contents of a component file when you are at the HotDocs template library. This may be useful if you need to make changes in the component file, but don't want to open the corresponding template to do it. It is also useful if you need to edit an interview template (see Create an Interview Template), as well as edit clause components in a clause library.

Some changes you make in the component file are not always reflected in the template. For example, if you rename a component at Component Manager without also renaming it in the template, you will receive errors when you assemble the document. Update your template with changes as necessary to avoid such problems.

To open Component Manager at the template library

- 1. At the HotDocs library, select the template or clause library whose component file you want to open.
- 2. Click the **Component Manager** button. The **Component Manager** window appears.
- 3. Make changes to the component file as necessary, such as create new components, edit existing components, and rename components. (See Use Component Manager to Work with Components.)

Use Component Manager to Work with Components

Using Component Manager allows you to work with individual components in a HotDocs template, including copying components between component files, creating new components, and editing existing components.

You can open and close Component Manager as needed, or you can leave it displayed as you work in the template, switching between the two windows. One advantage of leaving Component Manager open is the ability to view all the components in your template at once, and edit them simultaneously as needed.

To work with individual components in a component file

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component** Manager window appears.
- 2. Perform any tasks, as described in the following table:

То	Do This
Sort components in Component Manager either by alphabetical name or by component type	Click the 2 Sort Components button and then choose either Sort by Name or Sort by Type.
	To sort component lists that appear in the Dialog Editor and Computation Editor , right- click on the list and then choose your sorting option from the shortcut menu.
Display Component Manager side by side with the template development window	Adjust Component Manager to the width you want and click the Arrange Windows button. HotDocs adjusts the size of the template development window so that both windows can be viewed in full.
Save changes you have made to the component file	Click the Save Components button.
Specify certain assembly and interview preferences for the template and component file	Click the Component File Properties button. (See Change Component File Properties.)
Use Component Manager to insert a variable in the template text	Place your cursor in the template where you want the variable, select the variable from the Components list, and click the *Insert Variable button. (See Create and Insert a Variable Using Component Manager.)
Create a new component, including a variable, dialog, or clause	Click the Components drop-down button, select the type of component you want to create, and click the New Component button.
Make changes to an existing component	Select the component from the component list and click the Edit Component button. (You can also double-click the component.)
Remove a component from the component file	Select the component in the component list and click the Delete Component button.
	The component is removed from the component file and any associated dialogs, but references to it in the template and other

	components (such as scripts, prompts, or dialog text elements) will not be updated. If you've referred to this component in any of these ways, you must manually update these references or your template may not work.
Assign a new name to a component	Select the component(s) and click the Rename Component button. (See Rename Components in a Single Template.)
Make a duplicate copy of a variable or group of variables	Select the variable(s) and click the Duplicate Variables button. (See Make a Duplicate Copy of a Variable.)
Bring all open Component Editors to the front	Click the Restore All button. HotDocs brings all open Component Editors to the front so you can view and edit them.
Minimize all open Component Editors to the Windows taskbar	Click the Minimize All button. All open Component Editors are then minimized. However, you can still access them by clicking their icon in the Windows taskbar.
Close all open Component Editors	Click the Save and Close All button. All changes made to components are saved and all open Component Editors are closed.

To access the Component Manager toolbar using the keyboard, press F10.

You can open a component file for another template while viewing the current component file. Once opened, you can edit components or assign other properties. To do this, click the **Expand** button, and then select the component file. Once open, edit the component. You can also copy components between the open component files. See Copy Components From One File to Another.

For information on searching and replacing text strings in the component file, spell checking components, and printing lists of component properties, see Search Component Files, Print a List of Components, and Spell Check Components.

Create and Edit Multiple Components Simultaneously

Using Component Manager you can create and edit as many components in a given component file as you want, all at the same time. This allows you to compare the properties, advanced options, selection

options, and computation scripts of many variables and dialogs. To manage all the open component editing windows, you can either use the **Restore All**, **Minimize All**, and **Save and Close All** buttons in Component Manager; or you can use the Windows taskbar to switch between open windows.

To edit multiple components simultaneously

- 1. Open Component Manager. (See Open and Close Component Manager.)
- 2. At the **Component Manager** window, you can either edit existing components or create new ones:
 - To edit an existing component, select it from the list of components and click the **d** Edit **Component** button. Do this for each component you want to edit. (HotDocs adds the component window to the Windows taskbar so you can more quickly access it.)
 - To create a new component, click the **New Component** button. Do this for each component you want to create.
- 3. Make changes to each variable as necessary, using the following tools to help you manage the open windows:
 - To bring all open Component Editors to the front, click the **Restore All** button.
 - To minimize all open Component Editors, click the \exists Minimize All button.
 - To save and close all open Component Editors, click the Save and Close All button. (Any changes you have made to component properties will automatically be saved—you will not be prompted to save them.)
 - To switch between open Component Editor windows, select the component in the Windows taskbar. HotDocs will bring that Component Editor to the front.

If you create new components using Component Manager, they are not immediately inserted into the template—they are only saved in the component file. You must insert the variables into the template. See Insert a Variable Field in a Text Template.

How quickly HotDocs minimizes and restores component editing windows depends on your Windows settings. To restore instantaneously, right-click on the Windows desktop, choose **Properties** from the shortcut menu, click the **Appearance** tab, click **Effects**, and then clear **Use transition effects for menus and tooltips**. (This process may differ, depending on the version of Windows.)

To change a component in several component files, use Template Manager. (See Introduction: Use Template Manager.)

At a Glance: The Component File Properties dialog box

Component File Properties - Demo Employment Agreement.cmp		
General Interview Assembly Hidden Data (Word) HotDocs Server Answer Upload		
Template <u>ti</u> tle:		
Template <u>d</u> escription:		
B		
-		
Component file format: Current Format		
Template identifier: D 419dbc78-b86e-4cb4-852d-94b4b6bfce44 New		
Supplemental files: 👩 Additional Templates 🥞 Other Additional Files		
Shared component file:		
<u>O</u> K Cancel		

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar.

In the first text field A you can edit the title of the template, this title is used to identify the template in the template library file list, and in the second text field B you can edit the template description which will appear in the **Properties** tab of the library.

Underneath the two text fields are two drop-down menus. Using the first menu you can change the component file format, this can be used to make the component file compatible with earlier versions of HotDocs (see Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs). To keep the component file current with the version of HotDocs you have installed, choose **Current Format**.

Since the release of HotDocs 11.1, HotDocs no longer supports converting component files to pre-2009/10 formats.

Below this you can see the specific Template ID D associated with this template and use the **New** button to the right to generate a different ID.

If you have additional templates or files that need to be published with this template you can add these using the next two buttons. Clicking the first one copens the **Additional Templates** dialog box where you can add or remove other templates attached to this one. Clicking the second one copens the **Other Additional Files** dialog box where you can add or remove any other types of file you would like to attach to this template, for example you could attach a PDF guide or an image file. To learn more about using these dialogs see At a Glance: Additional Templates and Files (Component File Properties).

If you choose to use an earlier format of HotDocs be careful not to use new features that aren't supported in the older versions of HotDocs or you may experience errors during assembly.

In the drop-down list G at the bottom of the dialog you can choose to use the component file from another template, the menu will show you a list of available files.

Before you point a template to a shared component file, make sure you first close the template and then edit the component file from the template library. Shared and pointed component files must also be stored in the same folder.

There are further options available in the Interview, Assembly, Hidden Data, HotDocs Server and Answer Upload tabs.

To learn more about setting the component file properties follow the links below:

- Define a Custom Interview
- Change Component File Properties
- Specify Whether Component File Properties are Shared
- Make Templates Stop Sharing Component Files
- Specify a Word Template for Storing Post-Assembly Macros
- Work with Variables in Headers, Footers, Footnotes, and Text Boxes

At a Glance: Additional Templates and Additional Files (Component File Properties)

Using Component Manager

Additional Templates	Other Additional Files
When publishing this template, always include the following additional templates:	When publishing this template, always include the following additional files:
Δ	Δ
B Add Remove OK Cancel	B Add Remove OK Cancel

To access these features

- 1. On right side of the HotDocs Developer toolbar, click **Component Manager**.
- 2. Near the top of the Component Manager window, click **Component File Properties**.
- 3. On the **Genera**l tab, next to Supplemental files, click either **Additional Templates** or **Other Additional Files**.

Both of these dialogs behave in the same way but are used to attach different file types. The main portion of these dialog boxes is the files list A where you can see any templates or files currently attached to the template

You can use the **Add** and **Remove** buttons **b** to edit the additional file lists. Clicking the **Add** button will allow you to browse for a template or file to add while clicking on the **Remove** button with a list item highlighted will remove that item and it will no longer be published with the template.

For more information about the Component File Properties tabs see the links below:

- At a Glance: The Component File Properties Dialog Box
- At a Glance: Interview Tab (Component File Properties)
- At a Glance: Assembly Tab (Component File Properties)
- At a Glance: Hidden Data Tab (Component File Properties)
- At a Glance: HotDocs Server Tab (Component File Properties)
- At a Glance: Answer Upload Tab (Component File Properties)

At a Glance: Interview Tab (Component File Properties)

Generate default interview Use custom interview Interview component: Automatically disable irrelevant variables and dialogs Hide interview outline Hide End of Interview dialog	*
Use custom interview Interview component: Automatically disable irrelevant variables and dialogs Hide interview outline Itide End of Interview dialog	- d
Interview component: Automatically disable irrelevant variables and dialogs Hide interview outline Kide End of Interview dialog	- d
Automatically disable irrelevant variables and dialogs Defa Hide interview outline Hide End of Interview dialog	ult: Grav +
Hide interview outline	and a set of a
Hide End of Interview dialog	
The chu of sheet dalog	
Generate default titles for dialogs	
Combine default dialogs	
Resource button label:	

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **Interview** tab at the top of the dialog box to access the options for HotDocs interviews.

At the top of the dialog, use the two check boxes A to tell HotDocs to generate a default interview or to use the custom interview you define. If you select **Generate default interview**. HotDocs then asks the variables and processes the instructions in the template in the order it "reads" them. If you link a variable to a dialog, the dialog is displayed instead. Select **Use custom interview;** then from the drop-down list select the interview component you want to use. If you want to edit the interview component, or if you haven't selected a component from the drop-down list, click the **Edit Component** button to the right to edit the component or to create a new one.

You can set further interview options using the following five check boxes C:

• Automatically disable irrelevant variables and dialogs

Select this option to automatically disable unused variables in the interview. Use the **Default** drop-down list to the right to select the default style of irrelevant variables:

When this option is selected you can change the irrelevant variable style of individual variables by editing the variable, then at the Advanced tab select a style from the **When** irrelevant drop-down list.

- Gray causes HotDocs to gray any irrelevant variables.
- Hide causes HotDocs to hide all irrelevant variables.
- Show causes HotDocs to ask all variables, even if the variable is irrelevant.

• Hide the interview outline

Select this option to hide the interview outline and force the user to navigate the interview by only using the **Previous** and **Next** buttons.

• Hide the end of interview dialog

Select this option to hide the end of interview dialog. Then, when the user clicks **Next** at the last dialog in the interview, depending on which End of Interview action the user defined HotDocs either sends the assembled document to the word processor or displays the **Document** tab. .

• Generate default titles for dialogs

Select this option to have HotDocs generate default titles for dialogs. The way HotDocs generates default dialog titles is by seeing if the first item in a dialog is dialog element text. If so, HotDocs displays that text as the dialog title in the interview outline. If not, HotDocs displays the prompt from the first variable and appends the word "*etc*."

• Combine default dialogs

HotDocs presents the end user with "default" dialogs containing variables that you have not (yet) explicitly linked to a dialog component. This property controls whether those default dialogs contain only a single variable each, or whether HotDocs combines all adjacent, unlinked variables into single default dialog. When a variable is linked to a dialog component, HotDocs no longer presents it in a default dialog.

In the 11.1 release, this feature is valid only for templates published to the desktop environment.

Type the text you want to use as a **Resource button label**, that the user sees when they hover their mouse cursor over the resource button during the interview, in text field **D**.

You can set further component file options in the General, Assembly, Hidden Data, HotDocs Server and Answer Upload tabs.

At a Glance: Assembly Tab (Component File Properties)

General	Interview	Assembly	Hidden Da	ata (Word)	HotDocs Server	Answer Uploa	be
Product	title:			A			
Do n	ot use answ variable nan e to the "Typ	er files nes in summa peHere* boo	aries 🔽 kmark 🗐	Mark answe Enable Edit Update tabl (Desktop or	ers in assembled do Answer at Docume e of contents, refe ly)	ocuments ent Preview tal erences, fields,	b , etc.
Unansw	ered variabl	e placeholde	r: Default	-	C		•
Post-as	sembly macr	o file (Word	only):				
Maximu	m WHILE ite	rations: 10	0	Maximu	m processing stac	k depth: 100	

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **Assembly** tab at the top of the dialog box to access the options for document assembly.

In the text field A at the top of the page you can enter a product title that will appear in the assembly window title bar. If you leave this field empty, HotDocs will use the text *HotDocs Developer 11*.

Below that are a set of six check boxes B you can tick to set more assembly options:

- **Do not use answer files** Tick this box to keep users from saving their answers or using an existing answer file for this particular template. Once selected, this option causes the first information-gathering dialog to appear immediately after the template is selected for assembly and at the end of assembly users cannot save their answers.
- Mark answers in assembled documents Tick this check box to let users see their answers highlighted in the assembled Microsoft Word document. When you do this, HotDocs marks merge fields in the **Document Preview** tab Users can then click the **Highlight Answers** button in the assembly window and answers will be highlighted with a special color.
- **Use variable names in summaries** Tick this box to have HotDocs use variable names instead of prompts when generating Question and Answer Summaries.
- Enable Edit Answer at Document Preview tab To let users edit answers while viewing the assembled document in the Document Preview tab (Microsoft Word users only) tick this check box. As users change answers, answers throughout the document (and interview) will be updated to reflect the change.
- **Move to the "TypeHere" bookmark** To position the cursor at a certain place in the document after it has been assembled check this box and create a word processor bookmark in the template. (See Position the Cursor in the Assembled Document.)
- Update table of contents, references, fields, etc Select this option to instruct HotDocs to update the document's table of contents and/or index after assembling the final word processor document. HotDocs updates these references based on any changes applied during document assembly.

The **Update table of contents, references, fields, etc** option on the Assembly tab requires a local installation of Microsoft word. This feature does not work with WordPerfect templates or templates on HotDocs Server.

From the drop-down list you can select the placeholder you would like HotDocs to insert into the assembled document in place of any unanswered questions from the interview.

If you would like to specify a Word document template where you want to store your post-assembly macros click the Rossembly macro file (Word only) field D and locate the template that contains these macros. (See PLAY "MACRO" and Specify a Word Template for Storing Post-Assembly Macros for more information.)

Below this are two number fields **E** where you can set the **Maximum WHILE iterations** and the **Maximum processing stack depth**.

To prevent HotDocs from infinitely processing a WHILE expression, which will cause HotDocs to stop responding, type a number in the number field on the left. This number represents the number of times dialogs or variables in the template or script can be looped before HotDocs stops it from doing so.

To prevent HotDocs from infinitely recursing (or processing) a computation, which will also cause HotDocs to stop responding, type a number in the right hand number field. This number represents how many

instructions you want HotDocs to allow in the processing stack. When HotDocs reaches this limit, the recursion will stop.

Recursion happens when a computation "calls" itself over and over until the desired result is achieved. For example, you can use a recursive computation to scan a text string, character by character, for a specific value. As HotDocs searches for this value, it adds information to the processing stack. If too much information gets added to this stack, HotDocs may stop responding.

There are further options available in the General, Interview, Hidden Data, HotDocs Server and Answer Upload tabs.

At a Glance: HotDocs Server Tab (Component File Properties)

Using Component Manager

 Enable template for use with HotDocs Server Interview outline initially showing Resource pane initially showing Allow user to hide/show resource pane Single-page interview initially ON Allow user to turn single-page interview on/of Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	 Enable template for use with HotDocs Server Interview outline initially showing Resource pane initially showing Single-page interview initially ON Allow user to hide/show resource pane Gupdate table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	General	Interview	Assembly	Hidden I	Data (Word)	HotDocs Serve	er Answe	r Upload	
 Interview outline initially showing Resource pane initially showing Single-page interview initially ON Allow user to hide/show resource pane Single-page interview initially ON Allow user to turn single-page interview on/of Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	 Index complete for dee more been certer. Interview outline initially showing Resource pane initially showing Single-page interview initially ON Allow user to hide/show resource pane Allow user to turn single-page interview on/ Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	V Enab	le template	for use with	HotDocs	Server				
 Allow user to hide/show interview odulite Resource pane initially showing Allow user to hide/show resource pane Single-page interview initially ON Allow user to turn single-page interview on/of Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	 Interview obtaine midally showing Resource pane initially showing Single-page interview initially ON Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 		wiew outline	initially sho	wina		ser to bide/show	interview	outline	
 Resource pane interview initially ON Image: Allow user to turn single-page interview on/of Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	 Resource pane initially showing a pane with due to indepartor resource pane Single-page interview initially ON Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	Reso	vice pape ir	nitially show	ina		ser to hide/show	resource	nane	
Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server)	 Update table of contents, references, fields, etc (applies only to Word templates being assembled on a server) 	Sinal	e-nage inter	view initially	/ ON	Allow us	ser to turn single	-nage inte	rview on/	/of
					- 					
							12			_

You can open the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar. Click the **HotDocs Server** tab at the top of the dialog box to access the options for making templates compatible with HotDocs Server.

Tick the first check box A to allow you to browser test your template and publish it with HotDocs Server.

Below are six check boxes B where you can choose to set further options for the HotDocs Server interviews. You can select :

- Interview outline initially showing
- Allow user to hide/show interview outline
- Resource pane initially showing
- Allow user to hide/show resource pane
- Single-page interview initially on

• Update table of contents, references, fields, etc

The **Update table of contents, references, fields, etc** option, on the HotDocs Server tab, does not produce exactly the same results as the desktop option of the same name on the Assembly tab. Unlike in desktop HotDocs, using this feature with server is known to occasionally alter or lose advanced formatting features in the assembled document. When converting a template for use with HotDocs server, we recommend testing all features that previously relied on the desktop version of this option.

• Allow user to turn single-page interview on/off

There are further options available in the General, Interview, Assembly, Hidden Data, and Answer Upload tabs.

To learn more about setting the HotDocs Server component file properties follow the link below:

• Enable Templates for Use With HotDocs Server

At a Glance: Hidden Data Tab (Component File Properties)

Component File Properties - Demo Employment Agreement.cmp							
General Interview Assembly	Hidden Data (W	/ord)	HotDocs Server	Answer Upload			
Abocument Data to Remove — Invisible merge ID marks Redundant metafile graph	iics	VU	nused prope <u>r</u> ty co	odes			
BDocument Properties to Remove				(2)(1)			
Title	Company		Нуре	erlink base			
Subject Category			S <u>t</u> ati	istics			
Aut <u>n</u> or	Commonto		Cust	tom properties			
HotDocs Fields to Remove							
Answer bookmarks		V E	ditable text <u>b</u> ookm	narks			
Document Markup							
Accept all tracked change	s	Π	urn o <u>f</u> f change tra	icking			
Remo <u>v</u> e all comments							
EDOCX-Specific							
Remove HTML compatibility codes							
Remove unreferenced parts and elements							
Remove saved proofing codes							
Remove _GoBack bookmark							
Select All			OK	Cancel			

You can open the **Hidden Data** tab of the **Component File Properties** dialog box by first opening **Component Manager** and clicking the **Component File Properties** button on the toolbar then selecting the **Hidden Data** tab. The options are spilt into five categories; **Document Data A**, **Document Properties B**, **HotDocs Fields**, **Document Markup**, and **DOCX-specific**.

The first section of options A has three check boxes. You can tick these to have HotDocs remove invisible merge ID marks, redundant metafile graphics, and/or unused property codes.

In the second section of options B there eleven document properties that you can have HotDocs remove:

- Title
- Company
- Hyperlink base
- Subject

- Category
- Statistics
- Author
- Keywords
- Custom Properties
- Manager
- Comments

In the third section of options **G** you can choose to have HotDocs remove the following fields: answer bookmarks and/or editable text bookmarks.

In the forth section of options **D** you can control the options for Document Markup (for more information on marking up documents see Introduction: Create HotDocs Models). You can choose to accept all tracked changes, remove all comments and/or turn off change tracking.

In the fifth section of options gou can select options specific to DOCX templates. You can choose to have HotDocs remove HTML compatibility codes, unreferenced parts and elements, saved proofing codes, and/or _GoBack bookmarks.

At the bottom of the window there are two buttons **F** that you can use to quickly **Select All** the options or **Clear All** the options in this tab.

There are further options available in the General, Interview, Assembly, HotDocs Server and Answer Upload tabs.

To learn more about removing hidden data follow the links below:

- Remove Hidden Data from Assembled Documents
- Using the Hidden Data Remover dialog box
- Remove Hidden Data from Word Templates

Change Component File Properties

You can change certain template and component file properties at the component file level. Component file properties allow you to, among other things:

- Specify default titles and descriptions for the template.
- Define the interview for the template.

- Control which parts of the assembly window (such as the interview outline and *End of Interview* dialog) appear for a given template.
- Enable answer-editing at the **Document Preview** tab.
- Choose which hidden data can be removed from an assembled document.
- Define properties for how the interview will be displayed in a Web browser.

When you change component file properties, those changes usually affect a single template. However, if you are sharing component files (see Use One Component File for Multiple Templates), some properties may be applied to the shared file while others may be applied to the pointed component file. (See Specify Whether Component File Properties are Shared.)

To change component file properties

- 1. Open Component Manager. (See Open and Close Component Manager.)
- 2. Click the **Component File Properties** button. The **Component File Properties** dialog box appears. There are multiple tabs in the Component File Properties dialog box, for a more detailed description of each tab follow the links below:
 - At the **General** tab you are able to edit the template title, template description, the format of the component file and assign a shared component file.
 - At the **Interview** tab you can set options relating to the HotDocs interviews created by this template including using a default or custom interview, hiding the end of interview dialog, combining default dialogs, or generating default titles.
 - At the **Assembly** tab you can decide how HotDocs will assemble the finished document, including setting the processing stack depth, choosing the placeholder for any unanswered variables and setting a product title.
 - At the **Hidden Data** tab you can decide which pieces of hidden data are removed from the document.
 - At the **HotDocs Server** tab you can enable the template to work with HotDocs Server and set options for HotDocs Server interviews.
 - At the **Answer Upload** tab you can set the upload URL, friendly name and whether you would like answer files to be uploaded automatically.

Since the release of HotDocs 11.1, HotDocs no longer supports converting component files to pre-2009/10 formats.

Delete a Component from the Current Component File

You can delete a particular reference to a component in the template by highlighting the reference and pressing the **Delete** key. However, to delete a variable from the component file (so that it won't appear in variable lists), you must use Component Manager. You must also use Component Manager to delete formats, patterns, merge text, dialog element text, or other components from the current component file.

Recursion happens when a computation "calls" itself over and over until the desired result is achieved. For example, you can use a recursive computation to scan a text string, character by character, for a specific value. As HotDocs searches for this value, it adds information to the processing stack. If too much information gets added to this stack, HotDocs may stop responding.

To permanently delete a variable, format, pattern, or merge text group

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. From the **Components** list, select the variable, format, pattern, or merge text you want to delete.
- 3. Click the **Delete Components** button and click **Yes** when HotDocs confirms the removal. The component is removed. If a reference to the component still exists in the template, you must remove it or you will receive errors when you assemble the template.

To delete unused components in several templates, use Template Manager. (See Delete Components from Multiple Component Files.)

When you are viewing the component list, you can limit the list of components that are showing by clicking the **Components** drop-down button and selecting the specific component type.

Copy Components From One File to Another

You can copy components from one component file into another. Copying components makes it easy to use the same variables, dialogs, example formats, and so forth in several templates. Once you have copied a variable from one component file into another, you can change its properties to fit that specific template.

When you copy a component such as a dialog or a Computation variable, all of the variables associated with the component are automatically copied as well.

To copy components from one component file to another

1. At the template, click the **Component Manager** button. The **Component Manager** window appears.

- 2. Click the **Expand** button in the upper-right corner of Component Manager. The window expands to show a second component list.
- Click the Copen button. The Open dialog box appears. (You can also click the Other component file drop-down button and select one of the component files that is saved in the same folder as the current file.)
- 4. Locate the component file you want to display and click **OK**. HotDocs shows a list of components in that file.
- 5. In either the current component file or the other component file, select the components you want to copy.
- 6. Click the **Copy** button (or the **Copy** button). HotDocs copies the components. If you copy a dialog or a Computation variable, any variables referenced by that component will also be copied.
- 7. If a component already exists in the file to which you are copying and contains properties that are different, complete any of the following steps at the **Copy Components** dialog box:
 - Click **Overwrite** to overwrite that specific instance of an existing component with the one you are copying.
 - Click **Don't Overwrite** to not copy a specific component.
 - Click **Rename** to assign a new name to the component you are copying. Both the original and the copied component are saved in the component file.
 - Click **Always Overwrite** to overwrite all existing components with the ones you are copying. Once you select this option, you will not be warned when other duplicate components are found.
 - Click **Never Overwrite** to tell HotDocs to not copy (and overwrite) any existing components with ones you are copying.

Also, at the **Copy Components** dialog box, you can view the components you are copying. To view the component in the file from which you are copying, click the first **View** button. To see the component that exists in the component file into which you are copying, click the second **View** button.

Once you have copied the components, click the **Collapse** button in the upper-right corner of Component Manager to show only the current list of components.

To copy components between several component files at once, you can use Template Manager. (See Copy and Paste Components Across Multiple Component Files.)

Rename Components in a Single Template

If you have a template that contains several components that must be renamed, you should use Template Manager, which changes the name in both the template file and the component file. It

also updates any references to the component if it is used by other components. (See Rename Components Across Multiple Component Files.)

You can rename variables, dialogs, example formats, merge text values, dialog elements, and so forth. When you rename a component, it is changed everywhere it is used in the component file. For example, renaming a variable will update all references to it in dialogs, scripts, and prompts. However, you must update references to the component in the template text itself. This may include removing the old variable field and inserting a new variable field.

How you rename components depends on the component type. Main components can be renamed in both the template file and the component file; however supplemental components (such as example formats, patterns, and so forth) can only be renamed using Component Manager.

To rename a component while editing the template

- 1. Place the cursor in the variable field and click the ****Variable Field** button. The **Variable Field** dialog box appears.
- 2. Click the **Edit Component** button. The **Variable Editor** appears.
- 3. In the **Variable name** field, type the new variable name and click **OK**. HotDocs verifies that you want to rename the component.
- 4. Click Yes. The variable component is renamed, and the Variable Field dialog box appears again.
- 5. Click **OK**.
- 6. Remove any other references to the variable in the template and insert the new variable field. (See Insert a Variable Field in a Text Template.)

To rename a single component using Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Select the component from the **Components** list and click the **PRENAME Component** button. The **Rename Component** dialog box appears.
- 3. Type the new component name in the **New name** field and click **Rename**. HotDocs changes the name throughout the component file.
- 4. In the template, replace old component references with the new component. (See Insert a Variable Field in a Text Template.)

To rename multiple components using Component Manager

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Select the components from the Components list and click the *****Rename Component button. The Rename Components dialog box appears.
- 3. In the **Existing Component Names** column, select the components you want to rename.

- 4. Type the new name in the **New Component Names** column.
- 5. Optionally, to create multiple like-named components at once (for example, to rename two variables that use the word *Plaintiff* so the new names instead use *Client*), type the existing word or phrase (i.e. *Plaintiff*) in the **Replace** field, and then type the new word or phase (i.e. *Client*) in the **with** field, and click **Apply**. Where applicable, HotDocs changes all the names in the **New Component Names** column.
- 6. Click **Rename** to rename the components.
- 7. In the template, replace old component references with the new components. (See Insert a Variable Field in a Text Template.)

HotDocs lists all the components that were renamed in the **Renamed Components** list of Component Manager. To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

At a Glance: The Duplicate Variables dialog Box

Duplicate Variables		? 💌
A Existing Variables	Туре	New Variables
Employee Name	Text	Employee Name
Employee to Complete Trial Period	True/False	Employee to Complete Trial Period
Replace B with	6	in new variable names.
	, a	Duplicate Cancel

After opening the **Component Manage**r from the HotDocs Library and highlighting two or more variables, you can click on the **Duplicate Variable** button on the toolbar to open the **Duplicate Variables** dialog box where you can set the options for duplicating multiple variables.

In the List A you can see the variables you have selected in the **Existing Variables** column. The check-box column indicates which variables will be duplicated. In the **New Variables** column you can see a list of the new variables you will be creating. You must change these variables' names, either by typing each new

name in the New Variables column, or by "batch" replacing existing words or phrases in the variable name with the new words. You do this by using the options below.

In the **Replace** field **B** you can enter the word or phrase in the new variable name you want to replace. Then in the **With** field **C** to the right you can enter the new word or phrase in the new variable name you want to use. For example, if you want to duplicate the variables Plaintiff Name, Plaintiff Address, and Plaintiff Gender to create Defendant Name, Defendant Address and Defendant Gender, you can type Plaintiff in the Replace box, and then type Defendant in this box.

You can then click the **Apply** button **D** and replace existing words or phrases in the **New Variables** column with the new word or phrase you want to use. Once replaced, you can click the **Duplicate** button **D** and HotDocs will create the new variables using the names you have suggested.

To learn more about duplicating multiple components in component manager follow the link below:

• Make a Duplicate Copy of a Variable

Make a Duplicate Copy of a Variable

You can copy an existing variable to create a new variable. This may be useful, for example, if you want to create a variable that uses most (if not all) of the same properties of an existing variable, but you want the variable to have a different name.

You must use Component Manager to duplicate variables. You can duplicate one component or many, depending on your needs. You cannot duplicate dialogs, clauses, or database components.

To duplicate a single variable

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manage**r window appears.
- 2. Select the variable you want to copy and click the **Duplicate Variables** button. HotDocs opens the **Variable Editor** and displays a copy of the variable (as shown in the **Variable name** field).
- 3. Change the name and any other properties of the variable and click **OK**. The new variable is added to the component file.

To duplicate multiple variables at once

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Select the variables you want to create copies of and click the **Duplicate Variables** button. The **Duplicate Variables** dialog box appears.
- 3. In the **Existing Variables** column, select the variable(s) you want to duplicate.
- 4. Type the name of the new variable in the New Variables column.
- 5. Optionally, to create multiple like-named components at once (for example, to copy *Defendant*-related components to create *Plaintiff*-related components), type the existing word or phrase (i.e. *Defendant*) in the **Replace** field, and then type the new word or phase (i.e. *Plaintiff*) in the **with** field, and click **Apply**. Where applicable, HotDocs changes all of the names in the **New Variables** column.
- 6. Click **Duplicate** to create the new variables.

HotDocs lists all the variables that were duplicated in the **Duplicated Components** list of Component Manager. To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

Import or Export Computation Variables

You can copy variables between your own component files by using the **Expand** button (See Copy Components From One File to Another) but if you need to pass a computation on to another person, perhaps someone off-site or in another branch you will need to export the computation from the **Component Manager**. This creates a piece of HotDocs readable syntax which, when imported back into a Component Manager, will add the whole computation to the component file.

Export a Computation to the Clipboard

- 1. Open Component Manager for the template you wish to copy a computation variable from. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Identify the Computation you wish to import and click on it once to highlight it.
- 3. Click on the **Export a Computation to the Clipboard** button.
- 4. You can then paste the computation variable syntax into a text file by type ctrl-v or paste by right clicking with the mouse.

Import a Computation from the Clipboard

- 1. Select the entire computation variable syntax and type ctrl-c or select copy by right clicking on the mouse. This puts the information back onto the Clipboard.
- 2. Open Component Manager for the template you wish to add the computation variable to. (See Open and Close Component Manager.) The **Component Manage**r window appears.
3. Click the **Import a Computation from the Clipboard** button then you will see the new computation appear in the Components List.

If you copy anything else between exporting and importing or pasting into a text file you will loose the exported computation and have to re-start the process.

Edit Formats, Merge Text, Dialog Elements, and Patterns

You can edit supplemental components such as example formats, dialog elements, merge text, and patterns using Component Manager.

To edit supplemental components

- 1. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Click the Components drop-down button and select the type of supplemental component you want to edit. Your options include variable formats (like Text Formats and Number Formats), Merge Text, Dialog Elements, and Text Patterns. The list is filtered to show that kind of component.
- 3. Select the component you want to edit and click *d* **Edit Component**. The component is opened in its component editor.
- 4. Make any necessary changes.

To create a new example format, merge text group, dialog element text, or text pattern, click the **New Component** button while viewing that specific list of supplemental components at Component Manager.

If you want to assign a name to a variable format—for instance, because a name would be more descriptive than the actual format—type the name in the **Format name** field. This name will be displayed in format lists instead of the format itself.

Using One Component File for Multiple Templates

Be sure to back up your shared component files on a regular basis. If a shared component file becomes damaged, you could lose all of the components for several templates.

When you create a template file, HotDocs automatically creates a component file for that template. The component file stores variables, dialogs, and other components you use for that specific template.

When several related templates use many of the same components, you can create all of those components just once and store them in a shared component file. When you make a change to a shared component, (for example, if you change a prompt or resource text), the change appears in all templates that use that component file.

The way you share the same components with multiple templates is by pointing each template's own component file to the shared component file. Then, as you create and edit components within the template, the template bypasses its own component file (which is now known as the pointed component file or *actual component file*) and stores the information in the shared component file instead.

You must save all templates, pointed component files, and shared component files in the same folder.

It is best practice to point all the templates you want to share a component file to the shared component file *before* you start creating components in any of the templates. That way, HotDocs saves the components directly in the shared component file so they are ready for use when you automate other templates. Otherwise, you must copy those components to the shared file so HotDocs can find them.

Pointing the current component file to the shared file does not automatically copy the components there.

When specifying component file properties, HotDocs uses properties from the shared component file for all templates that point to it (except for those specified at the **General** tab of the **Component File Properties** dialog box). You can, however, use the pointed component file's properties. Additionally, if HotDocs has specified some of the properties you want to use in the shared component file, you can copy them to the pointed component file.

If you are pointing several templates to a shared component file, but you want each template to use its own custom interview, at the **Interview** tab, clear **Use interview properties stored in the shared component file**, select **Use custom interview**, and then specify the name of the interview computation in the **Interview component** text box.

You can unpoint a component file using Component Manager.

You can point new templates to as shared component file as you make them but you can also point pre-existing templates to a shared component file:

To point the current component file to a shared component file

- 1. Make sure you have closed all templates and component editors. (The only window you should leave open is the template library.)
- 2. At the template library, select the template whose component file you want to point to the shared component file.

- 3. Click the **Component Manager** button.
- 4. Click the **Component File Properties** button.
- 5. At the **General** tab, click the **Shared component file** drop-down button and select the shared component file. (This drop-down list shows all the component files listed in the same folder as the actual component file. Remember, you must save pointed and shared component files to the same folder.)
- 6. Optionally, to use the properties of the pointed component file instead of the shared, clear Use properties stored in the shared component file at each respective tab of the Component File Properties dialog box. To copy the properties of the shared component file into the pointed component file, click Copy Shared. (HotDocs will not update any properties you change after choosing to use the pointed component file in the shared file.)
- Click OK. The current template's component file is now pointed and the contents of the shared component file appear in the component list. (At Component Manager, the Component File Properties button changes appearance to indicate that the component file is now pointed.)

If you created variables before you pointed the component file to the shared file, you must copy those variables to the shared file.

HotDocs can display a warning when you edit a template that has a pointed component file. This warning reminds you that changes you make will affect all templates that point to the component file you are editing.

Insert a PLAY Instruction in a Template

To insert a PLAY instruction in a template

- 1. At the template, position the cursor in the template where you want the PLAY instruction.
- 2. If you are using Microsoft Word:
 - Click the **HotDocs** menu in the HotDocs toolbar and choose **Other Field** from the list of options. The **Other Field** dialog box appears.
 - Click the **Field** type drop-down button and choose **PLAY**.
 - In the **Macro name** field, enter the name of the macro you want to run.
 - Click **OK**. The instruction is inserted in the template.
- 3. If you are using WordPerfect:
 - Copy an existing variable or instruction field in the template.
 - Replace text between the chevrons (« ») with the PLAY instruction.

See Specify a Word Template for Storing Post-Assembly Macros for more information.

PLAY instructions are executed when you create an actual document from the assembly. This includes sending the document to the word processor, saving the document, or printing a copy of the document. If there are multiple instructions, they are processed in the order they are encountered.

Specify Whether Component File Properties are Shared

When several related templates use many of the same components, you can create all of those components just once and store them in a *shared component file*. The way you share the same components in multiple templates is by *pointing* each template's own component file to the shared component file. Then, as you create and edit components within the template, the template bypasses its own component file (which is now known as the *pointed component file* or *actual component file*) and stores the information in the shared component file instead.

When specifying component file properties (see Change Component File Properties), you can choose which component file will provide the options for the template. By default, templates use the shared component file's properties (except for those properties specified at the **General** tab of the **Component File Properties** dialog box). However, if a certain number of templates in your set require their properties be different from the others, you can choose to use the pointed component file's properties, instead.

For example, perhaps only a few of your pointed templates will be published for use with HotDocs Server. For those specific templates, you can enable them for use with HotDocs Server and set specific properties so they can be used on the Web.

Properties (such as the **Template title** and **Template description**) specified at the **General** tab of the **Component File Properties** dialog box are always stored in the pointed component file.

To choose which component file properties should be used for a pointed template

- 1. Point the template's component file. See Use One Component File for Multiple Templates.
- 2. With Component Manager open and the **Component File Properties** dialog box displayed, click the tab for the properties you want to use from the pointed component file. (For example, to use the pointed file's interview options, click the **Interview** tab.) The view changes to show those specific options.
- 3. Clear Use properties stored in the shared component file.
- Optionally, to copy the properties of the shared component file into the pointed component file, click **Copy Shared**. Any properties specified for this group at the shared component file are copied into the current (or pointed) component file.

The ability to use pointed component file properties for templates also lets you specify custom interviews for each template in your set. Specifically, if you are pointing several templates to a

shared component file, but you want each template to use its own custom interview, at the **Interview** tab, clear **Use Interview properties stored in the shared component file**, select **Use custom interview**, and then specify the name of the interview computation in the **Interview component** field.

Any properties you change after choosing to use the pointed component file will not be updated in the shared file.

Make Templates Stop Sharing Component Files

You can unpoint a component file if you decide you no longer want to share components between multiple templates. (See Use One Component File for Multiple Templates.)

To unpoint a component file

- 1. Make sure all templates and component editors are closed. (The only thing that should be open is the template library.)
- 2. At the template library, select the template whose component file you want to unpoint.
- 3. Click the **Component Manager** button. The **Component Manager** window appears.
- 4. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 5. At the **General** tab, click the **Shared component file** drop-down button and choose the blank line at the top of the list.
- 6. Click **OK**. The component list changes to show the components used in the original component file.

If you added or created components while the component file was pointed, you must copy those components back to your original component file so HotDocs can find them. See Copy Components From One File to Another for information on how to do this.

Control Whether Users See the End of Interview Dialog

By default, when users complete interviews, the last dialog they see is the *End of Interview* dialog, which is where they choose what to do with the assembled document. In some situations, you may want to control

whether users view the assembled document automatically without displaying these options. You can keep this dialog from appearing.

To hide the End of Interview dialog

- 1. Open Component Manager for the template whose *End of Interview* dialog you want to hide. (See Open and Close Component Manager.) The **Component Manager** window appears.
- Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 3. Click the **Interview** tab and select **Hide End of Interview dialog**.

When the *End of Interview* dialog is hidden and the user clicks **Next** at the last dialog in the interview, HotDocs will perform the *End of Interview* action the user specifies at **HotDocs Options**. These options include sending the document to the word processor or Filler, or viewing the assembled document at the **Document Preview** tab. (See Control What Happens When You Finish an Interview.)

Users can choose which options appear in the *End of Interview* dialog by making their selections at **HotDocs Options**. See Customize the End of Interview Dialog.

Position the Cursor in the Assembled Document

You can cause HotDocs to place the cursor at a specified place in the assembled document once the user has completed an interview and sent the assembled document to the word processor. You do this by inserting a "TypeHere" bookmark in your template and then by selecting a component file property.

To insert a TypeHere bookmark

- 1. Create a template or open an existing template for editing. (See Create a New Text Template File or Edit a Template.)
- 2. Place the cursor in the template where you want your cursor to appear when the assembled document is sent to the word processor.
- 3. Use your word processor's functionality to create a bookmark. Name it **TypeHere** and insert it into the text. (See your word processor's help file for information on creating bookmarks.)
- 4. Open Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 5. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 6. Click the Assembly tab and select Move to the "Type Here" bookmark.
- 7. Click **OK**.

Specify a Word Template for Storing Post-Assembly Macros

You can insert PLAY instructions in templates to play macros after assembly is complete. Such macros are used to update references in the document, remove unused markup coding, apply custom formatting to answers, and so forth.

With a DOTM or DOT template, the macro to be PLAYed can simply be included in the HotDocs template itself. After assembly, HotDocs temporarily attaches the original template to the assembled document so that when the PLAY instruction is processed, Word automatically finds the macro to be PLAYed.

With a DOCX or RTF template, however, you can't store macros in the template. While you can place macros to be PLAYed in a global template saved in Word's *Startup* folder, or add the macros to *Normal.dot*, this can be inconvenient, since you must either provide instructions for the user on how to save the global template to the *Startup* folder or you must instruct them on how to modify *Normal.dot*.

To accommodate this, you can store your post-assembly macros in a Word template, which you can then associate with the HotDocs template. You specify the name of this Word template file at the **Component File Properties** dialog box for the template. This Word template must be stored in the same folder as the HotDocs template that uses it.

To use a post-assembly macro template

- 1. Create a Word DOTM or DOT template and store your macro in it. (See the Microsoft Word documentation for instructions on doing this.)
- 2. Edit the template to which you want to attach the macro. (See Edit a Template.)
- 3. Insert the PLAY MACRO instruction in the template. (See PLAY "MACRO".)
- 4. Open the **Component File Properties** dialog box for the template. (See Change Component File Properties.)
- 5. Click the **Assembly** tab, and, in the **Post-assembly macro file** field, enter the name of the template you created in Step 1. (Remember, the Word macro template must be stored in the same folder as the HotDocs template.)

Post-assembly macros are played whenever the user creates a document from the template, specifically when the user prints the document, saves the document to disk, or sends the document to the word processor.

Remove Hidden Data from Assembled Documents

Removing hidden data from an assembled document is supported in Microsoft Word only.

As you work in Microsoft Word templates or documents, some personal information as well as some hidden document properties are stored in the document. This information may compromise the security of the assembled document as well as affect the size of the document file. To keep others from reviewing this information, or to reduce the size of the file, you can choose which of these Word properties should be removed from an assembled document.

Properties you select at the **Hidden Data** tab of the **Component File Properties** dialog box will be removed from the assembled document when users send the document to Microsoft Word.

In addition to removing hidden data from assembled documents, you can also remove hidden data from Word templates you're automating. See Remove Hidden Data from Word Templates for details.

To choose which properties will be removed from an assembled Word document

- 1. Open Component Manager. (See Open and Close Component Manager.)
- 2. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 3. Click the **Hidden Data (Word)** tab. The view changes to show the properties you can remove.
- 4. Select the types of data you want removed from the assembled document. You can remove content as well as properties from the assembled document. Specifically:
 - Options in the **Document Data to Remove** group include marks or other types of data Word uses to manage information in the document:
 - Invisible merge ID marks: Word merges revision ID numbers in the template each time you edit the template text. (Word uses these numbers to improve accuracy when merging or comparing related documents.) Each time you edit the text of a template, more IDs are added to the template and the size of the file increases. Removing these IDs reduces the size of the document.
 - Redundant metafile graphics: When you insert a graphic image in an RTF template, Word merges a Windows Metafile copy of the graphic in the template as well. This means there are two versions of the graphic in the template. Removing the metafile version can significantly reduce the size of the document.
 - Unused property codes: When you apply formatting properties (such as bold, underline, alignment, etc.) to text in a template, these codes are stored in the template. Selecting this property removes any codes that may be left in the assembled document after the section of text to which the property was applied has been removed through the use of IF instructions.
 - Options in the **Document Properties to Remove** group include information stored about the document, including information about who created the document. (Document properties typically appear on the **Properties** dialog box for a given document.)

- Options in the **HotDocs Fields to Remove** group include the bookmarks HotDocs merges in a document to indicate editable answers and text.
- Options in the **Document Markup** group include reviewer's comments, such as annotations or change-tracking suggestions.

Now, when the document is assembled and sent to the word processor, the data associated with each option you've selected will be removed from the document.

Work with Variables in Headers, Footers, Footnotes, and Text Boxes

Depending on whether you use Word or WordPerfect, how you insert a variable in a header, footer, footnote, or text box is different. The following table explains how to use variables in the various areas of a word processor template.

	Microsoft Word	WordPerfect
Headers and footers	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.
Footnotes	Create the variable in Component Manager and drag it into the field. Edit the variable using Component Manager.	Create the variable directly in the field, or create the variable in Component Manager and drag it into the field. Edit the variable directly in the field or use Component Manager.
Text Boxes	Variables in text boxes are ignored during document assembly.	Create the variable directly in the text box, or create the variable in Component Manager and drag it into the text box. Edit the variable directly in the box or use Component Manager.

Using buttons in the HotDocs Navigation toolbar will have no effect while editing the contents of a header or footer.

When inserting variables or instructions in a WordPerfect header, footer, footnote, or text box, you cannot highlight text and then replace it with the field. You must simply insert the variable by clicking in the text.

HotDocs 5 users: In HotDocs 5, you had to specify a component file property that instructed HotDocs to assemble variables in Word headers and footers. HotDocs 11 automatically assembles these, so the option has been removed from the Using buttons in the HotDocs Navigation toolbar will have no effect while editing the contents of a header or footer.

If a Word template contains headers and footers with variables, you may find your variables being asked "out of order." You can create a custom interview to control the order your variables and dialogs are asked. See **Define a Custom Interview** for details.

Using HotDocs Scripting

Scripting Overview

Introduction: Understand the HotDocs Scripting Language

Learning the Language

You can instruct HotDocs to perform certain tasks using the HotDocs scripting language. For example:

- Insert one template into another
- Hide variables in dialogs
- Perform some action based on an answer the user provides
- Add up several dollar amounts
- Find the number of years between two given dates
- Search a user's answer for a certain block of text

The HotDocs scripting language consists of instructions, expressions, operators, and values—such as text, numbers, dates, or answers users enter. To help you learn this language, HotDocs provides you with instruction and expression *models*.

Specifically, an *instruction model* tells HotDocs to perform some sort of function, while an *expression model* retrieves a special value. Most models also include *placeholders*, which you must replace with a value. Possible values include:

- text strings
- number amounts
- other models
- HotDocs variables

A model will not work until all its placeholders are replaced.

In addition to using instructions and expressions, you can use operators to control how the script is processed. Most operators are common mathematical signs, but there are also Boolean operators such as *AND* and *OR*. The operator *AND* means the statement to the left and the statement to the right must both be true. The operator *OR* means either the statement to the left or the statement to the right must be true. The operator () means to perform the operation between the parentheses first.

If your script requires input from another variable, you can create a parameter, an internal variable that accepts input from the variable field. This allows you to create a script that acts like a function, allowing that single script to work against a number of different values. You can also set local variables for your

scripts which only exist within that script and do not need to be maintained as regular variable components in the component file. These are useful for things like counters for WHILE loops and flags for IF statements.

Rules for Writing Scripts

Instruction and expression models are designed to help you accurately write scripts. However, there are still certain rules you must follow, particularly when it comes to replacing placeholders with values. Failure to follow these rules may result in syntax errors when HotDocs attempts to process the script during assembly.

When using literal values:

- Decimal numbers less than one must have a zero before the decimal point (0.125 not .125).
- Numbers cannot contain commas (1250 not 1,250).
- Dates must be in the form 3 JUN 1990.
- Text strings must be inside quotation marks.

To help you format these numbers correctly, use the **Enter a Date**, **Enter a Number**, **Enter some Text**, and **Enter True or False** expression models.

When manually typing the script:

- Component names must be entered exactly as they appear in the **Components** list, including capitalization.
- Components and operators must be separated by spaces.
- Operators must be entered exactly as they appear in the **Operators** list, including capitalization.
- Variables used inside a literal text string (or between quotation marks) in a computation must be inside chevrons (« »). If the variable name is not inside chevrons, the variable name—not the variable's value—will be merged as the answer. (For example, "«Employee Name» must wait «Number of Months» months before applying.")

If you want to include a chevron in a HotDocs text template either as a part of your document text or a literal text string, you need to double the opening chevron (but *not* the closing chevron), so that HotDocs does not interpret the chevrons (and any text between them) as a field. The output is a single chevron, as follows: input: ««some text», output: «some text».

- Keywords, such as TRUE, FALSE, AND, END IF, and so forth, must be in all capital letters.
- Only **TRUE** or **FALSE** can be used for a true/false value.
- If you want the computation to return a combination of text, variables, and values, you must join (or concatenate) the data with the Add (+) operator. Otherwise, HotDocs returns only the last piece of text, variable, or value.

• When using the RESULT expression, you must first set the computation to a value. If the computation will return text, set it to a specific value by typing text inside quotation marks at the start of the script. (To set a text computation to an empty value, type nothing between the quotation marks—not even a space.) If the computation will return a number, set it to a value by typing a number at the very beginning of the script. (To set a number computation to nothing, type 0 (zero).)

Writing the Script

There are three places in HotDocs where you can write scripts: The **Computation Editor**, the **Script** tab of the **Dialog Editor**, and the **Expression** box of the **IF Field** dialog box. Each of these places provides you with the tools you need to write the script. Which tools you use, however, depends on personal preference, as well as your skill level.

If you are learning the scripting language, we suggest that you select and drag instructions and expressions from the models lists to the **Script** box. You can also drag operators and components from their respective lists into the **Script** box. If there are placeholders, replace them by dragging components or other models onto them. To correctly format literal values (such as text, numbers, dates, and true/false values) use the **Enter...** expression models.

If you prefer to use the keyboard rather than the mouse, press the **Tab** key to move from one field in the dialog box to the next. Use the arrow keys to select specific components, operators, and models, and use the **Insert** key to bring the selected element into the **Script** box. To replace placeholders in the script, insert your cursor in the placeholder text and then locate the value you want to replace it with in the lists below and press the **Insert** key.

Once you become familiar with the instructions and expressions, you can type your scripts directly in the **Script** or **Expression** box. To ensure you use the correct instruction and expression keywords and component names, you can access lists of these things using keyboard shortcuts and other options available in the script editor. When you do this, HotDocs displays an auto-complete list from which you can choose the instruction or expression you are typing.

Use Operators when Scripting

An operator is a symbol or word that causes an operation such as addition or a comparison to be performed in a computation or expression. Operators are available at the **Operators** list wherever you can create a script or expression. Most operators can be used when working with both number and text values.

There are three types of operators:

- **Comparison operators:** These compare two values of the same type (text, number, date, multiple choice, or true/false). They return values of true or false depending on whether the comparison is true or not.
- **Arithmetic operators:** These calculate new values. Operands used in the script must be the same type.
- **Logical operators:** These return a true/false value based on a logical comparison of their operands, both of which must be true or false values.

The following tables explain how each operator works:

Comparison Operator	Description
=	The two items in the comparison are of equal value. For example:
	Birth Date = 17 Dec 1989
	Employee Name = "Louisa Gehrig"
!=	The two items in the comparison are not of equal value. For example:
	IF Exhibit A != TRUE
	IF Plaintiff Gender != "Male"
<	The first item in the comparison has a lesser value than the second item. For example:
	Account Balance < 9000
	COUNTER < 10
>	The first item in the comparison has a greater value than the second item. For example:
	Dependent Age > 18
<=	The first item in the comparison is less than or equal to the second item. For example:
	Client Age <= 65
	COUNTER <= 2
>=	The first item in the comparison is greater than or equal to the second item. For example:

	Taxed Income >= 75000
CONTAINS	The value of the first item is found in the value of the second item. For example:
	"massachusetts virginia kentucky pennsylvania" CONTAINS State Name
Arithmetic Operator	Description
+	Add the different components of the script together. For example:
	Value 1 + Value 2
	Street Address + ", " + City + ", " + State
-	Subtract the different components of the script from each other. For example:
	Monthly Income - Amount of Owed Child Support
*	Multiply the different components of the script. For example:
	Purchase Price * 0.625
/	Divide the different components of the script. For example:
	Yearly Salary / 12
-	The unary minus operator results in the numeric negation of the operand (which must be a Number value). For example:
	SET Loss Amount TO - Aggregate Amount
%	The unary percent operator results in the operand (which must be a Number value) divided by 100. It is a postfix operator. For example:
	Purchase Price + (Purchase Price * 6.25%)
Logical Operator	Description
AND	The statement to the left and the statement to the right must both be true. For example:
	IF Client is Married AND Client has Children

OR	The statement to the left or the statement to the right must be true. For example:
	IF Single OR Widowed
ΝΟΤ	The NOT operator results in the logical negation of the operand (which must be a True/False value). For example: IF NOT Client is Married

HotDocs also supports two other unary operators—unary plus (+) and the dollar sign (\$). While both produce numeric results in a script, the results are exactly the same as the operands. Therefore, they should not be used in a script.

The Add (+) operator can also be used to string together (concatenate) two text values.

The final operator, the parentheses (), instructs HotDocs to perform the operation inside the parentheses first.

HotDocs operators are processed in the following order of precedence, from highest to lowest. Operators listed on the same line have the same precedence.

```
()
NOT - %
* /
+ -
= != < > <= >=
AND
OR
```

So in other words, when HotDocs is evaluating an expression, first parenthesis are used to determine order of operations, then unary operators, then multiplication and division, then plus and minus, then comparisons, then logical AND, and last logical OR.

Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts

Inserting line breaks and paragraph ends are only relevant if you are merging text into an assembled Word document. In WordPerfect, each of the commands listed below inserts a hard return. (See the WordPerfect helps for an explanation.)

When including literal text strings in computation scripts, you can have HotDocs merge line breaks, paragraph ends, and tabs in the answer. To do this, at the **Script** field, either by manually entering the characters or using a dot code, use the following information:

To insert	Do one of the following
A line break	Press Enter or Shift+Enter . When you do this, HotDocs creates a new line of text in the same paragraph.
	When you use a line break to span a literal text string across two lines in the script editor, the color coding assigned to the text string changes to the default color. This doesn't affect how the computation will be processed, but it may make it more difficult to visually recognize the different portions of your script. To fix the color coding, click the Auto Format button.
	Insert a Line Break dot code («.lb») at the place you want HotDocs to start a new line. (See Insert Characters in Text Strings.)
A paragraph end	Press ${\bf Ctrl+Enter}.$ When you do this, HotDocs inserts a paragraph mark (${\rm I\!I}$) and starts a new paragraph of text.
	Insert a Paragraph Mark dot code («.pm») at the place you want HotDocs to start a new paragraph. (See Insert Characters in Text Strings.)
A tab character	Press Ctrl+T .
	If you want the Tab key to insert a tab character (instead of you pressing Ctrl+T), click the Options button and select Tab key inserts a tab in scripts . Now, whenever you press Tab or Shift+Tab , HotDocs will insert a tab in the script instead of taking you to another field in the dialog box.
	Insert a Tab Character dot code («.tc») at the place you want the text to be tabbed. (See Insert Characters in Text Strings.)

When creating multi-line Text variables, you can force HotDocs to merge a paragraph mark (rather than a line break) when the user presses **Enter**. For details, see Customize a Text Variable.

Use the Script Editor

When writing a computation script, there are several tools you can use to make the process easier.

To use the script editor

- 1. Edit the script or expression.
- 2. At the **Script** or **Expression** field, complete any of the following tasks:

То	Do This
Have HotDocs automatically complete keywords, component names, and Multiple Choice options as you type	Place your cursor in the Script field and press Ctrl+Spacebar . HotDocs displays a list of instruction and expression keywords, constant values (such as month abbreviations), and components. Type a portion of the keyword, component name, or Multiple Choice option for which you are searching. As you type, HotDocs filters the list to show only those keywords that contain the text you have typed. Once selected, press Enter to merge it into your script.
	To keep seldom-used keywords out of the auto-complete list, click the Options button and clear Include seldom-used keywords in auto-complete list .
Access just a list of components	Press the F5 key, select the component, and press Enter.
Access just a list of keywords	Press Shift+F5, select the keyword, and press Enter.
Display a syntactical hint of how an instruction or expression should be used	Place your cursor inside the keyword and press the F7 key. HotDocs displays a small ToolTip that shows the entire model as well as the type of value it produces, if it's an expression.
Indent matching pairs of IF and REPEAT instructions based on the level of their insertion	Click the Auto Format button.
Undo (or cancel) an action you just performed	Click the Dundo button. HotDocs reverses the change you made.
Redo (or re-implement) an action you just performed	Click the 🔉 Redo button. HotDocs reapplies the change you made.
Cut or copy and paste a selected portion of the script	Select the portion of the script you want to cut or copy, and then click the Cut button or the Copy button. The script is copied to the Clipboard. To paste the script in a new location, insert your cursor at that location and click the Paste button.

i i

Find a specific string of text in the script	Click the A Find button and enter your search text in the Find what field. To find the next instance of the text, click the A Find Next button.
	To find only those instances of text that are complete words, select Find whole words only . To find only those instances that have the same capitalization as the text for which you are searching, select Match case .
Find a specific string of text in the script and replace it with another string of text	Click the the Find and Replace button. Enter the search text for which you are searching in the Find what field, and then enter the replacement text in the Replace with field. Once you have entered the required text, click Replace , Replace All , or Find Next .
	To find and replace script text in a specific block of script, select the block of script first and then click the $\frac{ab}{ac}$ Find and Replace button. At the Find and Replace dialog box, select Replace only in selected text.
Move your cursor to a specific location in the script	Click the Go To button. This displays the Go To dialog box, where you can enter the line number or character position of where you want your cursor to move.
Indent or outdent a block of the script	Select the portion of the script you want to indent or outdent and click the FIndent button or the FOutdent button. (Click repeatedly to increase or decrease the indent.)
Cause HotDocs to ignore a section of the script when it processes it, or insert a comment in the script	Highlight the block of the script you want to comment and click the Comment Block button. This puts two forward slashes in front of each line of the script, which instructs HotDocs to ignore this section. To uncomment it, highlight the text and click the Comment Block button.
Match an IF or REPEAT instruction with its END IF or END REPEAT instruction (or vice versa)	Place your cursor inside the instruction, right-click, and select Match IF/REPEAT from the shortcut menu. (You can also press Ctrl+M .)
Highlight an entire IF or REPEAT instruction block (meaning everything between a beginning and ending IF/REPEAT instruction)	Place your cursor inside the instruction, right-click, and select Select IF/REPEAT from the shortcut menu. (You can also press Ctrl+Shift+M .)
View helpful information while using the script editor, including	Click the 😨 HotDocs Help button.

accessing help topics for the different instructions and expressions you can use in your script		
Customize the way the script editor works	Click the Options button and make your changes. (See Change Script Editing Options.)	
Insert line breaks and tab characters in a script	See Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts.	
Assign formatting characteristics to literal text strings in a script	Insert the corresponding dot code. See one of the following topics for details:	
	Change Font Properties of Text	
	Insert Characters in Text Strings	
	Add Punctuation and Capitalization to Sentences	

To access the script editor toolbar using the keyboard, press F10.

To access help for each instruction or expression, first select the model in one of the lists and then press **Ctrl+F1**.

For an explanation of the HotDocs scripting language, see Understand the HotDocs Scripting Language. For details on creating a Computation variable, see Customize a Computation Variable. For a list of instruction and expression models, see Introduction Instruction and Expression Models.

Local Variables

When writing more complex computation scripts, you may need to create temporary or special-purpose variables that are only used or needed in a particular computation or dialog script. Such temporary variables create a maintenance burden since they always show up in component lists and can require extra explanation for whoever maintains your templates in the future. To make your HotDocs scripting more maintainable, and easier to read, use local variables in these situations.

A **local variable** is somewhat similar to a regular variable, but instead of being usable anywhere in your template, it is only defined in the "local" context of a specific computation or dialog script. Local variables are useful in scripting situations where you would otherwise resort to defining a regular HotDocs variable just to keep track of temporary information that is not meaningful anywhere else but in this one script.

As an example, suppose you're writing a computation to remove spaces that may have been entered as part of an account number. You could do this using a WHILE instruction (link) to loop through each character of the account number, removing spaces as you find them:

```
SET Index TO 1
WHILE Index <= LENGTH( Account Number )
    IF MID( Account Number, Index, 1 ) = " "
        SET Account Number TO FIRST( Account Number, Index -1 )
        + LAST( Account Number, LENGTH( Account Number) - Index)
    ELSE
        INCREMENT Index
    END IF</pre>
```

END WHILE

Notice how this script relies heavily on the "Index" number variable. It keeps track of our position as we look through each character in the account number. This is a good example of a variable that should be defined "locally" instead of being defined in the Component Manager (as you would have done in earlier versions of HotDocs). As a local variable, Index is meaningful (and has an answer) only in the context of this script. You can create a local variable called "Index" in another computation as well, but the two would be isolated: setting one computation's local Index to a certain value would have no effect on the other computation's local Index. In fact, local variables do not store their answers in the answer file like regular HotDocs variables. These answers are stored in a special temporary location and are discarded as soon as the computation or script finishes running.

Here is a table that summarizes the differences between traditional HotDocs variables and local variables:

	Regular Variables	Local Variables
Defined in	Component Manager	The Locals tab of a Computation Editor or Dialog Editor window
Usability / Scope	Anywhere: in the template itself, in any script, or in other variables' prompts or resources	Only in the script where it is defined
Naming rules	Name must be unique across all the components in the component file	Name must only be unique within the script where it's defined
Available types	Text, Number, Date, True/False, Multiple Choice	Text, Number, Date, True/False
Shows up in general component lists?	Yes	No
Can be asked on a dialog?	Yes	No

Can be saved in an answer file?	Yes
Initial or default value	UNANSWERED

No UNANSWERED

You can use regular HotDocs variables (perhaps with "Ask automatically", "Warn when unanswered" and "Save in answer file" all turned off) in every instance where you could use a local variable. However, there are advantages to using local variables when possible, and these advantages become more significant as your template automation projects grow larger and more complex:

- Large or complex component files are easier to maintain when component lists are not cluttered with temporary variables.
- Naming local variables is easier, and scripts using local variables can be easier to read, because the name only has to make sense in the context of the script where it's defined and not in a global component list. In the above example, it is realistic to call the local variable "Index" instead of needing to call it "Account Number Index Temp," for example.
- Reusing components by copying them from one component file to another becomes less complicated because such copy operations no longer need to bring along additional temporary variables.

To Create a Local Variable

- 1. Insert a variable, selecting **Computation** as the variable type.
- 2. At the Variable Field dialog box, click the **dialog Edit Component** button to open the **Computation Editor.**
- 3. Click on the **Locals** tab.
- 4. Create a local variable
 - 1. Clicking the left-hand column of the **Parameter** table and enter a name
 - 2. Click the right-hand column to select the type of variable this will represent.
- 5. Click the **Properties** tab and enter the computation you want to use your local variable in. Drag and drop from the **Parameters and Local Variables** list at the bottom left.

You can tell at a glance whether local variables or parameters have been defined for the current computation or dialog script by looking at the Locals tab. If the Locals tab has an asterisk (*) on it, at least one local variable or parameter has been defined.

To view the defined local variables and drag them into your script as necessary, choose "Parameters and Local Variables" from the Components drop-down in the bottom left of the script editor.

You can define a local variable that has the same name as a regular variable elsewhere in the component file. In such a situation, HotDocs recognizes the local name before looking for the other component. For example, suppose you have a computation variable that defines a local variable called "Temp Number," and there is also a number variable called "Temp Number"

defined in Component Manager. If you refer to "Temp Number" within the computation where the local is defined, you will get the local answer associated with it. However, referring to "Temp Number" in the template, or in another script, will retrieve the regular answer from the answer file.

Parameters

When building complex templates or template systems, you may need to duplicate portions of script or logic multiple times. These duplications may be exact or with minor changes. Duplicating script or logic makes templates difficult to maintain because as times goes by and the logic needs refinement, you may introduce errors if you do not update each instance of the logic consistently. To make your HotDocs scripting more maintainable, and often easier to read, encapsulate reused logic in computation variables that use parameters.

A **parameter** is like a local variable for a computation, but instead of initializing the parameter in your script (as you might initialize a local variable using the SET instruction), the initial value of the parameter is copied *into* your computation from wherever the computation was invoked.

For example, this computation determines whether a particular child is under 18 years of age:

Computation name: Decedent Child Under Eighteen

Script:

AGE(Decedent Child Birthdate) < 18

The part of the script that determines whether a child is a minor may need to be duplicated in numerous places. You could define similar computation variables that only differ in the date variable to which they are referring, causing unnecessary extra work and cluttering up your components list, or you could use parameters.

Parameters allow you to write the logic only once:

```
Computation name: "Under Eighteen"
Parameter: "person" (Date)
Script:
```

AGE(person) < 18

Instead of referring to another date variable ("Decedent Child Birthdate" in the former example), the computation refers to a local parameter. So, whenever this latter computation is referred to, instead of simply referring to it by name, we must also provide a date as "input" to the computation:

Under Eighteen (Decedent Child Birthdate)

The very same computation can be referred to ("called") from elsewhere using different date variables or date expressions:

```
Under Eighteen ( Beneficiary Birthdate )
Under Eighteen ( Beneficiary Birthdate + 5 YEARS )
```

In this way, parameters essentially allow you to write computations that work much like HotDocs' built-in expression models. They allow you to re-use script logic rather than duplicating entire computation variables to perform the same action on different values. This is very much like defining functions within other programming environments.

As a more complex example, let's consider a longer computation that calculates the date of the nearest Monday on or after a given date. It uses a parameter and a local variable.

Computation Name: "Next Monday" Parameter: "origin date" (Date) Local Variable: "days away" (Number) Script: // Find the number of days between origin date and Monday (which is day of week no. 2) SET days away TO (2 - DAY OF WEEK(origin date)) // If origin date is later in the week than Monday, add 7 days to get to next week IF days away < 0 SET days away TO days away + 7 END IF // calculate the date origin date + days away DAYS

A computation with parameters can be referred to (called) from any other computation or script. It can also be used in merge fields, either directly in a template or in literal text: «Next Monday(TODAY):04 July 2012» will use the above computation variable (Next Monday) to calculate the next Monday after the current date (TODAY), format it according to the given format example (04 July 2012), and merge the result wherever the field appears (whether in a template, prompt, resource, etc.).

When you add, remove, or change the parameters of a computation, you are changing the way that computation is referred to by its callers. In this way, changing the types or order of a computation's parameters is akin to changing the computation's name: care should be taken to not break things.

For users with a prior knowledge of programming it may be useful to think of HotDocs Parameters as Value Parameters. Therefore any change made in the script does not change any variable that has been passed in when calling the computation.

Computations used as filters on REPEAT instructions are not permitted to have parameters.

To create a computation variable that accepts a parameter:

- 1. Insert a variable, selecting **Computation** as the variable type.
- 2. At the Variable Field dialog box, click the **dialog box**, click t
- 3. Click on the **Locals** tab and create a parameter by clicking on the left hand column of the **Parameter** table and entering a name, then the right hand column to select the type of variable this will represent.
- 4. Click on the **Properties** tab and enter the computation that you would like your new parameter to complete. In the computation use the parameter (drag and drop from the **Parameters and Local Variables** list at the bottom left) as the placeholder for the variables you would like to use later.
- 5. Click **OK** on the open dialogs to return to the template.

To use this computation variable from another computation or script,

- 1. Open the **Computation Editor** for the computation from which you wish to use your computation with parameters.
- 2. From the list of **Available Components** in the lower left corner, drag and drop your computation with parameters into the script. HotDocs will automatically generate placeholders for each of the parameters required by the computation.
- 3. Drag and drop variables or values to replace the placeholders as you would do when replacing placeholders for the built-in expression models.

To use a this computation variable from a field in a template

- 1. Click on the **Variable Field** button on the HotDocs tab and select Computation from the **Type** list.
- 2. At the **Variable** field use the drop-down list to find and click on the parameterized computation. You will now see a table with available parameters, and an empty **Expression** column.

3. Click the button to the right of the table and drag the variable you would like to use from the **Variables** list up into the **Expression** field. (You can also manually enter or edit expressions here.)

To test a computation that includes parameters

- 1. Create a computation that includes parameters.
- 2. Click the **Test** button.
- 3. To facilitate testing, HotDocs presents an automatically-generated dialog to gather values for the computation's parameters. Enter a value for each parameter. (Note that this dialog is only part of a test assembly for the computation. In actual usage, the call to the computation would specify the parameter values.)
- 4. If your computation causes subsequent dialogs to be asked, enter values for those additional variables as well.
- 5. Switch to the **Result** tab as necessary to view the result of your computation given the values you entered.
- 6. Switch back to the **Interview** tab to test results given other input.

Just as with regular computations, you can also use the DEBUG instruction to step through a computation line by line and observe how the values of parameters and other variables are affected by your script.

Using Instruction Models

Full List of Instruction Models

Instruction	Description
ADD TEXT TO MULT_CHOICE_VAR; CLEAR MULT_CHOICE_VAR	These instructions allow you to modify options of a Multiple Choice variable. The CLEAR instruction removes all the current options, and the ADD instruction adds options to the variable.
ASCEND VAR; DESCEND VAR	The ASCEND instruction sorts lists of answers (gathered using a REPEAT instruction) in alphanumeric order, from 1 to 9, and from A to Z. The DESCEND instruction sorts lists of answers from 9 to 1, and from Z to A.
ASK DIALOG	The ASK DIALOG instruction allows you to control the order in which dialogs appear in an interview. (See Control When Your Dialogs Appear.)

ASK VAR	Sometimes a variable needs to be asked by itself. You can use the ASK VAR instruction so that during the interview, HotDocs displays the variable in its own default dialog.
ASSEMBLE "FILENAME"	You can use the ASSEMBLE instruction to add templates to the assembly queue. Unlike the INSERT instruction, an ASSEMBLE instruction waits until the current document is assembled before starting the next assembly session.
CONCEAL VAR	This instruction, which you use in a dialog script, keeps variables from appearing in the Select From Answer Source dialog box of an answer source.
DEBUG	This instruction steps through the template or script field by field or line by line. It helps you determine why the template you are automating (or the script you are writing) is producing results you don't expect.
DEFAULT VAR TO VALUE	This instruction suggests a value for a variable if the variable is unanswered.
ERASE VAR; ERASE DIALOG	The ERASE instructions let you clear answers in a dialog. Specifically, ERASE VAR clears answers for a specified variable in a dialog, while ERASE DIALOG clears all answers in the dialog. This may be useful when you are using a temporary dialog to store lists of answers from two or more dialogs.
FILTER COMPUTATION_VAR	The FILTER instruction filters out certain entries from a repeated list, based on conditions you specify.
FORMAT "LIST_FORMAT"	The FORMAT "LIST_FORMAT" instruction allows you to create a sentence- style list within a computation. (If you create the REPEAT instruction using a REPEAT field, you can specify the list format by choosing a style from the Format drop-down list. See <u>Punctuate a Sentence-Style List</u> .)
GRAY ALL; GRAY VAR; UNGRAY ALL; UNGRAY VAR	These instructions, which are used in a dialog script, control whether components in a dialog appear grayed or ungrayed, depending on answers a user enters. GRAY ALL dims all components in the dialog, while UNGRAY ALL makes all of the components active again. Likewise, GRAY VAR dims a single component, and UNGRAY VAR enables the component again.
HIDE ALL; HIDE VAR; SHOW ALL; SHOW VAR	These instructions, which are used in a dialog script, control whether the user is able to see variables in a dialog. The HIDE ALL instruction hides all variables in the dialog, while SHOW ALL reveals the variables again. Likewise, HIDE VAR hides a single variable, and SHOW VAR reveals the variable.
IF EXPRESSION; ELSE IF; ELSE; END IF	You can make sections of templates or scripts conditional by using IF instructions. A conditional section will be included only if a condition you specify is <i>true</i> . The ELSE IF instruction allows two or more conditions to be included in an IF instruction. The ELSE instruction establishes a final condition for an IF instruction, specifying that if all preceding conditions are false, the following information should be included. It must be the last item of the IF instruction.

INCREMENT NUM_VAR; DECREMENT NUM_VAR	The INCREMENT and DECREMENT instructions cause HotDocs to increase or decrease a number variable, usually a counter, by the value of <i>1</i> .
INSERT "FILENAME"	This instruction inserts a clause, a clause library, or a template into the document currently being assembled. When HotDocs encounters an INSERT instruction, it immediately processes the instruction and inserts the template, clause, or clause library into the current document. If there are variables to be answered, HotDocs presents them before finishing the interview of the main document.
LANGUAGE CODE	This instruction tells HotDocs to format numbers and dates in a particular language.
LIMIT NUM	The LIMIT instruction limits the number of times a dialog can be repeated. It is placed in the script of the dialog that must be limited to a specific number of repetitions.
OMIT VAR	The OMIT VAR instruction, which you use in a dialog script, keeps variables from appearing in the Edit Answer Source dialog box of an answer source.
PLAY "MACRO"	The PLAY "MACRO" instruction plays a word processor macro after the document is assembled and either sent to the word processor, printed, or saved.
QUIT	The QUIT instruction allows you to close the variable without losing the work you have done. Normally, HotDocs will not save an invalid computation. The only way to exit an invalid computation is to click Cancel , which erases the script. The instruction is usually placed at the beginning of an unfinished or invalid computation.
REPEAT DIALOG; END REPEAT	A REPEAT instruction gathers lists of answers and merges them into a document.
REQUIRE ALL; REQUIRE VAR	The REQUIRE instruction requires users to answer questions in a dialog before they can advance to the next dialog in the interview.
SET VAR TO VALUE	This instruction lets you specify a given value for a variable's answer automatically, rather than allow the user to specify an answer. With the SET instruction, you can transfer names and other values from one variable to another.
WHILE EXPRESSION; END WHILE	The WHILE EXPRESSION instruction allows you to repeatedly process (or loop through) an answer or set of answers until a certain condition is met, such as a certain answer is found or a limit is reached.

ADD TEXT TO MULT_CHOICE_VAR; CLEAR MULT_CHOICE_VAR

Placeholder	Replace With
TEXT	A text value, such as a Text variable or an actual word or name
MULT_CHOICE	Any Multiple Choice variable in the template

These instructions allow you to modify options of a Multiple Choice variable. The CLEAR instruction removes all the current options, and the ADD instruction adds options to the variable.

Using these two models, you can create an entire Multiple Choice variable using answers the user provides.

For example, a template requires the user to first enter a list of all the committee member names. Later, the user can identify which committee member is the chairperson. You can allow the user to identify the chairperson by presenting a Multiple Choice variable that has the names of all committee members as options. The following script would create that Multiple Choice variable:

CLEAR Committee Chairperson REPEAT Committee Members List ADD Member Name TO Committee Chairperson

END REPEAT

In this example, the CLEAR instruction first removes any existing options from the Multiple Choice variable *Committee Chairperson*. Then, the script repeats the *Committee Members List* dialog, gathering the names of each committee member. The ADD instruction then adds each member's name to the Multiple Choice variable. After the committee members have all been entered, you can present the *Committee Chairperson* variable for the user to identify the chairperson.

Additionally, if the Multiple Choice variable you are clearing uses any option prompts, these prompts will also be cleared. When adding new options to the Multiple Choice variable, you can also add new prompts. To do this, use a vertical bar to separate the option from the prompt in the ADD instruction. For example:

CLEAR Marital Status ADD "Single|Client is single" TO Marital Status ADD "Married|Client is married" TO Marital Status ADD "Divorced|Client is divorced" TO Marital Status

In this example, the Multiple Choice variable *Marital Status* is cleared. The script then adds a literal text value (denoted by quotation marks) for the new option. The option is immediately followed by a vertical bar, which tells HotDocs to use the text following the bar as a prompt for the option.

Both the CLEAR and ADD instructions must be used in a computation script, which must be processed before the Multiple Choice variable is asked. To add options to a Multiple Choice variable, you should

create the variable beforehand and assign a temporary option. Then, as the user provides the answers you want to include as options, the CLEAR instruction removes the temporary option, and the ADD instruction places the user's answers as options in the variable.

Unlike options and prompts, the merge text for a Multiple Choice variable comes from another component—a Merge Text component. This means you cannot create and add merge text to a Multiple Choice variable "on the fly."

ASCEND VAR; DESCEND VAR

Placeholder	Replace With
VAR	A repeated variable

The ASCEND instruction sorts lists of answers (gathered using a REPEAT instruction) in alphanumeric order, from 1 to 9, and from A to Z. The DESCEND instruction sorts lists of answers from 9 to 1, and from Z to A.

For example, the following script would insert a list of clients in alphabetical order, from A to Z. Even though it lists *Client First Name* first, it sorts by *Client Last Name*:

```
"" REPEAT Client Information
ASCEND Client Last Name
RESULT + Client First Name + " " + Client Last Name + "
"
```

END REPEAT

The first empty set of quotation marks sets the computation value to nothing. Then, the repeated dialog, *Client Information*, asks for each client's first and last name. Next, the ASCEND instruction sorts the list of names by last name. Finally, the RESULT expression tells HotDocs to merge the names in the document.

The ASCEND and DESCEND instructions can only **sort** on a single variable; however, you can sort multiple variables by including multiple ASCEND or DESCEND instructions.

ASK DATABASE



The ASK DATABASE instruction allows you to control when the record selection screen of a database component is shown in an interview. (See Control When Your Record Selection Screens Appear.)

In general, the ASK instruction tells HotDocs to display a record selection screen or a dialog as soon as the instruction is processed. You can insert an ASK instruction directly in the template, or use a Computation variable. In fact, you can control the entire interview by using a series of ASK and other instructions in a single computation. (See ASK DIALOG and Define a Custom Interview for details.)

ASK DIALOG

Placeholder	Replace With
DIALOG	Any dialog in the template

The ASK DIALOG instruction allows you to control the order in which dialogs appear in an interview. (See Control When Your Dialogs Appear.)

ASK instructions are also useful when some interview questions should only be asked in certain situations. In the following example, HotDocs only asks the *Buyer Information* dialog if the user is a first-time buyer:

«IF First Time Buyer» «ASK Buyer Information» «END IF»

In general, the ASK instruction tells HotDocs to display a dialog as soon as the instruction is processed. You can insert an ASK instruction directly in the template, or use a Computation variable to ask several dialogs at once. In fact, you can control the entire interview by using a series of ASK and other instructions in a single computation. (See Define a Custom Interview for details.)

ASK VAR

 Placeholder
 Replace With

 VAR
 Any variable in the template

Sometimes a variable needs to be asked by itself. You can use the ASK VAR instruction so that during the interview, HotDocs displays the variable in its own default dialog.

In the following example, an IF expression evaluates if more than a year has passed since the last time the client assembled this document. If so, the *Insurance Company Name* variable is asked:

ASK Insurance Company Name

END IF

You can insert the ASK VAR instruction directly in the template or you can use it in a Computation variable. Be aware, however, that if you insert the instruction directly in the template, you can ask only one variable, and any answers the user provides will not be merged into the document at that place.

ASSEMBLE "FILENAME"

Placeholder	Replace With
FILENAME	The name of another HotDocs template

Complete instructions on using the ASSEMBLE instruction can be found in the topic, Start a New Assembly From a Template.

You can use the ASSEMBLE instruction to add templates to the assembly queue. Unlike the INSERT instruction, an ASSEMBLE instruction waits until the current document is assembled before starting the next assembly session.

For example, you may or may not need to include a cover letter with the document you are assembling. If you do, the following script adds the template, *Cover Letter*, to the assembly queue.

«IF Cover Letter Required» «ASSEMBLE "Cover Letter.docx"» «END IF»

In this script, if the user opts to assemble a cover letter, then the ASSEMBLE instruction is processed and HotDocs assembles the cover letter template after the main document is assembled.

Understand How Templates Are Added to the Assembly Queue

To achieve the best results when using the ASSEMBLE instruction, you should understand how HotDocs adds the templates it finds in ASSEMBLE instructions to the assembly queue—particularly if your template uses a custom interview. There are two different scripts that HotDocs processes during an assembly:

- The *template* script is the body of the template where HotDocs merges the answers to produce the assembled document.
- The *interview* script is the set of instructions that determine what dialogs to include in the interview. If your template uses a custom interview, this is your custom interview computation (e.g., INTERVIEW). Otherwise, the interview script is the same as the template script for a default interview.

Each time HotDocs processes one of these scripts, it first removes from the Assembly Queue any templates that were added by ASSEMBLE instructions in the current template. Then, as HotDocs encounters each ASSEMBLE instruction, it adds that template to the assembly queue. Thus, if you have an ASSEMBLE instruction in your custom interview script, the template in that instruction will be added to the assembly queue whenever HotDocs processes your interview script.

This behavior can cause a problem if the ASSEMBLE instructions in your interview script do not match the instructions in the template script. For example, if HotDocs adds templates to the assembly queue while processing the interview script, those templates could be removed when HotDocs subsequently processes the template script. Likewise, if your template script adds templates to the assembly queue, they could be removed when the interview script is processed.

To ensure that templates are added to, and remain in, the assembly queue, you should include identical ASSEMBLE instructions in both your interview and template scripts. This way, no matter which script HotDocs processes last, it will put the desired templates in the assembly queue. (As noted above, both scripts are identical if you use a default interview, so this information is only important when you use a custom interview in your template.)

You can add command-line options to an ASSEMBLE instruction (for example, ASSEMBLE "subpoena.docx /pr"). If the command-line option includes a file path and name, enclose the path and name in double quotation marks (for example, ASSEMBLE "subpoena.docx /sa /af=""L Chang.anx"""). (Four command line options were designed specifically for use with ASSEMBLE instructions. They are: Suggest Save, Suggest Save New, Save Answers, and Save Answers Prompt. They control the saving of answers after each ASSEMBLE instruction is processed.)

CONCEAL VAR

Placeholder	Replace With
VAR	A variable used in an answer source

This instruction, which you use in a dialog script, keeps variables from appearing in the **Select From Answer Source** dialog box of an answer source.

Answer sources are associated with specific dialogs in a template. When users view the dialog during the interview, they can click a special button, which displays a list of answers that were entered during previous interview sessions. They can either select an existing set of answers or add a new set to the list.

After creating an answer source for one dialog, you can use it with other dialogs, including dialogs in other templates. However, when using an answer source with multiple dialogs, each variable must be represented in both the answer source and in each dialog. If a variable that is referenced in the answer source isn't included in the dialog (or vice-versa), answers in the answer source will get "mixed together" whenever you add, edit, or delete records.

In some situations though, it isn't always practical or relevant to show the user every variable—either in the dialog or in the answer source. To accommodate this, you can use the HIDE, CONCEAL, and OMIT instructions to manipulate these variables in both the dialog and the answer source. Specifically, HIDE keeps a variable from appearing on the dialog, while CONCEAL keeps it from appearing in the answer source. OMIT keeps it from being associated with the answer source at all. Often, you must use a combination of these instructions to achieve your desired result.

For example, in one template, the user must enter both a *Creditor Name* and a *Creditor Address*. However, in a second template that uses the same dialog and answer source, *Creditor Address* isn't needed. You can keep it from appearing in the **Select From Answer Source** dialog box for this template by using the following script:

CONCEAL Creditor Address

Answers for *Creditor Address* are still saved in the answer source, even though they do not appear when the user opens the **Select From Answer Source** dialog box. (You would most likely include the instruction *HIDE Creditor Address* in this script as well, which would keep *Creditor Address* from appearing on the actual interview dialog. To keep the variable from appearing when the user edits a record in the answer source, see OMIT VAR.)

See Suggest an Answer Source for Dialogs for more information.

DEBUG

This instruction steps through the template or script field by field or line by line. It helps you determine why the template you are automating (or the script you are writing) is producing results you don't expect.

For full information on how to use this instruction, please see the following topics:

- Introduction: Debugging Templates
- Insert Debugging Instructions in Templates and Scripts
- Step Through a Template or Script

DEFAULT VAR TO VALUE

Placeholder	Replace With
VAR	Any type of variable in the template
VALUE	A value that corresponds with the variable type

This instruction suggests a value for a variable if the variable is unanswered.

For example, in the following script, the variable *Attorney Name* is defaulted to the literal value of *Sam Jones*:

DEFAULT Attorney Name TO "Sam Jones"

In this example, when HotDocs processes this script, it first determines whether *Attorney Name* has been answered. If it has, the DEFAULT instruction has no effect and HotDocs uses the answer already given. If *Attorney Name* has not been answered, however, HotDocs suggests the answer *Sam Jones*. When this variable appears during the interview, users can accept this answer by moving to the next dialog, or they can enter a different answer in the answer field.

You can also use a DEFAULT instruction to suggest an answer that has already been given in the interview. For example:

DEFAULT Trustee Name TO Client Name

In this example, when HotDocs processes this script, it first determines whether *Trustee Name* has been answered. If it has, the DEFAULT instruction has no effect and HotDocs uses the answer already given. If *Trustee Name* has not been answered, HotDocs then checks to see what answer has been given for *Client Name* and suggests that as the answer. It is important to note, however, that if *Client Name* is unanswered, *Trustee Name* will likewise be unanswered.

Do not use the DEFAULT instruction in the script of a repeating dialog unless the instruction is used in conjunction with a conditional expression (on another variable in the same dialog, at the

same REPEAT index) or a **LIMIT** instruction. If you use it by itself in a repeated dialog script, it will always add an unanswered dialog to the interview, which will produce an incorrectly assembled document.

The DEFAULT and SET instructions both assign answers to variables. Click here for an explanation of the differences between the two.

To default two or more options for a Multiple Choice variable, separate each option with a vertical bar (for example, DEFAULT MC Variable TO "Option1|Option2|Option3").

ERASE VAR; ERASE DIALOG

Placeholder	Replace With
VAR	The name of a variable in a dialog you want to clear
DIALOG	The name of the dialog whose contents you want to clear

The ERASE instructions let you clear answers in a dialog. Specifically, ERASE VAR clears answers for a specified variable in a dialog, while ERASE DIALOG clears all answers in the dialog. This may be useful when you are using a temporary dialog to store lists of answers from two or more dialogs.

For example, say you have two repeated dialogs—one containing plaintiff names and another containing defendant names. If you need to generate a single list of all parties in the case, you can combine the two lists into a single repeated dialog. To do this, you would want to erase any existing values from the combined list before populating it with the names from the plaintiff and defendant dialogs.

Using the ERASE instruction in the script keeps the combined list up to date each time the computation script is processed. For example, if the user adds or removes names in the plaintiff or defendant dialogs during the interview, the ERASE instruction will make sure they are properly added or removed when the combined list is regenerated.

The following script demonstrates how to accomplish this:

ERASE Combined List SET Counter TO 0 REPEAT Plaintiff Information INCREMENT Counter SET Combined Name[Counter] TO Plaintiff Name
END REPEAT REPEAT Defendant Information INCREMENT Counter SET Combined Name[Counter] TO Defendant Name END REPEAT

FILTER COMPUTATION_VAR

Placeholder	Replace With
COMPUTATION_VAR	Any Computation variable in the template that results in a true or false value

The FILTER instruction filters out certain entries from a repeated list, based on conditions you specify.

In the following example, the script filters out all corporate entities from a list of vendors:

....

REPEAT Vendor Information

FORMAT "A, B, and C"

FILTER No Corporate Vendors

RESULT + Vendor Name

END REPEAT

In the preceding script, No Corporate Vendors is a Computation variable with the following script:

Vendor Type != "Corporation"

First, the empty quotation marks set the value of the repeat to "nothing." HotDocs then repeats the *Vendor Information DI* dialog. After the user enters all the information, HotDocs processes the responses, filters out all corporate entities, and displays a modified list in the specified format.

You can use the **AND** operator in the computation to filter out entries based on two or more conditions.

A filter can be as complicated as it needs to be, but it must result in either true or false. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a true or false value—it is not a filter. But the expression YEARS FROM(Child's Birth Date, TODAY) <= 17 can only result in true or false. It can correctly filter all children under the age of 18 from a list.

FORMAT "LIST_FORMAT"

Placeholder	Replace With
LIST_FORMAT	An example of the conjunction and punctuation desired, for example, <i>a</i> , <i>b</i> , <i>and c</i> or <i>a</i> ; <i>b</i> ; or <i>c</i> .

The FORMAT "LIST_FORMAT" instruction allows you to create a sentence-style list within a computation. (If you create the REPEAT instruction using a REPEAT field, you can specify the list format by choosing a style from the **Format** drop-down list. See Punctuate a Sentence-Style List.)

For example, perhaps you want the items in your list to appear with the final comma preceding the *and*. Your script would look like this:

"" REPEAT Education Information FORMAT "A, B, and C" RESULT + Degree Designation END REPEAT RESULT

Once a user has provided the list of educational degrees in **Education Information**, the FORMAT instruction ensures that the requisite commas are in place.

FORMAT instructions should be placed immediately after the REPEAT instructions for repeated dialogs, and before other instructions influencing the REPEAT (such as **FILTER** or **ASCEND** and **DESCEND**.

GRAY ALL; GRAY VAR; UNGRAY ALL; UNGRAY VAR

Placeholder Replace With

VAR Any variable used in a dialog

These instructions, which are used in a dialog script, control whether components in a dialog appear grayed or ungrayed, depending on answers a user enters. GRAY ALL dims all components in the dialog, while UNGRAY ALL makes all of the components active again. Likewise, GRAY VAR dims a single component, and UNGRAY VAR enables the component again.

For example, say you want certain variables in a dialog to appear based on the user's family situation. How the user answers the Multiple Choice variable, *Family Status*, controls which questions are asked:

GRAY ALL
UNGRAY Family Status
IF Family Status = "Married, with children"
UNGRAY ALL
ELSE IF Family Status = "Married, no children"
UNGRAY Spouse Name
ELSE IF Family Status = "Separated, with children"
UNGRAY Number Of Children

END IF

The initial GRAY ALL instruction dims all variables in the dialog, making them inactive. Then the *Family Status* Multiple Choice variable is immediately ungrayed so the user can choose an option. Depending on the answer to this Multiple Choice variable, some or all of the other variables are ungrayed.

In the script above, you could also gray each variable in the dialog (except the Multiple Choice variable); however, graying all of the variables and then ungraying the Multiple Choice variable immediately after saves a lot of repetitive typing.

HIDE ALL; HIDE VAR; SHOW ALL; SHOW VAR

 Placeholder
 Replace With

 VAR
 A variable used in an answer source

These instructions, which are used in a dialog script, control whether the user is able to see variables in a dialog. The HIDE ALL instruction hides all variables in the dialog, while SHOW ALL reveals the variables again. Likewise, HIDE VAR hides a single variable, and SHOW VAR reveals the variable.

In the following example, a certain estate planning template may be used for both wills and trusts. The initial dialog uses a Multiple Choice variable called *Document Type* to ask which type of document will be assembled. Then, depending on how the user answers the variable, HotDocs asks either the executor/testator names or the trustee/grantor names:

```
HIDE ALL

SHOW Document Type

IF Document Type = "Will"

SHOW Executor Name

SHOW Testator Name

ELSE IF Document Type = "Trust"

SHOW Trustee Name

SHOW Grantor Name

END IF
```

You should not HIDE or SHOW a variable in a repeated-as-spreadsheet dialog based on another variable in the same dialog. When you use HIDE or SHOW in a spreadsheet dialog, the entire column is shown or hidden. Attempting to do this may produce unexpected results.

IF EXPRESSION; ELSE IF; ELSE; END IF

Placeholder Replace With

EXPRESSION A statement that can be evaluated as true or false

You can make sections of templates or scripts conditional by using IF instructions. A conditional section will be included only if a condition you specify is *true*. The ELSE IF instruction allows two or more conditions to be included in an IF instruction. The ELSE instruction establishes a final condition for an IF instruction, specifying that if all preceding conditions are false, the following information should be included. It must be the last item of the IF instruction.

Each IF instruction or expression must end with an END IF instruction. This instruction completes a section of conditional logic. HotDocs automatically creates an END IF paired with each IF instruction. These pairs can be nested, allowing you to test several conditions before applying a single effect.

For example, in the following script, HotDocs uses an IF instruction to insert a paragraph about vacation time—but only if the new employee qualifies for paid vacation:

«IF Vacation Days»In addition, «Employee Name» shall be allowed «Number of Vacation Days:ten» for vacation time.

«Employee Name» shall also receive seven paid holidays, including New Year's Day, Memorial Day, Fourth of July, Labor Day, Thanksgiving (including the day after), and Christmas.«END IF»

More complex situations can also be handled using IF expressions. For example, in the following computation script, a single paragraph in a template may change depending on how close a project is to completion. Using the IF, ELSE IF, and ELSE instructions, the correct paragraph can be inserted:

IF Project Status = "Complete"
 "Upon finishing the project..."
ELSE IF Project Status = "In Process"
 "While working on the project..."

ELSE

"Before working on the project..."

END IF

You can use IF expressions anywhere. Operators such as **AND** and **OR** can link multiple conditions, giving the **user** greater control over the **interview**. However, creating complicated IF instructions directly in the template can make the assembly process sluggish. Consider using Computation variables or INSERT instructions instead.

In past versions of HotDocs, using END by itself instead of the complete END IF instruction was sufficient. Beginning with 11.1, HotDocs no longer processes the shortened version, and now requires the complete instruction.

Please see Introduction: Make Parts of Templates Conditional for further information on using IF instructions and expressions.

INCREMENT NUM_VAR; DECREMENT NUM_VAR

Placeholder Replace With

NUM_VAR A Number variable

The INCREMENT and DECREMENT instructions cause HotDocs to increase or decrease a number variable, usually a counter, by the value of 1.

In the following example, you want to create a list of potential employees. However, you want the list to include only those applicants with four or more years of schooling. To do this, you would use the WHILE instruction to loop through a list of applicants. You would then use the INCREMENT instruction to keep track of which repetition you are on so that the correct information can be merged into a new list.

SET Applicant Count TO 1

SET Prospect Count TO 0

WHILE ANSWERED(Applicant Name[Applicant Count])

IF Applicant Years of Schooling[Applicant Count] >= 4

INCREMENT Prospect Count

SET Prospect Name[Prospect Count] TO Applicant Name[Applicant Count]

END IF

INCREMENT Applicant Count

END WHILE

INSERT "FILENAME"

Placeholder	Replace With
FILENAME	The file name and folder path of another HotDocs template
	Include the folder path if the inserted template resides in a folder that is different from the main template.

The INSERT instruction inserts a clause, a clause library, or a template into the document currently being assembled.

Templates can be inserted from any location, as long as you specify the correct folder path information.

The INSERT instruction differs from the ASSEMBLE instruction in that HotDocs immediately processes the INSERT instruction and inserts the template, clause, or clause library into the current document. If there are variables to be answered, HotDocs presents them before finishing the interview of the main document.

To create a simple INSERT instruction, you can click the **#INSERT Field** button and insert the instruction directly in the template. However, if you want to insert a more complex instruction using conditional logic, you may need to use a Computation variable.

Often an organization's documents include sections that are used frequently (boilerplate language), such as company letterhead or attorney information blocks. Rather than recreate these parts of a document each time you automate a template, you can save just the letterhead or the information block as its own template and then use an INSERT instruction to include it in the templates that require it. For example, let's say you have a specific attorney information block you use in pleadings. You would create a template that contains only the attorney information and then use the INSERT instruction to insert it:

«IF Attorney Information Required» «INSERT "Attorney Information Block.docx"» «END IF»

In text templates, if the inserted template contains unanswered variables, HotDocs asks these variables. After the inserted template is assembled, HotDocs finishes assembling the main template. In contrast, with form templates, HotDocs appends the inserted form template to the main form. Once it finishes assembling the main form, it then assembles the inserted form template.

Inserted text templates may have formatting that differs from the original template. Headers, footers, and margins can often be controlled more easily by using word processor section breaks.

LANGUAGE CODE

Placeholder	Replace With
CODE	Any of the following language codes:

ENG (English)DEU (German)DES (Swiss German)DEA (Austrian German)FRA (French)NLD (Dutch)ESN (Spanish)ITA (Italian)PTB (Brazilian Portuguese)

This instruction tells HotDocs to format numbers and dates in a particular language.

For example, the following script allows the template to use Spanish formats:

«LANGUAGE ESN» «Start Contract Date:3 Juno 1990»

Optionally, if your date or number format requires non-U.S. thousands and decimal separators, you can specify which separators you want to use in the LANGUAGE instruction. The first character must be the thousands separator and the second character must be the decimal separator. For example:

«LANGUAGE ".," FRA»

For more detailed information on using foreign language templates, see Create a Foreign Language Template.

LIMIT NUM

Placeholder	Replace With
NUM	A whole number, Number variable, or numeric expression

The LIMIT instruction limits the number of times a dialog can be repeated. It is placed in the script of the dialog that must be limited to a specific number of repetitions.

When setting repeat limits, assign a number value or numeric expression. For example, the dialog, *Daily Schedule Information*, gathers information about the scheduled activities for each day of the work week, so it needs to be limited to five repetitions. The following would be placed in the dialog's script:

LIMIT 5

You can let the **user** change the limit each time a **document** or form is assembled by replacing the NUM placeholder with a Number variable. Make sure the Number variable gets answered before the REPEAT instruction is processed.

To control the number of viewable rows in a spreadsheet (but still allow users to enter as many answers as they need), enter the number in the **Rows to display field**. (Make sure **Spreadsheet** is selected as the **Style**.)

OMIT VAR

Placeholder	Replace With
VAR	A variable used in an answer source

The OMIT VAR instruction, which you use in a dialog script, keeps variables from appearing in the **Edit Answer Source** dialog box of an answer source.

Answer sources are associated with specific dialogs in a template. When users view the dialog during the interview, they can click a special button, which displays a list of answers that were entered during previous interview sessions. They can either select an existing set of answers or add a new set to the list.

After creating an answer source for one dialog, you can use it with other dialogs, including dialogs in other templates. When using an answer source with multiple dialogs, each variable must be represented in both the answer source file and in each dialog. If a variable that is referenced in the answer source file isn't included in the dialog (or vice-versa), answers in the answer source will get "mixed together" whenever you add, edit, or delete records.

In some situations though, it isn't always practical or relevant to show the user every variable—either in the dialog or in the **Select From Answer Source** dialog box. To accommodate this, you can use the HIDE, CONCEAL, and OMIT instructions to manipulate these variables in both the dialog and the answer source. Specifically, HIDE keeps a variable from appearing in the dialog, while CONCEAL keeps it from appearing in the answer source altogether, but still allows you to use it in the dialog. Often, you must use a combination of these instructions to achieve your desired result.

For example, say you have a dialog that shows information about a client, including how much a client owes in payments to the firm. Because information about amounts owed changes, it would not make sense to include it in the answer source. To keep it from appearing in the answer source—both in the **Select From Answer Source** dialog and in the **Edit Answer Source** dialog—you would use the OMIT instruction, like this:

OMIT Amount Owed

The variable would be asked on the dialog, however, so the user could answer it.

See Suggest an Answer Source for Dialogs for more information.

PLAY "MACRO"

Placeholder	Replace With
"MACRO"	A word processor macro that performs a certain function in your template

The PLAY "MACRO" instruction plays a word processor macro after the document is assembled and either sent to the word processor, printed, or saved.

Where you store the macro depends on which word processor you are using:

- For WordPerfect users, the macro can be stored anywhere. The PLAY instruction must include the file name of the macro, and if the macro is stored anywhere other than the default macro folder, the instruction must include a full path to the file as well.
- For Word DOT users, the macro must be stored in the template itself, in *Normal.dot*, or in any global template that is automatically loaded when you start Word.
- For Word RTF or DOCX users, the macro must be stored either in *Normal.dot*, in any global template that is automatically loaded when you start Word, or in a Word template you associate with the template through Component Manager. (See Specify a Word Template for Storing Post-Assembly Macros for details.)

Macros can be helpful in many situations, such as making sure the format of an inserted clause matches the rest of the document. For example, the text in the *Authority Clause* document may be formatted differently than the text in the main document. You could create a macro that can adjust the formatting so that it's uniform:

«INSERT "Authority Clause.docx"»

«PLAY "Standard_Format"»

To insert a PLAY instruction in a template

- 1. At the template, position the cursor in the template where you want the PLAY instruction.
- 2. If you are using Microsoft Word:
 - Click the **HotDocs** menu in the HotDocs toolbar and choose **Other Field** from the list of options. The **Other Field** dialog box appears.
 - Click the Field type drop-down button and choose PLAY.
 - In the **Macro name** field, enter the name of the macro you want to run.
 - Click **OK**. The instruction is inserted in the template.
- 3. If you are using WordPerfect:
 - Copy an existing variable or instruction field in the template.
 - Replace text between the chevrons (« ») with the PLAY instruction.

See Specify a Word Template for Storing Post-Assembly Macros for more information.

PLAY instructions are executed when you create an actual document from the assembly. This includes sending the document to the word processor, saving the document, or printing a copy of the document. If there are multiple instructions, they are processed in the order they are encountered.

QUIT

The QUIT instruction allows you to close the variable without losing the work you have done. Normally, HotDocs will not save an invalid computation. The only way to exit an invalid computation is to click **Cancel**, which erases the script. The instruction is usually placed at the beginning of an unfinished or invalid computation.

For example, perhaps you aren't sure about the specific variable names that need to be included because the variables have not yet been created. Normally, HotDocs won't allow an unfinished script to be saved. The QUIT instruction, though, makes this possible:

QUIT IF Client History = "____" SET ____ TO "Returning" ELSE SET ____ TO "New" END IF Another useful place to include a QUIT instruction is at the end of a computation, which allows you to enter "developer comments" about the computation script. You can also apply a comment block to the section of the script you don't want processed. To do this, select that section of the script and click the **Comment Block** button. (To uncomment the script, click the **Developer Block** button.)

If you use a QUIT instruction in a script, it will cause all the scripting after the instruction to lose its syntax-aware formatting. To restore this formatting once you remove the QUIT instruction, click the **Auto Format** button.

REPEAT DATABASE; END REPEAT

	Replace With
DATABASE	A database component

A REPEAT instruction gathers multiple database records and merges them into a document.

For example, the following script will bring up a record selection screen during the interview and allow you to choose multiple records to be inserted into the assembled document:

«REPEAT Client Data»

«END REPEAT»

Each REPEAT instruction must also include an END REPEAT instruction, which tells HotDocs to stop repeating the variables within the instruction.

REPEAT DATABASE instructions cannot be nested.

REPEAT DIALOG; END REPEAT

Placeholder Replace With

<u>DIALOG</u> A dialog with a repeat style specif

A REPEAT instruction gathers lists of answers and merges them into a document.

For example, the following script gathers a list of editors and inserts them into the assembled document:

```
«REPEAT Editor Information»
Editor: «Editor First Name» «Editor Last Name»
«END REPEAT»
```

Each REPEAT instruction must also include an END REPEAT instruction, which tells HotDocs to stop repeating the variables within the instruction.

This next computation script actually contains two repeated dialogs—one nested in another. It produces a list of editors as well as the titles and authors he or she is currently working with:

```
""
REPEAT Editor Information
    RESULT + "Editor: " + Editor First Name + " " + Editor Last Name + "
"
    REPEAT Book Information
        RESULT + "Book Title: " + Book Title + "
" + "Author: " + Author First Name + " " + Author Last Name + "
"
END REPEAT
END REPEAT
```

You can create up to three levels of sublists by nesting REPEAT instructions. (See Introduction: Create Lists Within a List.)

For more information about creating lists of answers, see Introduction: Include Lists in Your Documents.

Repeats can not be nested more than four levels deep.

While editing complicated sequences, you can jump from the END REPEAT instruction to its associated REPEAT instruction, or vice versa, by placing the cursor within the REPEAT or END REPEAT chevrons, then clicking the **Sequences** button.

REQUIRE ALL; REQUIRE VAR



The REQUIRE instruction requires users to answer questions in a dialog before they can advance to the next dialog in the interview.

REQUIRE instructions ensure that important information is not left out of the assembled document. For example, a certain document requires the user to enter the date when a legal filing was first made. Later in the template, this date is used to calculate a deadline for subsequent filings. Many users, however, may not take the time to look up the initial filing date, which creates problems for calculating the deadline. Using the REQUIRE instruction in a dialog script, as shown in this example script, can help resolve this problem:

REQUIRE Initial Filing Date

When a dialog script contains a REQUIRE instruction, a red mark appears on the dialog icon in the interview outline. The question in the dialog is also marked. This mark remains until the required variable is answered. If a user tries to move to another dialog, HotDocs displays an error message, then moves the cursor to the first required answer field in the dialog. Users cannot advance to the next dialog without first providing the required answers.

It may be helpful to include text in the dialog that provides users with information about which fields are mandatory and why. See Add Text to Your Dialogs for details.

You can customize the marks used in the dialog by specifying your options at HotDocs Options. See Customize the Look of the Dialog Pane.

Although current and past versions of HotDocs Developer allow the user to click Finish on an Interview Template than contains unanswered required variables or ERRORTEXT variables which have an incorrect answer in them, this behavior will be changed in the future so your templates should not rely on it.

SET VAR TO VALUE

Placeholder	Replace With
VAR	Any variable
VALUE	A value appropriate for the associated variable

For a more detailed explanation of setting variables to values, see Automatically Assign Answers to a Variable.

This instruction lets you specify a given value for a variable's answer automatically, rather than allow the user to specify an answer. With the SET instruction, you can transfer names and other values from one variable to another.

For example, if the plaintiff is the same as the client, you can use a SET instruction to automatically enter the plaintiff's name as the client's, saving the user from typing the name a second time:

IF Is Plaintiff Also Client SET Client Name TO Plaintiff Name END IF

The SET Instruction can also be used to set custom error messages in interview dialogs. Used with the ERRORTEXT expression in a dialog script, SET can be used to create conditional error messages to help minimise incorrect answers. To see examples of this method see ERRORTEXT.

Because HotDocs repeatedly processes SET instructions during the course of an interview, you must not let the user specify a different answer for a variable whose value is being SET. When HotDocs updates the interview, the user's answer will be replaced with the value from the SET instruction. To suggest an answer for the user and allow them to change it, use the DEFAULT instruction. (See Differences Between SET and DEFAULT Instructions for an explanation.)

If you are grouping two or more child (inserted) dialogs in a parent dialog, you can SET the child dialog's status to TRUE. This forces the contents of the child dialog to automatically appear in the interview outline. (See Group Child Dialogs in a Parent Dialog for details.)

To set two or more options for a Multiple Choice variable, separate each option with a vertical bar (for example, SET MC Variable TO "Option1|Option2|Option3")

Placeholder	Replace With
EXPRESSION	An expression that results in true or false
	A True/False expression can be as complicated as it needs to be, but it must result in either true or false. For example, the expression YEARS FROM(Child's Birth Date, TODAY) produces a number (the age of a person), not a true or false value—it is not a True/False expression. But the expression

WHILE EXPRESSION; END WHILE

YEARS FROM(Child's Birth Date, TODAY)> 17 can only result in true or false. It is a True/False expression.

The WHILE EXPRESSION instruction allows you to repeatedly process (or loop through) an answer or set of answers until a certain condition is met, such as a certain answer is found or a limit is reached.

Before using the WHILE instruction, you should understand the following:

- The WHILE instruction must be used in a computation or dialog script—it cannot be inserted directly into a template.
- When using the WHILE instruction inside of a REPEAT instruction, the WHILE instruction will not affect the underlying COUNTER variable associated with the REPEAT instruction. If you need to count something within a WHILE loop, you must create your own temporary counter. Additionally, to access repeated variables in a WHILE loop, you must use explicit indexing.
- Unless the instructions inside the WHILE loop include an instruction that increments the temporary counter, the WHILE instruction will repeat until the Maximum WHILE iterations limit is reached. You can specify this property at the Component File Properties dialog box. (See Change Component File Properties.) To avoid problems like this, make sure you increment the temporary counter.

In the following example, you want to create a list of signers in a will. Since the signers may include both beneficiaries and fiduciaries, you want to merge both lists into one. Because some fiduciaries may also be beneficiaries, you will want to remove any duplicate names. To loop through the list of fiduciaries, you will use the WHILE instruction:

SET Signer Count TO 0
REPEAT Beneficiary Information
INCREMENT Signer Count
SET Signer Name[Signer Count] TO Beneficiary Name
END REPEAT
REPEAT
REPEAT Fiduciary Information
SET Lookup TO 1
WHILE Lookup <= Signer Count AND Fiduciary Name != Signer Name[Lookup]
INCREMENT Lookup
END WHILE
IF Lookup > Signer Count
INCREMENT Signer Count

SET Signer Name[Signer Count] TO Fiduciary Name

END IF

END REPEAT

In the first part of this script, the *Beneficiary Information* dialog is repeated, and as answers are entered, their values are set to be used for *Signer Name* (which is the variable that will be repeated to insert all the names of the signers). Then, in the second part of the script, as the *Fiduciary Information* dialog is repeated, HotDocs uses the WHILE expression to test whether the name of the fiduciary is the same as any of the beneficiary names. If it is not, it will likewise be added to the *Signer Information* dialog. (When you insert the REPEAT instruction for the *Signer Information* dialog in the template, clear the **Ask Automatically** option at the **Dialog Editor** (**Options** tab). See Control Whether Dialogs are Asked Automatically for details.)

In the next example, you need to remove unwanted space characters from a user's account number. Here, the WHILE instruction is used to repeat an answer, character by character, so that HotDocs can check to see if there are space characters in the answer. If there are, HotDocs removes them and rewrites the answer.

SET Count Index TO 1
WHILE Count Index <= LENGTH(Account Number)
 IF MID(Account Number, Count Index, 1) = " "
 SET Account Number TO FIRST(Account Number, Count Index -1) +
 LAST(Account Number, LENGTH(Account Number) - Count Index)
 ELSE
 INCREMENT Count Index</pre>

END IF

END WHILE

This script uses a temporary counter (*Count Index*) to keep track of which character in the answer HotDocs is looking at. Any time the answer is repeated and HotDocs finds a space character, it removes it by concatenating the characters before and after the space character. HotDocs then makes sure that the new character it is now examining isn't a space character either. If it is not, HotDocs increments the temporary counter, moves to the next character, and repeats this process.

Using Expression Models

Full List of Expression Models

Expression	Description	
Enter a Date	You can use this expression to format your dates correctly as you write computations.	
<u>Enter a Number</u>	You can use the Enter a Number expression to format numbers correctly as you write computations. Numbers in computations and expressions must be in numeric form and cannot contain commas. If you enter a number that contains a comma, it will be removed from the value. Decimals must have one digit to the left of the decimal point, even if it is only a zero.	
Enter some Text	You can use the Enter some Text expression to format a text string correctly as you write computations. When used in computations and expressions, text strings must be inside quotation marks. This expression adds those quotation marks to the text string.	
Enter True or False	You can use this expression to enter a TRUE or FALSE value in a computation or expression script. True/False values must use uppercase letters.	
<u>ABSOLUTE VALUE(NUM)</u>	Using the ABSOLUTE VALUE expression, you can find the absolute value of a given number. You can calculate a negative number, but have it appear as a positive number.	
<u>AGE(DATE)</u>	The AGE(DATE) expression produces an age, in years, by calculating the number of years between the current date (as determined by your computer's system clock) and a date you provide in the computation script.	
<u>ANSWERED(DIALOG)</u>	HotDocs can determine whether a dialog has been answered using the ANSWERED expression. Even if only one variable in the dialog is answered, the expression returns a value of true.	
<u>ANSWERED(VAR)</u>	You can use the ANSWERED expression to determine whether a HotDocs variable has been assigned a value. If so, the expression receives the value of <i>true</i> .	
<u>CEILING(NUM)</u>	You can use the CEILING function to find the smallest integer that isn't smaller than the Number variable.	
<u>COUNT(DIALOG)</u>	You can find out how many sets of answers a user provides for a repeated dialog. A repeated dialog is any dialog used in a REPEAT instruction. This expression produces a number, based on each answered dialog.	
<u>COUNT(</u> MULT CHOICE VAR)	This expression counts how many options a user chooses when answering a Multiple Choice variable. The result it produces is a number.	
<u>COUNTER</u>	You can use the COUNTER expression to keep track of the current number of repetitions of a repeated dialog. Each time a user clicks the	

	Next button at a repeated dialog and provides additional information, th value of COUNTER increases.		
<u>DATE - NUM DAYS</u>	You can subtract any number of days from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.		
<u>DATE - NUM MONTHS</u>	You can subtract a certain number of months from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.		
<u>DATE - NUM YEARS</u>	You can subtract a certain number of years from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.		
<u>DATE + NUM DAYS</u>	You can add any number of days to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.		
<u>DATE + NUM MONTHS</u>	You can add any number of months to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.		
<u>DATE + NUM YEARS</u>	You can add a certain number of years to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.		
<u>DATE OF(DAY, MONTH,</u> <u>YEAR)</u>	This expression finds a date value based on day, month, and year values.		
DAY OF(DATE)	This expression returns the day portion (1 to 31) of a given date.		
DAY OF WEEK(DATE)	This expression determines on which day of the week a specific date falls and converts that value to an integer.		
<u>DAYS FROM(START,</u> <u>FINISH)</u>	This expression allows you to find the number of days between two dates.		
<u>ERRORTEXT</u>	This instruction allows you to create an error message that can appear during the interview.		
EXPONENTIAL(NUM)	An EXPONENTIAL function is a function that quickly accelerates, where the Number variable is the exponent of <i>e</i> (roughly 2.71828), and it can be used in calculating exponential growth.		
<u>FIRST(TEXT, NUM)</u>	Using this expression, you can return any number of characters starting with the first character in an answer value.		
FLOOR(NUM)	You can use the FLOOR function to find the largest integer that isn't bigger than the Number variable entered.		

<u>FORMAT(VALUE,</u> <u>"EXAMPLE")</u>	Sometimes you may need to add a date, number, or true/false value to a text value. You can do this by formatting the date, number, or true/false value as text.		
<u>FUTURE VALUE (RATE,</u> <u>TERM, PAYMENT,</u> <u>PRESENT VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.		
	If you know the rate, term, payment, present value and type of a savings scheme or loan you can use the FUTURE VALUE function to work out how much the final amount will be after interest.		
<u>INTEGER(TEXT)</u>	Sometimes you may have a text value that contains number characters, as in the case of a time of day value. The INTEGER expression allows you to convert those number characters into numeric values so you can perform calculations or compare them with other values.		
<u>LAST(TEXT, NUM)</u>	The LAST expression finds and returns a certain number of characters from the end of a text string.		
<u>LENGTH(TEXT)</u>	The LENGTH expression counts the number of characters—including spaces and punctuation—in a text value, such as a Text variable.		
<u>LOGARITHM(NUM)</u>	The LOGARITHM function will find the common logarithm of a Number variable. You can use it to find the exponent of 10 that would be needed to make the Number variable entered.		
<u>MAX(NUM, NUM)</u>	The MAX expression compares two number values and returns the greater of the two.		
<u>MID(TEXT, NUM, NUM)</u>	Like the FIRST and LAST expressions, this expression extracts a specified number of characters from within a text string.		
<u>MIN(NUM, NUM)</u>	The MIN expression compares two number values and returns the lesser of the two.		
MONTH OF(DATE)	This expression returns the month portion of a given date.		
<u>MONTHS FROM(DATE,</u> <u>DATE)</u>	The MONTHS FROM expression calculates the number of months between two given dates.		
MULT CHOICE=TEXT; MULT CHOICE!=TEXT	The MULT_CHOICE = TEXT expression returns true when the user chooses a Multiple Choice option that is equal to (=) a given text value. If it is not equal (!=), the expression returns false. The MULT_CHOICE != TEXT expression functions in the opposite way—testing instead to see if an answer is not equal to (!=) a given text value.		
<u>NATURAL LOGARITHM(</u> <u>NUM)</u>	You can use the NATURAL LOGARITHM function to find the exponent of <i>e</i> (roughly 2.71828) that would be needed to make the Number variable entered.		

NOT TRUE FALSE	You can use the NOT TRUE_FALSE expression to find out if a True/False variable is false.		
<u>OTHER(</u> <u>MULT CHOICE VAR)</u>	This expression determines whether the user has chosen the <i>Other</i> option of a Multiple Choice variable and, if so, returns the text entered in the <i>Other</i> field. It can also be used to test whether the user has selected the <i>None of the Above</i> option.		
<u>PAYMENT (RATE, TERM,</u> <u>PRESENT VALUE, FUTURE</u> <u>VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.		
	If you know the rate, term, present value, future value and type of a savings scheme or loan you can use the PAYMENT function to work out how much money needs to be paid in each payment period to reach the full amount		
POSITION(TEXT, TEXT)	The POSITION expression finds the position of a certain character or character string in a given text value. It is useful if you need to find a character you know will be in an answer but are not sure where it will appear. It returns a number value, which represents the first character.		
<u>POWER(NUM, NUM)</u>	The POWER expression generates a numeric value, based on a given exponent.		
<u>PRESENT VALUE (RATE,</u> <u>TERM, PAYMENT, FUTURE</u> <u>VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.		
	If you know the rate, term, payment, future value and type of a savings scheme or loan you can use the PRESENT VALUE function to work out how much the investment is worth currently (i.e. how much money you would need to invest over the same amount of payment periods to equal the return)		
<u>RATE (TERM, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.		
	If you know the term, payment, present value, future value and type of a savings scheme or loan you can use the RATE function to work out what the interest rate is per payment period.		
<u>REMAINDER(NUM, NUM</u>)	The REMAINDER expression returns the remainder of a division. If the denominator is a zero, HotDocs generates a divide by zero error.		

<u>REPLACE(TEXT, TEXT,</u> <u>TEXT, NUM)</u>	This expression lets you search a string of text for a given character string and replace the results with new text.		
<u>RESULT</u>	As you write computations, you often need HotDocs to acknowledge what the result would be at that point in the script. You can update this answer by using the RESULT expression.		
ROUND(NUM, NUM)	You can round a number value to a specified number of places.		
SELECTION(MULT CHOICE VAR, NUM)	This expression lets you retrieve individual options (answers) selected in a Multiple Choice variable. It returns a text value that corresponds to the defined answer (as designated by the NUM placeholder).		
<u>SPACE(TEXT, TEXT)</u>	This expression tests whether the variable is answered. If it is, it merges the answer, followed by a space character. If the variable is unanswered, it merges nothing ("").		
<u>SQUARE ROOT(NUM)</u>	Finding the square root of a number means finding an answer that, when multiplied by itself, gives the original number. You can use the SQUARE ROOT function to find the square root of a Number variable.		
<u>STRIP(TEXT, TEXT,</u> <u>TRUE FALSE, TRUE FALSE</u>)	This expression removes a specified character or characters from the beginning or end of a text answer. By default, HotDocs removes the characters from both the beginning and the end of the text. If you want to specify just one or the other, you must use the <i>TRUE_FALSE</i> parameters.		
<u>SUM(</u> COMPUTATION VAR)	Using the SUM expression, you can add repeated number values.		
<u>SUM(NUM VAR)</u>	Using the SUM expression, you can add repeated number values.		
<u>TERM (RATE, PAYMENT,</u> <u>PRESENT VALUE, FUTURE</u> <u>VALUE, TYPE)</u>	The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.		
	If you know the rate, payment, present value, future value and type of a savings scheme or loan you can use the TERM function to work out how many payment periods are required to reach the full amount.		
TEXT CONTAINS TEXT	The TEXT CONTAINS TEXT expression determines whether the first text value contains the same text as the second value. If it does, it returns the value of <i>true</i> .		
TEXT ENDS WITH TEXT	The TEXT ENDS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value ends with the second TEXT value.		

TEXT STARTS WITH TEXT	The TEXT STARTS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value begins with the second TEXT value.			
TODAY	This expression returns the current date, according to your computer's system clock.			
TRIM(TEXT)	You can use the TRIM function to remove any white space characters from the beginning and end of a Text variable.			
TRUNCATE(NUM, NUM)	You can truncate a decimal number a specified number of places after a decimal point.			
<u>UNANSWERED</u>	This expression removes an assigned value from a variable. It is used most often with the SET VAR TO VALUE instruction.			
UNION(MULT_CHOICE, MULT_CHOICE)	This expression creates a single list of all unique options (answers) that have been selected across two Multiple Choice variables.			
VALUE(VAR, EXPRESSION)	This expression returns a default value for the variable type if the variable is unanswered. If the variable is answered, the value is the answer the user specifies			
<u>YEAR OF(DATE)</u>	You can use this expression model to find the year portion of a given date.			
YEARS FROM(DATE, DATE)	This expression calculates the number of years between two given dates.			
ZERO(NUM VAR)	This expression returns the value of <i>zero</i> only if a Number variable is unanswered. If the Number variable is answered, the value is the answer the user specifies.			

Enter a Date

You can use this expression to format your dates correctly as you write computations.

To use the expression, drag the **Enter a Date** expression into the **Script** or **Expression** field. The **Enter a Date** dialog appears where you can type a date into the field, or you can use the pop-up calendar to select a date. When you click OK, HotDocs inserts the date at the cursor position in the correct format, for example, *3 JUN 1990*.

Enter a Number

You can use the **Enter a Number** expression to format numbers correctly as you write computations. Numbers in computations and expressions must be in numeric form and cannot contain commas. If you enter a number that contains a comma, it will be removed from the value. Decimals must have one digit to the left of the decimal point, even if it is only a zero.

To use the expression, drag the **Enter a Number** expression into the **Script** or **Expression** field. The **Enter a Number** dialog box appears. Enter a number and click **OK**. The correctly formatted number is inserted at the cursor position.

Enter some Text

You can use the **Enter some Text** expression to format a text string correctly as you write computations. When used in computations and expressions, text strings must be inside quotation marks. This expression adds those quotation marks to the text string.

To use the expression, drag the **Enter some Text** expression into the **Script** or **Expression** field. The **Enter some Text** dialog box appears. Enter some text and click **OK**. HotDocs inserts that text—in quotation marks—at the cursor position.

Enter True or False

You can use this expression to enter a TRUE or FALSE value in a computation or expression script. True/False values must use uppercase letters.

To use the expression, drag the **Enter True or False** expression into the **Script** or **Expression** field. The **Enter True or False** dialog box appears. Click either **True** or **False**. The keyword you choose is inserted in all capital letters at the cursor position.

ABSOLUTE VALUE(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value for which you want the absolute value returned. Can be a Number variable or a fixed number value.

Returns a **Number** value

Using the ABSOLUTE VALUE expression, you can find the absolute value of a given number. You can calculate a negative number, but have it appear as a positive number.

For example, you may need to send a notice to a client about an account balance. Not knowing whether it will be a positive or negative balance, you would create the computation *Absolute Value of Final Balance*, which would return a positive expression, regardless. The computation script is:

```
ABSOLUTE VALUE( Final Balance )
```

Once you have created the computation, you can use an IF/ELSE expression to merge the desired text with the correct value in the document:

```
«IF Final Balance < 0»
Your account is $«Absolute Value of Final Balance» overdrawn.
«ELSE»
You have $«Absolute Value of Final Balance» in your account.
«END IF»</pre>
```

AGE(DATE)

Placeholder	Tooltip	Replace With
DATE	d: Date	A date value, which you want to check against the current date. This can be a Date variable or a fixed date value.

Returns a **Number** value

The AGE(DATE) expression produces an age, in years, by calculating the number of years between the current date (as determined by your computer's system clock) and a date you provide in the computation script.

For example, the following script determines the age of the user based on his or her birth date:

AGE(Birth Date)

In the following conditional script, HotDocs determines whether the client is under the age of 18. If so, the *Parent-Guardian Information* dialog is asked. If the client is over the age of 18, no dialogs are asked.

```
IF AGE( Birth Date ) < 18
ASK Parent-Guardian Information
END IF
```

ANSWERED(DIALOG)

Placeholder	Tooltip	Replace With
DIALOG	d: Dialog	A dialog name

Returns a **True/False** value

HotDocs can determine whether a dialog has been answered using the ANSWERED expression. Even if only one variable in the dialog is answered, the expression returns a value of true.

Let's suppose you have a dialog that gathers information about a decedent (*Decedent Information*). From within this dialog, a user could open an inserted dialog that asks questions about the decedent's assets (*Decedent's Assets*). Later in the template, you could ask additional questions based on whether these dialogs have been answered:

IF ANSWERED(Decedent Information) AND ANSWERED(Decedent's Assets)

ASK Asset Distribution Information

END IF

ANSWERED(VAR)



Returns a True/False value

You can use the ANSWERED expression to determine whether a HotDocs variable has been assigned a value. If so, the expression receives the value of *true*.

If you are testing whether a user has ANSWERED a variable, you must make sure the variable is presented to the user using a custom dialog. Using the ANSWERED(VAR) expression alone will not automatically force HotDocs to display the variable for the user.

For example, you may place a variable for the second line of an address (*Client Address 2*) in a custom dialog; however, not all users will provide information for that variable. In the template text, you can surround *Client Address 2* with an IF expression that merges that variable into the document—only if the user answers it:

«Client Name»
«Client Address 1»
«IF ANSWERED(Client Address 2)»
«Client Address 2»
«END IF»
«Client City», «Client State» «Client ZIP»

In this script, the ANSWERED expression is used with an IF expression to insert or remove *Client Address 2*, based on whether the user has provided that information. (Without it, the default unanswered text, ****Client Address 2****, would be inserted in the assembled document.)

Even if a user chooses not to answer a variable that has been tested using the ANSWERED expression, HotDocs will still warn that it is unanswered. If you don't want a warning to appear, clear **Warn when unanswered** at the **Advanced** tab of the Variable Editor. (Control How HotDocs Processes a Variable.)

CEILING(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

You can use the CEILING function to find the smallest integer that isn't smaller than the Number variable, for example:

CEILING(84.2) = 85

COUNT(DIALOG)

Placeholder	Tooltip	Replace With
DIALOG	dlg: Dialog	A dialog name

Returns a **Number** value

You can find out how many sets of answers a user provides for a repeated dialog. A repeated dialog is any dialog used in a REPEAT instruction. This expression produces a number, based on each answered dialog.

The following example determines if there is more than one fiduciary. If there is, certain prefixes and plural abbreviations are added to the fiduciary title so the paragraph is structured correctly.

I appoint «REPEAT Fiduciary Information:a, b, and c»«Fiduciary Name:LIKE THIS»«END REPEAT» as «IF COUNT(Fiduciary Information) > 1»Co-«Fiduciary Title»s«ELSE»«Fiduciary Title»«END IF».

This expression uses the COUNT instruction to determine if *Fiduciary Information* is answered more than once. If it is, the *Co*- prefix is inserted before *Fiduciary Title*, and the plural *s* is inserted at the end of the variable. For example, in the assembled document, the answer may be inserted as *Co-Executors* or *Co-Personal Representatives*.

The difference between COUNT and COUNTER is that COUNT counts the number of repetitions in a list, while COUNTER gives you the number of the current repetition.

COUNT(MULT_CHOICE_VAR)

Placeholder	Tooltip	Replace With
MULT_CHOICE_VAR	m: Multiple Choice Variable	A Multiple Choice variable with the Select option set to All That Apply

Returns a **Number** value

This expression counts how many options a user chooses when answering a Multiple Choice variable. The result it produces is a number.

For example, suppose the user wants to generate a list of cities in which an author plans to make appearances. There might be one—or many—depending on the schedule. HotDocs can merge the correct term—*city* or *cities*—into the document once it knows how many cities were selected from the Multiple Choice variable:

```
IF COUNT( Publicity Tour City ) = 1
    "city"
ELSE IF COUNT( Publicity Tour City ) > 1
    "cities"
END IF
```

COUNTER

Returns a **Number** value

You can use the COUNTER expression to keep track of the current number of repetitions of a repeated dialog. Each time a user clicks the **Next** button at a repeated dialog and provides additional information, the value of COUNTER increases.

For example, a user may want to create a word processor table that contains a numbered list of clients:

«REPEAT Client Information»
«COUNTER». «Client Name»
«END REPEAT»

In the example above, *Client Information* repeats the *Client Name* variable. Each time a user enters a different client, COUNTER is incremented and merged into the assembled document. For example:

- 1. John TeNgaio
- 2. Erica Nees
- 3. Lisa Alvey
- 4. Jonathan Rainwater

HotDocs also uses COUNTER as a way to compare two incrementing number values. For example, perhaps you want to list the last child named in a repeated dialog:

```
REPEAT Children Information
ASCEND Child Birth Date
IF COUNTER = COUNT( Children Information )
RESULT + "The youngest child is " + Child Name
END IF
END REPEAT
```

In this computation script, HotDocs first sets the value of the computation to nothing. It then processes the REPEAT instruction, sorting the children based on their birth dates. It uses COUNTER to determine when the last answer in the dialog is given (by comparing it to the COUNT of the dialog), and then merges the name of the youngest child in the list into the document.

The difference between COUNT and COUNTER is that COUNT counts the number of repetitions in a list, while COUNTER gives you the number of the current repetition.

DATE - NUM DAYS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number value

Returns a **Date** value

You can subtract any number of days from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation calculates the date the books need to arrive at the warehouse before they can be shipped:

Shipping Date - 14 DAYS

DATE - NUM MONTHS

Placeholder	Replace With	
DATE	A date value, such as a Date variable	
NUM	A number value	

Returns a **Date** value

You can subtract a certain number of months from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

In the following example, HotDocs subtracts four months from the *Shipping Date* and inserts the new date:

Shipping Date - 4 MONTHS

DATE - NUM YEARS

Placeholder	Replace With	
DATE	A date value, such as a Date variable	
NUM	A number value	

Returns a **Date** value

You can subtract a certain number of years from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation subtracts two years from Marriage Date:

Marriage Date - 2 YEARS

DATE + NUM DAYS

Placeholder Replace With

DATEA date value, such as a Date variableNUMA number value

Returns a **Date** value

You can add any number of days to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation script adds 90 days to the Date variable, *Purchase Date*:

Purchase Date + 90 DAYS

DATE + NUM MONTHS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number value

Returns a **Date** value

You can add any number of months to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

For example, this computation determines what the date will be six months from the date the document is assembled:

TODAY + 6 MONTHS

DATE + NUM YEARS

Placeholder	Replace With	
DATE	A date value, such as a Date variable	
NUM	A number value	

Returns a **Date** value

You can add a certain number of years to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.

In this script, HotDocs adds 30 years to the date the loan originated:

Loan Origination Date + 30 YEARS

DATE OF(NUM, NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	day: Number	A number value representing the day of the month (from 1 to 31)
NUM	month: Number	A number value representing the month (from 1 to 12)
NUM	year: Number	A number value representing the year (should be four digits)

Returns a **Date** value

This expression finds a date value based on day, month, and year values.

You can use this expression to compare a date the user gives with another date, such as a cut-off date for when an employee had to be hired to qualify for a yearly bonus:

IF Hire Date <= DATE OF (15, 8, YEAR OF(TODAY))

Employee Name + " qualifies for the annual bonus."

END IF

HotDocs uses an IF instruction to compare *Hire Date* with *August 15* of the current year. If the comparison returns a *true* value, the employee qualifies for the bonus.

Suppose, in this next example, a new employee qualifies for a benefits package on the first day of the second month of employment—regardless of what day the employee was hired during the first month of employment. You can calculate that date with the DATE OF expression:

DATE OF (1, MONTH OF(Hire Date + 1 MONTHS), YEAR OF(Hire Date + 1 MONTHS))

The first parameter in the expression, 1, tells HotDocs to specify the first day of the month. The second parameter identifies the month of the hire date and adds one month. The third parameter determines the year of the hire date (plus one month).

DAY OF(DATE)

Placeholder	Tooltip	Replace With
DATE	d: Date	A date value

Returns a **Number** value

This expression returns the day portion (1 to 31) of a given date.

The following computation is used to determine when a new employee can begin accruing vacation days. If the employee is hired on the first day of the month, he or she immediately begins accruing time off. Otherwise, he or she begins accruing at the beginning of the next month:

IF DAY OF (Hire Date) = 1 SET Start Accruing Date TO Hire Date

ELSE

SET Start Accruing Date TO DATE OF(1, MONTH OF(Hire Date + 1 MONTHS), YEAR OF(Hire Date + 1 MONTHS))

END IF

DAY OF WEEK(DATE)



Returns a **Number** value

This expression determines on which day of the week a specific date falls and converts that value to an integer.

HotDocs Developer Help File These integers are as follows: Sunday = 1 Monday = 2 Tuesday = 3 Wednesday = 4 Thursday = 5 Friday = 6 Saturday = 7

For example, perhaps you want to determine whether a payment due date falls on a Saturday or Sunday. If it does, HotDocs moves the payment due date to the following Monday. The following script shows how this works:

```
IF DAY OF WEEK( Payment Date ) = 7
Payment Date + 2 DAYS
ELSE IF DAY OF WEEK( Payment Date ) = 1
Payment Date + 1 DAYS
ELSE
Payment Date
END IF
```

DAYS FROM(DATE, DATE)

Placeholder	Tooltip	Replace With
DATE	start: Date	A date value, such as a Date variable
DATE	finish: Date	A date value, such as a Date variable. (These can be in any order.)

Returns a **Number** value

This expression allows you to find the number of days between two dates.

In the following example, a buyer has 60 days to make a payment on an account balance. If the buyer has miss ed the payment deadline, HotDocs merges a warning into the document:

IF DAYS FROM(Purchase Date, TODAY) > 60

"Your account is past due."

ELSE

"Your account is current. Thank you."

END IF

This example uses an IF/ELSE IF expression to determine the text that must be inserted.

ERRORTEXT

Returns a **Text** value

ERRORTEXT can only be used in a dialog script.

Use the ERRORTEXT expression to create an error message that appears when the error condition you specify occurs during an interview. If a user provides information that triggers the error message, the user must correct the error before HotDocs enables access to the next dialog.

If you make the ERRORTEXT expression conditional, the error message appears both as a popup, and at the top of the dialog, accompanied by a red error symbol. If you do not make the ERRORTEXT expression conditional, as soon as the user clicks the dialog, the error message appears and displays continuously at the top of the dialog. In this case, there is no accompanying popup.

You can make the error conditional by adding an IF instruction to the script. You could, for example, warn users when they type in a forbidden answer.

Say you work for a delivery company that does not deliver to certain areas of the United States, you could use an ERRORTEXT expression to warn customers when they type the name of a state that is outside of your delivery area. Your dialog script might look something like this:

IF state = "Hawaii" Or state = "Alaska"

SET ERRORTEXT TO "Unfortunately we do not deliver to «state»"

END IF
If a customer types either "Hawaii" or "Alaska," the error message appears and they must change their answer before continuing with the interview.

HotDocs blocks the user from continuing to the next dialog, because HotDocs handles an ERRORTEXT expression like a required variable. You can use this feature to generate error messages if the user leaves a necessary question unanswered in an interview.

In a similar scenario to the example given above, suppose your delivery company needs to make sure that customers type the name of a country when filling out their address. Your script, to alert the user when they leave the country field blank, might look something like this:

IF !ANSWERED (Country)

SET ERRORTEXT TO "Country is a required field"

END IF

If a customer does not type the name of a country before trying to move to the next dialog, the error message appears. An Input Error icon also appears in the interview outline until the user provides an acceptable value for this variable.

In your ERRORTEXT message, it may be helpful to include information for users on how they can meet the conditions needed to complete the dialog and continue with the interview.

Although current and past versions of HotDocs Developer allow the user to click Finish on an Interview Template than contains unanswered required variables or ERRORTEXT variables which have an incorrect answer in them, this behavior will be changed in the future so your templates should not rely on it.

EXPONENTIAL(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

An EXPONENTIAL function is a function that quickly accelerates, where the Number variable is the exponent of *e* (roughly 2.71828), and it can be used in calculating exponential growth.

Example:

```
EXPONENTIAL( NUM ) = e NUM
```

EXPONENTIAL(3) = e^{3}

EXPONENTIAL(3) = 20.0855369231877

FIRST(TEXT, NUM)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable, from which the specified number of characters will be returned. This can be a fixed text value, inside quotation marks.
NUM	count: Number	A number value, such as a Number variable or a fixed number value. It specifies the number of characters you want returned.

Returns a **Text** value

Using this expression, you can return any number of characters starting with the first character in an answer value.

The following computation looks at the client's first, middle, and last names and returns only the first character from each of these variables. When merged together, these characters create the client's initials:

```
FIRST( Client First Name, 1 ) + FIRST( Client Middle Name, 1 ) + FIRST( Client
Last Name, 1 )
```

In the following example, the first four characters of a client's last name are merged with a case number to create a file number.

FIRST(Client Last Name, 4) + Case Number

FLOOR(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

You can use the FLOOR function to find the largest integer that isn't bigger than the Number variable entered, for example:

FLOOR(23.7) = 23

FORMAT(VALUE, "EXAMPLE")

Placeholder	Tooltip	Replace With
VALUE	v: Value	Any Number, Date, or True/False variable
"EXAMPLE"	example: Text	A format example (in quotation marks) you want used with the value. Must be in a format HotDocs can recognize.

Returns a **Text** value

Sometimes you may need to add a date, number, or true/false value to a text value. You can do this by formatting the date, number, or true/false value as text.

For example, perhaps you want to create a list of items with their associated monetary values. Because these two values are different in nature, they cannot be added together without first representing the number value as a text value:

....

REPEAT Purchase Information

RESULT + Item Name + ", " + FORMAT(Item Amount, "\$9,999.00") + "

...

END REPEAT

In this script, HotDocs first sets the computation value to nothing. Then HotDocs repeats *Purchase information* and then places the answers for both *Item Name* and *Item Amount* (which is formatted to appear as a text value) in the same text string, separated by a comma. If *Purchase Information* is answered more than once, HotDocs manually inserts a hard return (as shown before the END REPEAT) to create a column of amounts.

FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Term	term: Number	A number variable representing the total number of payment periods.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Present Value (optional)	[present value: Number]	A number variable representing the current value of the future amount. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, term, payment, present value and type of a savings scheme or loan you can use the FUTURE VALUE function to work out how much the final amount will be after interest.

For example, if you have an annual rate of 7.5% over 3 years and you pay \$100 a month then you can work out the future value by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate)

Term: 36 (amount of months in 3 years) Payment: -100 Present Value: 0 Type: 0

Future Value = 4,023.1381682

This function works in much same way as the FV(rate, nper, pmt, pv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for FV.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- TERM (Rate, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)

INTEGER(TEXT)

Placeholder	Tooltip	Replace With
TEXT	t: Text	A text value, such as a Text variable

Returns a **Number** value

Sometimes you may have a text value that contains number characters, as in the case of a time of day value. The INTEGER expression allows you to convert those number characters into numeric values so you can perform calculations or compare them with other values.

INTEGER searches the beginning of a text string for number characters and converts those it finds to numeric values. When it encounters a non-number character (such as a letter or punctuation mark) it stops processing the instruction.

For example, if you tried to find the integer of the word *cat*, the INTEGER expression would return 0 (zero) since there are no number characters in *cat*. However, if you used INTEGER on the text value 12:30, it

would return the number value 12 since those characters are numbers. (As explained earlier, it stops processing when it reaches a punctuation mark, which in this case is a colon.)

One of the main uses for the INTEGER expression is to compare time values. In the following computation, HotDocs is attempting to determine if a given time value falls after 5:30 P.M. Because time values are text values, the Text variable, *Call Time*, must first be converted to an integer before it can be used in the comparison:

```
IF Call Time CONTAINS "p"
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
":" ), 2 ) ) /60 ) > 5.5
ELSE
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
":" ), 2 ) ) /60 ) > 17.5
```

END IF

In this script, *Call Time* is a Text variable with a 24-hour or 12-hour time pattern (99:99 or 99:99 A.M.). HotDocs first determines if *Call Time* is in the afternoon (P.M.). If it is, the script uses the INTEGER expression to convert all the digit characters up to the first non-digit character (the colon) into a numeric value. This number represents the hours portion of the total time. Using the MID expression to locate the two digit characters after the colon, it also converts these characters into an integer and divides the value by 60. This number represents the minutes portion of the total time. These two numbers are added together, and if the result is greater than 5.5 (the equivalent of 5:30), the result is *true*. If the result is not greater than 5.5, the result is *false*.

The second portion of the script (after the ELSE expression) performs the same functions on a nonafternoon time value—that is, one that is either in 24-hour format or in the morning (A.M.).

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable, from which the specified number of characters will be returned. This can be a fixed text value, inside quotation marks.
NUM	count: Number	A number value, such as a Number variable or a fixed number value. It specifies the number of characters you want returned.

LAST(TEXT, NUM)

Returns a **Text** value

The LAST expression finds and returns a certain number of characters from the end of a text string.

For example, the following text computation returns the last four digits of a Social Security number:

LAST(Social Security Number, 4)

In the next example, the user wants to make the answer to *Item Type* plural. Using the LAST expression, HotDocs checks to see if the last letter in the value is a specific letter. If so, HotDocs inserts the correct plural suffix.

```
IF LAST( Item Type, 1 ) = "s" OR LAST( Item Type, 1 ) = "z"
    Item Type + "es"
ELSE
    Item Type + "s"
```

END IF

This example script does not take into consideration words that end in "y" or "x" or any other letter that would cause yet a different result. It is only shown here in its most basic form to demonstrate how the LAST expression functions.

LENGTH(TEXT)

Placeholder	Tooltip	Replace With
ТЕХТ	t: Text	A text value, such as a Text variable. HotDocs counts the characters in this value and assigns a numeric value.

Returns a **Number** value

The LENGTH expression counts the number of characters—including spaces and punctuation—in a text value, such as a Text variable.

For example, let's say you want a user to add a descriptive paragraph about the novel he or she has just reviewed. If the description is brief, you'd like to keep it in the same paragraph as the lead-in sentence. However, if the user has much to say about the novel, you would like to start a new paragraph:

```
«IF LENGTH( Plot Description ) <= 150»
```

```
The novel's plot description is as follows: «Plot Description»
«ELSE»
The novel's plot description is as follows:
«Plot Description»
«END IF»
```

LOGARITHM(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

The LOGARITHM function will find the common logarithm of a Number variable. You can use it to find the exponent of 10 that would be needed to make the Number variable entered, for example:

```
NUM = 10 <sup>LOGARITHM( NUM )</sup>
1000= 10 <sup>3</sup>
LOGARITHM ( 1000 ) = 3
```

MAX(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n1: Number	A number value, such as a Number variable

NUM

n2: Number

A number value, such as a Number variable (These values can be in any order.)

Returns a **Number** value

The MAX expression compares two number values and returns the greater of the two.

In this example, HotDocs returns the value of either the *Monthly Salary* or the *Monthly Expenses*, depending on which is the greater value:

MAX(Monthly Salary, Monthly Expenses)

MID(TEXT, NUM, NUM)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	A text value, such as a Text variable, from which the specified number of characters will be returned. This can be a fixed text value, inside quotation marks.
NUM	start at: Number	A number value, such as a Number variable or a number you type. This number specifies where HotDocs starts "returning" characters.
NUM	count: Number	A number value, such as a Number variable or a number you type. It specifies the number of characters you want returned.

Returns a **Text** value

Like the FIRST and LAST expressions, this expression extracts a specified number of characters from within a text string.

For example, a form template may require that the text variable, *Telephone Number* (with the telephone number pattern), be split into three pre-formatted fields—the area code, the prefix, and the number. In the following example, three different computations would be scripted and inserted into the different fields as follows:

MID(Telephone Number, 2, 3)
MID(Telephone Number, 7, 3)
LAST(Telephone Number, 4)

The first computation, which you would place in the area code field, tells HotDocs to include three characters, starting with the second character (this takes into consideration the opening parenthesis.)

The second computation (the prefix field) starts at the seventh character (again, taking into account the parentheses and space characters between the area code and the prefix), and inserts the next three characters.

The third expression, which uses the LAST expression, returns the last four digits of the phone number by counting backwards from the last character.

MIN(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n1: Number	A number value, such as a Number variable
NUM	n2: Number	A number value, such as a Number variable. (These values can be in any order.)

Returns a **Number** value

The MIN expression compares two number values and returns the lesser of the two.

In the following number computation, the expression compares the two values, *Shipping Costs* and *Labor Costs*, and returns the lesser of the two values:

MIN(Shipping Costs, Labor Costs)

MONTH OF(DATE)

Placeholder	Tooltip	Replace With
DATE	d: Date	A date value, such as a Date variable

Returns a **Number** value

This expression returns the month portion of a given date.

For example, in the following script, a new employee has temporary status until the end of his or her third month with the company. HotDocs uses the MONTH OF expression to specify which month that is:

DATE OF(1, MONTH OF (Hire Date + 3 MONTHS), YEAR OF (Hire Date + 3 MONTHS)) - 1 DAYS

In this example, HotDocs finds the first day of the fourth month of employment. HotDocs then subtracts one day to reveal the last day of the third month—either the 28th, 29th, 30th or 31st—depending on the month. It then inserts the new date into the document.

MONTHS FROM(DATE, DATE)

Placeholder	Tooltip	Replace With
DATE	start: Date	A date value, such as a Date variable
DATE	finish: Date	A date value, such as a Date variable. (These values can be in any order.)

Returns a **Number** value

The MONTHS FROM expression calculates the number of months between two given dates.

The following example finds the number of months between the judgment date and today—in months:

MONTHS FROM(Judgment Date, TODAY)

MULT_CHOICE=TEXT; MULT_CHOICE!=TEXT

Placeholder	Replace With
MULT_CHOICE	A Multiple Choice variable
TEXT	A text value that is either equal to (=) or not equal to (!=) one of the options in the given Multiple Choice variable, inside quotation marks.

Returns a **Number** value

The MULT_CHOICE = TEXT expression returns true when the user chooses a Multiple Choice option that is equal to (=) a given text value. If it is not equal (!=), the expression returns false. The MULT_CHOICE !=

TEXT expression functions in the opposite way—testing instead to see if an answer is not equal to (!=) a given text value.

In the following True/False expression, if the user chooses *Credit Card* as the payment method, HotDocs asks the user for the credit card information:

IF Method of Payment = "Credit Card"

ASK Credit Card Information

END IF

In the next expression, if *Credit Card* is not chosen as a method of payment, HotDocs inserts a template which can gather alternate payment information about the user:

IF Method of Payment != "Credit Card"
 INSERT "Alt payment method.docx"
END IF

When writing this script, you can use the auto-complete functionality to access your list of Multiple Choice options. Specifically, press **Ctrl+Spacebar** (after you enter the operator) to display a list of the different Multiple Choice options. See **Use the Script Editor** for full details on using auto-complete as you write scripts.

NATURAL LOGARITHM(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

You can use the NATURAL LOGARITHM function to find the exponent of *e* (roughly 2.71828) that would be needed to make the Number variable entered, for example:

NUM = e NATURAL LOGARITHM(NUM)

 $1000 = e^{6.90775527898214}$

```
HotDocs Developer Help File
```

```
NATURAL LOGARITHM( 1000 ) = 6.90775527898214
```

NOT TRUE_FALSE

Placeholder	Replace With
TRUE_FALSE	A true/false value, such as a variable or expression that results in true or false.

Returns a **Number** value

You can use the NOT TRUE_FALSE expression to find out if a True/False variable is false.

In the following script, HotDocs asks whether the user is a United States citizen. If the user is not, HotDocs asks the user for Visa information:

«IF NOT US Citizen» «ASK Visa Information» «END IF»

OTHER(MULT_CHOICE_VAR)

Placeholder	Tooltip	Replace With
MULT_CHOICE	m: Multiple Choice Variable	A Multiple Choice variable that has either the Other option specified, or the None of the Above option specified

Returns a **Text** value

This expression determines whether the user has chosen the *Other* option of a Multiple Choice variable and, if so, returns the text entered in the *Other* field. It can also be used to test whether the user has selected the *None of the Above* option.

For example, a user is given a list of lending agents from which to choose. If the user doesn't see the correct name on the list, he or she can select *Other* and specify the correct name. HotDocs then asks for the city in which the lending agent operates, as shown in the following script:

IF Lending Agent = OTHER(Lending Agent)
ASK Lending Agent City
END IF

In this next example, you want to create a list of company representatives. However, if the user doesn't select a company representative, you want the text *No representative selected* merged.

""
"IF OTHER (Company Representative) = "None of the Above"
"No representative selected"
ELSE FORMAT (Company Representative, "a, b, and c")
END IF

PAYMENT (Rate, Term, Present Value, Future Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Term	term: Number	A number variable representing the total number of payment periods.
Present Value	present value: Number	A number variable representing the current value of the future amount.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, term, present value, future value and type of a savings scheme or loan you can use the PAYMENT function to work out how much money needs to be paid in each payment period to reach the full amount.

For example, if you have an annual rate of 7.5% over 3 years and you know that the future value is \$4,023.1381682 then you can work out the monthly payment by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate)
Term: 36 (amount of months in 3 years)
Present Value: 0
Future Value: 4,023.1381682
Type: 0

Payment = -100

This function works in much same way as the PMT(rate, nper, pv, fv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for PMT.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- TERM (Rate, Payment, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

POSITION(TEXT, TEXT, TRUE_FALSE)

Placeholder	Tooltip	Replace With
TEXT	in: Text	A text value, such as a Text variable

ТЕХТ	search for: Text	The character or character string for which you want to search
TRUE_FALSE	[last instance: True/False]	A True/False variable where False indicates HotDocs should start looking for the search value at the left (beginning) of the
(optional)		text and True indicates HotDocs should start from the right (end). Optional. Default is <i>FALSE</i> .

Returns a **Number** value

The POSITION expression finds the position of a certain character or character string in a given text value. It is useful if you need to find a character you know will be in an answer but are not sure where it will appear. It returns a number value, which represents the first character.

The following script finds the hyphen in the variable, *Case Number*, and returns a number value, representing its numeric position in that given text string.

POSITION(Case Number, "-")

In the next example, the POSITION expression is used as part of a larger computation to test whether a given time falls after 5:30 P.M. POSITION locates the colon (:) in the time value so HotDocs can process the text before and after the colon to find the correct result:

```
IF Call Time CONTAINS "p"
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
    ":"), 2 ) ) /60 ) > 5.5
```

ELSE

```
INTEGER( Call Time ) + ( INTEGER( MID( Call Time, 1 + POSITION( Call Time,
":" ), 2 ) ) /60 ) > 17.5
```

END IF

This script first determines if the value of *Call Time* is in the afternoon (P.M.). If it is, the script uses the POSITION expression to locate the first non-digit character (the colon) so the INTEGER expression can convert all of the digit characters leading up to it into a numeric value.

Once identified, this number represents the hours portion of the total time. Using the MID expression to locate the two digit characters after the colon, it also converts these characters into an integer and divides the value by 60. This number represents the minutes portion of the total time. The hours and minutes are added together, and if the result is greater than 5.5 (the equivalent of 5:30), the result is true. If the result is not greater than 5.5, the result is false.

The second portion of the script (after the ELSE expression) performs the same functions on a nonafternoon time value—that is, one that is either in 24-hour format or in the morning (A.M.).

POWER(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	base: Number	A number value, such as a Number variable, to be raised to a power
NUM	exponent: Number	The exponent (or number that indicates the operation of repeated multiplication)

Returns a **Number** value

The POWER expression generates a numeric value, based on a given exponent.

For example, say you want to calculate the future value of an investment:

POWER((1 + Annual Rate of Return), Number of Years) * Amount Invested

HotDocs adds 1 to Annual Rate of Return and then raises it to the power of Number of Years. It then multiplies that number by Amount Invested.

PRESENT VALUE (Rate, Term, Payment, Future Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Term	term: Number	A number variable representing the total number of payment periods.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, term, payment, future value and type of a savings scheme or loan you can use the PRESENT VALUE function to work out how much the investment is worth currently (i.e. how much money you would need to invest over the same amount of payment periods to equal the return)

For example, if you have an annual rate of 7.5% over 3 years, paying \$100 a month and you know that the future value is \$4,023.1381682 then you can work out the present value by using the following figures:

Rate: 0.00625 (0.075/12 to find the monthly rate)
Term: 36 (amount of months in 3 years)
Payment: -100
Future Value: 4,023.1381682
Type: 0

Present Value = 0

This function works in much same way as the PV(rate, nper, pmt, fv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for PV.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- TERM (Rate, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

RATE (Term, Payment, Present Value, Future Value, Type)

Placeholder Tooltip	Replace With	
---------------------	--------------	--

Term	term: Number	A number variable representing the total number of payment periods.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Present Value	present value: Number	A number variable representing the current value of the future amount.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the term, payment, present value, future value and type of a savings scheme or loan you can use the RATE function to work out what the interest rate is per payment period.

For example, if you are paying \$100 a month over 3 years and you know that the future value is \$4,023.1381682 then you can work out the rate by using the following figures:

Term: 36 (amount of months in 3 years) Payment: -100 Present Value: 0 Future Value: 4,023.1381682 Type: 0

Rate = 0.00625 (7.5% per annum)

This function works in much same way as the RATE(nper, pmt, pv, fv, type, guess) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for **RATE**.

See also:

- TERM (Rate, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

REMAINDER(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	dividend: Number	A number value, such as a Number variable, to be divided (a numerator)
NUM	divisor: Number	A number value, such as a Number variable, by which to divide (a denominator)

Returns a **Number** value

The REMAINDER expression returns the remainder of a division. If the denominator is a zero, HotDocs generates a divide by zero error.

In this basic example, HotDocs divides 10 by 3. The remainder of that division is 1:

REMAINDER(10, 3)

In this next example, a user enters a time value in number format (such as 6 hours). However, using the REMAINDER expression (as well as TRUNCATE and ROUND), HotDocs causes the value to appear in *hours:minutes* format:

```
FORMAT( TRUNCATE ( Number of Hours, 0 ), "9" ) + ":" +
FORMAT( ROUND ( 60 * REMAINDER( Number of Hours, 1 ), 0 ), "09" )
```

This script takes the value of *Number of Hours*, which may have a decimal value, and truncates it to a whole number. Then, using the REMAINDER expression, *Number of Hours* is divided by 1 and the remainder of the division is multiplied by 60 (as in 60 minutes). HotDocs then rounds that value and brings these two values together in a string, separated by a colon. The value is then formatted correctly.

REPLACE(TEXT, TEXT, TEXT, NUM)

Placeholder	Tooltip	Replace With
ТЕХТ	in: Text	The name of a Text variable or the string of characters you need to search
ТЕХТ	search for: Text	The text for which you are searching, which may include the following special characters:
		\\ backslash character
		\h non-breaking hyphen
		\l line break
		\p paragraph mark
		\s non-breaking space
		\t tab
TEXT	replace with: Text	The text you want to use as a replacement, which may include the following special characters:
		\\ backslash character
		\h non-breaking hyphen
		\l line break
		\p paragraph mark
		\s non-breaking space
		\t tab
NUM	[replace count: Number]	Controls the number of times the character is replaced
(optional)		For example, if no number is specified, all found instances will be replaced; however, if you include a 1 as a parameter, only the first found instance will be replaced. Optional. Default is <i>< all ></i> .

Returns a **Text** value

This expression lets you search a string of text for a given character string and replace the results with new text.

For example, perhaps you want to take the information in an address block (which will most likely appear as separate lines) and display it as a single line, with each "part" separated by a comma. The following script removes all line breaks (\l) from the Text variable, *Multi-Line Address*, replaces them with a comma and space, and then SETs that result to the Text variable *Single-Line Address*:

SET Single-Line Address TO REPLACE(Multi-Line Address, "\1", ", ")

RESULT

Returns a Date, Number, Text or True/False value

As you write computations, you often need HotDocs to acknowledge what the result would be at that point in the script. You can update this answer by using the RESULT expression.

For example, let's suppose you are creating a list of editors. You want to combine *Editor First Name TE* and *Editor Last Name TE* as well as the literal text *Editor*:. You must use RESULT to force HotDocs to acknowledge the result of the computation before you add the next item to the text string:

""
REPEAT Editor Information
RESULT + Editor First Name + " " + Editor Last Name + ", Editor" + "
"
END REPEAT

In this computation, the RESULT expression returns the value of *Editor First Name* and adds it to the value of *Editor Last Name*. The RESULT expression updates the list each time a new editor is added to the list. If no RESULT expression were used, HotDocs would merge just the first name entered in the list.

ROUND(NUM, NUM)

Placeholder

Tooltip

Replace With

NUM	n: Number	A number value, such as a Number variable, to be rounded
NUM (optional)	[to places: Number]	A number value that indicates the number of places (0-7) to the right of the decimal point after which the number will be rounded. Optional. Default is <i>0</i> .

Returns a **Number** value

You can round a number value to a specified number of places.

The following example looks at the first two digits after the decimal point and rounds the value based on the third digit. (The resulting value is *5.93*.)

ROUND(5.9274, 2)

SELECTION(MULT_CHOICE_VAR, NUM)

Placeholder	Tooltip	Replace With
MULT_CHOICE	m: Multiple Choice Variable	A Multiple Choice variable
NUM	index: Number	A number value that indicates which selected option to return

Returns a **Text** value

This expression lets you retrieve individual options (answers) selected in a Multiple Choice variable. It returns a text value that corresponds to the defined answer (as designated by the NUM placeholder).

In the following example, you want to generate a list of employees that have various different work projects they need to complete. Multiple employees may work on one individual project. Once you have this list, you want to generate a work list report for each employee on the list.

To accomplish this, you first repeat a dialog (*Employee List*) that asks which employees are supposed to work on a given assignment (using the Multiple Choice variable, *Employee Names*). As HotDocs repeats this list, the UNION expression adds each selected, original name from each repetition to a new Multiple Choice variable, *Unique List*.

Once all of the unique answers have been added to *Unique List*, HotDocs then uses the SELECTION expression to retrieve each individual answer from *Unique List*. The result of the script merges these names in a report.

```
ERASE Unique List
```

REPEAT Employee List

SET Unique List TO UNION(Unique List, Employee Names)

END REPEAT

ERASE Project Participant

SET Index TO 1

```
WHILE SELECTION(Unique List, Index ) != ""
```

SET Project Participant[Index] TO SELECTION(Unique List, Index)

INCREMENT Index

END WHILE

SPACE(TEXT, TEXT)

Placeholder	Tooltip	Replace With
ТЕХТ	t: Text	A Text variable or other expression that produces a text result
TEXT (optional)	[append: Text]	A character or text string that can be used in place of the space character. You can include the following special characters in your search:
		\\ backslash character
		\h non-breaking hyphen
		\I line break
		\p paragraph mark
		\s non-breaking space
		\t tab
		Optional. Default is [space].

Returns a **Text** value

This expression tests whether the variable is answered. If it is, it merges the answer, followed by a space character. If the variable is unanswered, it merges nothing ("").

For example, perhaps you need to merge a client's full name. Some clients, however, do not have a middle name. You can create a script that includes this middle name (if it's provided), followed by a space. If no middle name is given, nothing will be merged.

Client First Name + " " + SPACE(Client Middle Name) + Client Last Name

Sometimes you may want to merge a character other than a space. The second optional parameter for this expression allows you to specify what this character should be.

In the following example, the script uses the SPACE expression to determine if each of the variables in the address block are answered. If so, it merges the answer to the variable, followed by a line break character (rather than a space character). This merges each "part" of the address on its own line.

```
SPACE(Address Line 1, "\l") +
SPACE(Address Line 2, "\l") +
SPACE(City, ",") + SPACE(State) + SPACE(Zip Code)
```

SQUARE ROOT(NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable.

Returns a **Number** value

Finding the square root of a number means finding an answer that, when multiplied by itself, gives the original number. You can use the SQUARE ROOT function to find the square root of a Number variable, for example:

SQUARE ROOT(64) = 8

STRIP(TEXT, TEXT, TRUE_FALSE, TRUE_FALSE)

Placeholder	Tooltip	Replace With
ТЕХТ	t: Text	The name of a Text variable, or the string of characters you need to search
ТЕХТ	characters: Text	The character or string of characters for which you want to search. You can strip any alphanumeric characters from a text string, including the following special characters: (Make sure you include the backslash.)
		\\ backslash character
		\h non-breaking hyphen
		\I line break
		\p paragraph mark
		\s non-breaking space
		\t tab
TRUE_FALSE (optional)	[at beginning: True/False]	The value of TRUE if characters should be stripped from the beginning of the text. Optional. Default is <i>TRUE</i> .
TRUE_FALSE (optional)	[at end: True/False]	The value of TRUE if characters should be stripped from the end of the text. Optional. Default is <i>TRUE</i> .

Returns a **Text** value

This expression removes a specified character or characters from the beginning or end of a text answer. By default, HotDocs removes the characters from both the beginning and the end of the text. If you want to specify just one or the other, you must use the *TRUE_FALSE* parameters.

For example, perhaps you want to remove punctuation or space characters from the end of an answer (because the punctuation is already included in the document text). The following script will help you accomplish this:

```
SET Product Description TO STRIP(Product Description, " !.,?", FALSE, TRUE)
```

Because the FALSE and TRUE parameters are used, HotDocs strips the characters from the end of the answer (TRUE) and not the beginning (FALSE).

TRIM(TEXT) is equivalent to STRIP(TEXT, "\t\", TRUE, TRUE). See TRIM(TEXT) for more information.

SUM(COMPUTATION_VAR)

Placeholder	Tooltip	Replace With
COMPUTATION_VAR	c: Computation Variable	A repeated Computation variable

Returns a **Number** value

Using the SUM(COMPUTATION_VAR) expression, you can add computation values that have been repeated.

For example, let's say you have a repeated dialog that contains three variables—*Item Name, Item Amount*, and *Item Quantity*. For each line item, you create a fourth variable, a computation called *Total Amount* that multiplies *Item Amount* by the number of items the user purchases (or *Item Quantity*). You can then add all of the *Total Amount* values and receive one sum total:

SUM(Total Amount)

In order for SUM(COMPUTATION_VAR) to work properly in a template, the script that actually calculates the sum (for example, the REPEAT instruction) must be processed before the variable containing the SUM expression is asked.

Also, when using DOCX or RTF templates, headers or footers are always processed before the rest of the template text. This means that if you are using SUM(COMPUTATION_VAR) in a header or footer, the script that calculates the sum must be asked in the header or footer.

SUM(NUM_VAR)

Placeholder Tooltip Replace With

NUM_VAR *n: Number* A repeated Number variable

Returns a **Number** value

Using the SUM expression, you can add repeated number values.

In this computation script, HotDocs totals the values of the repeated Number variable Monthly Payment:

```
SUM( Monthly Payment )
```

In the next example, HotDocs then takes the total monthly payments the user is making and compares it to the user's monthly income. If the monthly payments are greater than 36 percent of the monthly income, the loan application is rejected:

```
SUM( Monthly Payment ) > ( Monthly Income * 0.36 )
```

TERM (Rate, Payment, Present Value, Future Value, Type)

Placeholder	Tooltip	Replace With
Rate	rate: Number	A number variable representing the interest rate per payment period.
Payment	payment: Number	A number variable representing the payment made in each payment period.
Present Value	present value: Number	A number variable representing the current value of the future amount.
Future Value (optional)	[future value: Number]	A number variable representing the value after the last payment period. Optional. Default is <i>0</i> .
Type (optional)	[type: Number]	A number variable in which you can enter the number 0 or 1. 0 meaning payments are due at the end of the payment period and 1 meaning payments are due at the beginning of the payment period. Optional. Default is <i>0</i> .

Returns a **Number** value

The HotDocs financial expressions can be used in templates that work with calculating savings, loans or investment values.

If you know the rate, payment, present value, future value and type of a savings scheme or loan you can use the TERM function to work out how many payment periods are required to reach the full amount.

For example, if you have an annual rate of 7.5% paying \$100 a month and you know that the future value is \$4,023.1381682 then you can work out how many months you will need to reach that amount by using the following figures:

```
Rate: 0.00625 (0.075/12 to find the monthly rate)
Payment: -100
Present Value: 0
Future Value: 4,023.1381682
Type: 0
```

```
Term = 36
```

This function works in much same way as the NPER(rate, pmt, pv, fv, type) function does in Mircosoft Excel. For more information on that function you can see the Microsoft Office help for NPER.

See also:

- RATE (Term, Payment, Present Value, Future Value, Type)
- PAYMENT (Rate, Term, Present Value, Future Value, Type)
- PRESENT VALUE (Rate, Term, Payment, Future Value, Type)
- FUTURE VALUE (Rate, Term, Payment, Present Value, Type)

TEXT CONTAINS TEXT

Placeholder	Replace With
TEXT	A text value, such as a Text variable
ТЕХТ	A text value, such as a Text variable. Any text you type must be in quotation marks.

Returns a **True/False** value

The TEXT CONTAINS TEXT expression determines whether the first text value contains the same text as the second value. If it does, it returns the value of *true*.

In this example, some states in the United States are officially recognized as "commonwealth" states. In the following script, HotDocs examines the answer the user provides for the variable *State Name* to see if the user has listed one of these states. If so, HotDocs attaches the correct designation to the merged answer:

IF "massachusetts virginia kentucky pennsylvania" CONTAINS State Name

"The Commonwealth of «State Name»"

ELSE

"The State of «State Name»"

END IF

At first glance, this computation may seem backward. You may think you would test *State Name* to see if it contains any of the commonwealths listed in the text string (for example, IF State Name CONTAINS "massachusetts virginia kentucky pennsylvania"). However, if you used that method, the answer the user assigns to *State Name TE* would have to contain everything between the quotation marks—"massachusetts virginia kentucky pennsylvania"—and you would never produce a *true* statement. Of course, you could test *State Name* against each individual commonwealth (for example, IF State Name CONTAINS "massachusetts" OR IF State Name CONTAINS "virginia" and so forth), but by "switching" the values for the placeholders and placing all of the commonwealths in one text string, you eliminate a lot of repetitive typing.

Also see **TEXT STARTS WITH TEXT** and **TEXT ENDS WITH TEXT** for similar functions.

TEXT ENDS WITH TEXT

Placeholder	Replace With
TEXT	A text value, such as a Text variable
ТЕХТ	A text value, such as a Text variable. Any text you type must be in quotation marks.

Returns a **True/False** value

The TEXT ENDS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value ends with the second TEXT value.

See also TEXT STARTS WITH TEXT.

TEXT STARTS WITH TEXT

Placeholder	Replace With
ТЕХТ	A text value, such as a Text variable
ТЕХТ	A text value, such as a Text variable. Any text you type must be in quotation marks.

Returns a **True/False** value

The TEXT STARTS WITH TEXT expression is used in a similar way to the TEXT CONTAINS TEXT expression except that HotDocs will specifically check if the first TEXT value begins with the second TEXT value.

See also TEXT ENDS WITH TEXT.

TODAY

Returns a **Date** value

This expression returns the current date, according to your computer's system clock.

For example, this computation script figures out the number of days between the day the user purchased the item and today's date (or the date the user assembles the document.)

DAYS FROM(Purchase Date, TODAY)

You also can create a Date variable called TODAY. See Create a Date Variable that Inserts the Current Date.

TRIM(TEXT)

Placeholder	Tooltip	Replace With
ТЕХТ	t: Text	A text value, such as a Text variable.

Returns a **Text** value

You can use the TRIM function to remove any white space characters from the beginning and end of a Text variable, for example:

The client, Mary Rogers, has approved...

would be returned as:

The client, Mary Rogers, has approved...

TRIM(TEXT) is equivalent to STRIP(TEXT, " t^{1} , TRUE, TRUE). See STRIP(TEXT, TEXT, TRUE_FALSE, TRUE_FALSE) for more information.

TRUNCATE(NUM, NUM)

Placeholder	Tooltip	Replace With
NUM	n: Number	A number value, such as a Number variable
NUM (optional)	[to places: Number]	A number value specifying the number of places (0-7) to the right of the decimal point at which to truncate the number. Optional. Default is <i>0</i> .

Returns a **Number** value

You can truncate a decimal number a specified number of places after a decimal point.

For example, the following script truncates the number 5.9375 to include only the first two digits after the decimal point. The truncated value is 5.93:

TRUNCATE(5.9375, 2)

In the next example, however, a Computation variable tests if the value of *Rent Amount* includes cents. The variable is then formatted to eliminate the text *AND NO CENTS* from being merged when the variable contains only a whole number:

```
IF Rent Amount = TRUNCATE( Rent Amount, 0 )
FORMAT( Rent Amount, "NINE DOLLARS" )
ELSE
FORMAT( Rent Amount, "NINE DOLLARS AND TWELVE CENTS" )
```

END IF

Specifically, this computation compares the actual value of *Rent Amount* against its truncated value. If the values are equal, HotDocs formats the answer to exclude the text, *AND TWELVE CENTS*. Otherwise it includes the text in the format.

The difference between the TRUNCATE and the ROUND expressions is that TRUNCATE simply "cuts off" a number at a specified digit, while ROUND actually rounds a number up or down, based on where you want the number rounded.

UNANSWERED

This expression removes an assigned value from a variable. It is used most often with the SET VAR TO VALUE instruction:

SET Property Value TO UNANSWERED

If you use this instruction, do not allow the user to change the answer by asking the variable in the interview. Because HotDocs can potentially reprocess the interview several times, the answer the user enters will always be replaced with the UNANSWERED value. Only use this value if you want a variable to be unanswered in the assembled document.

Do not confuse the value of UNANSWERED with that of NOT ANSWERED. UNANSWERED is an actual value assigned to a variable, while NOT ANSWERED (or !ANSWERED) is used to determine whether a value has been assigned.

UNION(MULT_CHOICE, MULT_CHOICE)

Placeholder	Tooltip	Replace With
MULT_CHOICE	m1: Multiple Choice Variable	A Multiple Choice variable
MULT_CHOICE	m2: Multiple Choice Variable	A Multiple Choice variable

Returns a Multiple Choice value

This expression creates a single list of all unique options (answers) that have been selected across two Multiple Choice variables.

In the following example, you want to generate a list of employees that have various different work projects they need to complete. Multiple employees may work on one individual project. Once you have this list, you want to generate a work list report for each employee on the list.

To accomplish this, you first repeat a dialog (*Employee List*) that asks which employees are supposed to work on a given assignment (using the Multiple Choice variable, *Employee Names*). As HotDocs repeats this list, the UNION expression adds each selected, original name from each repetition to a new Multiple Choice variable, *Unique List*.

Once all of the unique answers have been added to *Unique List*, HotDocs then uses the SELECTION expression to retrieve each individual answer from *Unique List*. The result of the script merges these names in a report.

ERASE Unique List REPEAT Employee List SET Unique List TO UNION(Unique List, Employee Names) END REPEAT ERASE Project Participant SET Index TO 1 WHILE SELECTION(Unique List, Index) != "" SET Project Participant[Index] TO SELECTION(Unique List, Index) INCREMENT Index

END WHILE

VALUE(VAR, EXPRESSION)

Placeholder	Tooltip	Replace With
VAR	v: Variable	The name of a variable
EXPRESSION (optional)	[default: Value]	The specific value you want to assign to the variable if the user leaves it unanswered. Otherwise, HotDocs will use one of the following default values:
		"" for Text variables
		0 for Number variables
		1 JAN 1800 for Date variables
		FALSE for True/False variables
		"" for Multiple Choice variables

Returns a Date, Number, Text or True/False value

This expression returns a default value for the variable type if the variable is unanswered. If the variable is answered, the value is the answer the user specifies.

For example, users will sometimes purposely not answer a question in the interview. However, all variables must be answered or the script will fail. Using VALUE assigns an answer (albeit a default one) so that the script or expression can be processed correctly and still return a value.

In the following example, suppose you need to calculate a sum. However, one of the variables in the calculation, *Sales Tax*, may not be required in order to produce the result. Since the variable may be left unanswered, you can use the VALUE expression to assign a default value of 0 to the variable so that the script can be processed correctly.

The optional *EXPRESSION* placeholder allows you specify the value that should be returned if the user leaves the question unanswered. Otherwise, HotDocs will use the default value, shown in the table, above.

Services Cost + Product Cost + VALUE(Sales Tax)

YEAR OF(DATE)



Returns a **Number** value

You can use this expression model to find the year portion of a given date.

For example, in the following script, a new employee has temporary status until the end of his or her third month with the company. HotDocs uses the YEAR OF expression to specify the year:

DATE OF(1, MONTH OF(Hire Date + 3 MONTHS), YEAR OF(Hire Date + 3 MONTHS)) - 1 DAYS

In this example, HotDocs finds the first day of the fourth month of employment. HotDocs then subtracts one day to reveal the last day of the third month—either the 28th, 29th, 30th or 31st—depending on the month. It then merges the new date into the document.

YEARS FROM(DATE, DATE)

Placeholder	Tooltip	Replace With
DATE	start: Date	A date value, such as a Date variable
DATE	finish: Date	A date value, such as a Date variable (These values can be in any order.)

Returns a **Number** value

This expression calculates the number of years between two given dates.

In the following example, the YEARS FROM expression finds the client's age at the time of the hearing:

YEARS FROM(Hearing Date, Birth Date)

In this next example, the expression determines whether an employee has worked at the company for 25 years or more. If the employee has, HotDocs merges a paragraph into the document with details of a retirement party:

«IF YEARS FROM(Hire Date, Retirement Date) >= 25» We would like to honor you at a retirement party on «Party Date» at «Party Time» at «Party Location».
«END IF»

ZERO(NUM_VAR)

Placeholder	Tooltip	Replace With
NUM	n: Number	A Number variable

Returns a **Number** value

This expression returns the value of *zero* only if a Number variable is unanswered. If the Number variable is answered, the value is the answer the user specifies.

For example, users will sometimes leave a Number variable blank instead of entering *0*. If the Number variable is then used in a Computation variable, because it is unanswered, the computation will not be processed. The ZERO expression ensures the variable is assigned a value so the computation can return a value.

In this example, the ZERO expression is used in the Computation variable *Invoice Total* to return 0 if the Number variable *Shipping Charge* is unanswered.

```
Total Price + ZERO( Shipping Charge )
```

You can suggest default answers for unanswered variables of other types. See VALUE(VAR, EXPRESSION) for details.

Setting Values for Variables

Automatically Assign Answers to a Variable

Variables normally get their values from the answers users enter during an interview, but sometimes you may want to assign an answer to a variable instead of allowing the user to specify the answer.

For example, a document might include the address of the client and, in another place, the address of the client's spouse. Once the client's address has been entered by the user, you could use a SET instruction to automatically fill in the same address for the spouse, since it will be the same.

SET instructions are inserted using a Computation variable, which you can insert in the template where you want the SET instruction to take effect.

When you set a variable to a value, you should clear the **Save in answer file** option for the variable **(Advanced** tab of the variable editor). That way, when HotDocs builds the interview, the answer can be set without prompting the user to save the answer file. (A changed answer file warning like this could be puzzling to users, especially if users don't add or change any answers in the interview.)

To set a variable to a value

- 1. Create a Computation variable. (See Customize a Computation Variable.)
- 2. From the Instruction models list, drag the SET VAR TO VALUE instruction into the Script field.
- 3. Drag a variable you want to receive the new answer from the **Variables** list onto the **VAR** placeholder.
- 4. Replace the VALUE placeholder with the answer you are assigning to the variable. You can do this by dragging a variable name from the Variables list, by dragging another instruction or expression into the Script field, or by entering an actual value. (See Values in SET Instructions for more details about these different types of values.)
- 5. Click **OK** after the placeholders have been replaced with actual values.

You should never SET a variable's value and then cause the variable to be asked later in the template. If you do this, the value assigned by the SET instruction will always overwrite users' answers. If you want to suggest an answer for users but allow them to change it, use the DEFAULT instruction. (See DEFAULT VAR TO VALUE instruction model and Differences Between SET and DEFAULT Instructions.)

To set two or more options for a Multiple Choice variable, separate each option with a vertical bar (for example, SET MC Variable TO "Option1|Option2|Option3").

If you intend to SET values for all the variables in a repeated dialog, you should make sure the **Trim endmost iterations whose answers are grayed or hidden** option is turned off for that dialog. Otherwise the dialog's COUNT and/or the number of repetitions that show up in the interview outline may be unpredictable.

Values in SET Instructions

You can set a variable to three kinds of values—a fixed value, another variable's value, and a value created by an expression. Be sure the variable and the value are the same type—set a Text variable to a text value, a Number variable to a number value, and so forth.

Fixed Values

A fixed value is a number, a date, a string of text, or *TRUE* or *FALSE* that is typed into the SET instruction. For example:

SET Receiving Office TO "Ridgefield" SET Maximum Weight TO 5000 SET Client is Married TO TRUE SET Property Items TO "Furniture|Automobiles"

Be sure to follow the rules for entering fixed values—for example, text must be in quotation marks and numbers cannot include commas. See Understand the HotDocs Scripting Language for an explanation.

Variables

You can set a variable to the value of another variable. That way, the user supplies the answer. For example:

SET Receiving Office TO Customer City

Expressions

You can use an expression to produce the value for a SET instruction. For example:

```
SET Total Weight of Shipment TO (Weight of One Unit * Number of Units) + (Weight of Pallet * (Number of Units / 12))
```

Use SET Instructions to Help Write Computations

Sometimes a SET instruction makes a computation easier to write. When there's a long expression that is used in the computation more than once, set a variable to the expression and then use the variable wherever you need the expression after that in the computation.

For example, you might create a variable like *Total Weight of Shipment* just for use within the computation—it might not be used anywhere else in the template. In the following script, the SET instruction (the first instruction in the script) assigns the value of an entire expression to the variable. You can then use the variable elsewhere in the computation instead of writing the expression again:

SET Total Weight of Shipment TO (Weight of One Unit * Number of Units) + (Weight of Pallet * (Number of Units / 12))

IF Total Weight of Shipment > 10000

"Peoria Warehouse"

ELSE IF Total Weight of Shipment > 5000

"South Chicago Warehouse"

ELSE IF Total Weight of Shipment > 200

"Chicago Office Warehouse"

ELSE

"Local Office"

END IF

Differences Between SET and DEFAULT Instructions

Both the SET instruction and the DEFAULT instruction assign answers to variables. However, there are differences between the way the two instructions operate in a script. These differences are outlined in the following table:

SET Instruction	DEFAULT Instruction
Sets the value of a variable each time the instruction is processed, even if the variable is already answered. You should never SET a variable's value and then cause the variable to be asked later in the template. If you do this, the value assigned by the SET instruction will always overwrite the user's answer. If you want to suggest an answer for a user but give them the chance to change the answer, use the DEFAULT instruction	Sets the value of a variable the first time the instruction is processed and if the variable is unanswered. (If the variable is already answered, the DEFAULT instruction has no effect.)
When the instruction is processed, causes HotDocs to prompt the user to save the answer file. (To avoid warnings like this, clear the Save in answer file option for the variable (Advanced tab of the Variable Editor).)	When the instruction is processed, does not cause HotDocs to prompt the user to save the answer file.

If you intend to SET values for all the variables in a repeated dialog, you should make sure the **Trim endmost iterations whose answers are grayed or hidden** option is turned off for that dialog. Otherwise the dialog's COUNT and/or the number of repetitions that show up in the interview outline may be unpredictable.

Linking Templates to a Database

Introduction: Link Templates to a Database

Each time you use a text or form template to assemble a document, HotDocs prompts you for the information the template requires. Once you have answered the questions, HotDocs merges your answers and carries out your instructions to produce the finished, assembled document.

There are three ways to provide HotDocs with the information it needs during document assembly. You can use any (or a combination) of these methods:

- 1. Enter the information manually.
- 2. Retrieve the information from an answer file (or answer source, such as an Outlook Contacts list).
- 3. Retrieve the information from a database using a database component.

The last option, which requires you to create a database component, allows you to use answers from a database during document assembly. That way, information from a company data store or other database-driven program, such as some case managers, can also be used to assemble a HotDocs document.

Before you can retrieve answers from a database, you must first set up a connection to your database from within a HotDocs template. Then, once the connection is specified, you can link variables in your template to fields in a database table. Finally, you can specify several options that control the scope of the records from which the user chooses and how this data appears to them during the interview.

For a more technical explanation of how database components works, see Understand Database Connectivity.

Use Supported Databases

Supported Databases

The following is a list of database programs and servers (and their corresponding OLE DB providers) with which the HotDocs has been tested and is certified to work. These databases include:

- Microsoft Access (Microsoft.Jet.OLEDB)
- Microsoft SQL Server (SQLOLEDB)

- Oracle (It is strongly recommended that you use the Oracle Provider for OLE DB (OraOLEDB) instead of the Microsoft OLE DB Provider for Oracle (MSDAORA).)
- Any ODBC level 1-compliant database (using the Microsoft OLE DB provider for ODBC (MSDASQL))

HotDocs may also function with other OLE DB providers, as long as the following conditions are met: 1) The OLE DB provider must accept plain-text command strings, 2) the OLE DB provider must return rowsets (ADO Recordsets) as the result of commands—not records and/or streams, and 3) the database must reside in a location accessible by HotDocs. OLE DB providers that expose data in individual Rows and/or Fields (but not Rowsets) will not work. Please keep in mind that these are recommended guidelines and meeting these guidelines does not necessarily mean your provider will work with HotDocs.

Providers.ini

If you are able to establish a connection to a database using an OLE DB provider, but are having trouble accessing certain data types (most noticeably dates), you can use a file called Providers.ini, which supplies HotDocs with information specific to each OLE DB provider it accesses. This file augments information not provided by the database and supersedes HotDocs' own defaults or information which may have been provided by the database.

The Providers.ini file was installed and saved to the HotDocs program folder at the time HotDocs was installed. You can edit its contents using any text editor. It contains a header explaining the file's syntax and the possible options which can be specified there. HotDocs queries each OLE DB provider it uses for as much of this information as possible, but it is possible the information is not supplied. If this is the case, you must create an entry for the OLE DB provider in Providers.ini. (Examples for doing this are included in the file itself.)

HotDocs needs to know certain details about how to build database commands for a given OLE DB provider. Among these are:

- How the database expects dates to be formatted.
- How dates are delimited.
- How database literals (table and column names) that contain spaces are delimited.
- How schema and/or catalog names are separated from table names.

Understand Database Connectivity

HotDocs uses ActiveX Data Objects (or ADO) as the primary means of communicating with databases. (HotDocs 5 used ODBC.) ADO allows greater flexibility in the types of databases you can access and gives you greater control over how you extract information from a database.

In order to connect HotDocs to a database using an ADO connection, you must link your template to the database using an OLE DB provider. An OLE DB provider acts as a "translator" between HotDocs and the database, telling the two how to interpret the information that is being passed from the database to the template. Many database programs and/or clients include their own native OLE DB providers, but if you are using a database for which no provider is available, you can use an ODBC driver for your database and the OLE DB provider for ODBC. An ODBC-type connection allows you to connect to the database using an ODBC DSN (data source name), which associates a database driver with the database itself. This help file includes instructions for setting up both types of connections—see Create a Database Component Using ADO or Create a Database Component Using ODBC. (For additional information on ADO connections and data link properties, please either refer to the Windows system help or refer to the help files installed with your particular database program.)

Once the connection to the database is made, HotDocs must know what information to retrieve from the database. You do this by linking variables in the template to fields in a database table or view. Optionally, you can set up filtering and sorting options and designate other custom options that change how records are retrieved from the database.

To perform these various tasks, you can use the features available in the Database Editor, which is a component-editing dialog box like those used to edit other components. However, if you have experience writing database commands, you can specify an option that will allow you to define these filters, etc. using your own command text. Be aware, however, that writing custom commands requires you to have experience using the query language (SQL or other database query language) required by your specific database program. Because of the large number of databases (and potential language variants) little effort is made in this help file to teach this. This help file does contain, however, some general guidelines to help you write your custom commands. (See Use a Custom Command to Retrieve Data from a Database.)

If you are converting database components created in HotDocs 5 to the latest HotDocs format, you should consider changing the type of connection you are using to use an OLE DB provider for the specific database program, rather than the OLE DB provider for ODBC. For information, see Convert ODBC-based Connections to ADO.

At a Glance: The Database Editor

🛄 New Da	tabase - Da	tabase Editor	- Demo	Editor List.	mp	? 🗙
Properties	Field Map	Sort & Filter	Options	Resource	Used In Notes	
Componen	t <u>n</u> ame:		_			
Title:						
			B			
Prompt:						
			0			
	G					
C <u>o</u> nnectior	n properties	are defined by	:	Co <u>m</u> mand ty	pe:	
an ADO connection string 🕞 🔹 Simple 🔋 🔹						
Connection string:						
						^ ∎
			•			-
Allow database write-back						
		odate		ŌK	Cancel	Sa <u>v</u> e

After creating a **New Database Component** at the **Component Manager**, HotDocs will open the **Database Editor** at the Properties tab. From here you can set up the connection string and enter the required information. To set more database options such as field mapping and filters use the tabs at the top of the window.

In the first text field $\frac{A}{A}$ you can enter a name for your new database component, in the second field $\frac{B}{A}$ you can enter a title for your database, and in the third field $\frac{C}{A}$ you can enter a prompt to appear with your database.

Below these text fields are two drop down menus. In the menu on the left **D** you can select what the connection properties will be defined by. You can choose from the following three options:

- an ADO connection string
- a Microsoft Data Link (.uld) file
- an OBDC Data Source Name

ADO connections require you to specify an OLE DB provider while Microsoft Data Link connections require the connection information to be stored in a separate file. Finally, an ODBC connection requires you to specify an OLE DB provider for ODBC. (ODBC connections require you to create and use a data source for your database application.)

In the menu on the right gou can choose the command type for your database: simple or User-defined (SQL or custom). If you choose a simple connection, HotDocs will build the connection string for you based on the options you select when you specify the OLE DB provider. If you choose a user-defined connection, you must specify all the information yourself in the Connection string box.

onnection properties are defined by:	Command type:	
a Microsoft Data Link (.udl) file 🛛 🔻	Simple	•
licrosoft Data Link file:	51 ⁻	

If you select **a Microsoft Data Link (.uld) file** from the **Connection Properties** menu then the options below change (see image above). You can either type in the location of the Microsoft Data Link file straight in the field or use the **Open** button to navigate to the file.

Connection properties are defined by:	Command type:	
an ODBC data source	Simple	-
ODBC Data Source Name (DSN):	64 - 851	
		-

If you select **an OBDC Data Source Name** from the **Connection Properties** menu then the options below change (see image above). You can then select an option from the **ODBC Data Source Name** menu: dBASE files, Excel files or MS Access Database.

At the bottom of every set of options is a check box to allow database write-back. Tick this box to save any changes you make in HotDocs back to the original database you have linked to.

To learn more about using the database editor follow the links below:

- Understand Database Connectivity
- Create a Database Component Using ADO
- Create a Database Component Using ODBC
- Save ADO Connection Information in a Separate File
- Convert ODBC-based Connections to ADO

- Connect to Microsoft Excel using the Database Component
- Edit a Database Component
- Link Variables to Database Fields
- Remember Selected Records
- Save Changed Answers Back to the Database
- Change Database Component Properties
- Limit the Number of Database Records from Which a User Chooses
- Sort Records in a Database Table
- Select Related Database Records

At a Glance: The Data Link Properties dialog box

🚺 Data L	ink Properties				
Provider	Connection Advanced All				
Select	Select the data you want to connect to:				
OLE DB Provider(s)					
Microsoft Jet 4.0 OLE DB Provider Microsoft OLE DB Provider for Analysis Services 10.0 Microsoft OLE DB Provider for Indexing Service					
Mic	rosoft OLE DB Provider for ODBC Drivers				
Microsoft OLE DB Provider for Oracle Microsoft OLE DB Provider for Search Microsoft OLE DB Provider for SQL Server Microsoft OLE DB Simple Provider MSDataShape					
	Next >>				
	OK Cancel Help				

After opening the **Database Editor** from the **Component Manager** and clicking the **Edit Connection String** button you will see the **Data Link Properties** dialog box. From here you can select the **OLE DB**

Provider you would like to use for the **Database Component**. Once you have selected, clicking the **Next** button will take you to a set of options specific to each choice.

Clicking the **Help** button on this page will take you to the **Microsoft Data Link Help** file where you can find more detailed help instructions.

To learn more about setting the options for data link properties follow the links below:

- Create a Database Component Using ADO
- Save ADO Connection Information in a Separate File
- Convert ODBC-based Connections to ADO
- Specify a User Name and Password for a Database

Create a Database Component Using ADO

In order for programs such as HotDocs to communicate with certain database programs, the database must understand the command HotDocs issues to retrieve data from it. Likewise, HotDocs must understand how to work with the data that is retrieved from the database. One way this is accomplished is by using a data presentation layer known as ActiveX Data Objects, or ADO. ADO allows HotDocs to communicate with a wide range of sources using OLE DB providers for each specific database program. The OLE DB provider acts as a translator between the specific database and HotDocs—it reads information from the database and tells HotDocs how to interpret the data so HotDocs can use it.

That said, in order to connect with a database, you must tell HotDocs to which database you want it to connect. This requires you to create at least one database component for your template, which is where information about the connection is stored—including information about the OLE DB provider you are using, the name of the database, and any security information that is required to access data in the database. Once you have created the database component and assigned a name to it, you must define a connection string, which is what HotDocs will use to connect to the database. This string contains information about the OLE DB provider, the location of the actual database file or server, and any other information needed to establish the connection.

To create a database component

- At the template you want to connect to the database, click the **Component Manager** button. The **Component Manager** window appears.
- 2. Click the **Components** drop-down button and select **Databases** from the list. The list changes to show only database components.
- 3. Click the **New Component** button. HotDocs displays the **Database Editor**.
- 4. Type a name for the component in the **Component name** field.

- 5. Optionally, perform the following tasks:
 - In the **Title** field, enter a title for the database component. The title will be used in the interview outline and dialog title bar.
 - In the **Prompt** field, type the information about the database table you want the user to see. This information will appear above the database table in the assembly window. (See Create a Prompt for a Variable.)
 - Click the **Resource** tab and provide a resource option. A resource helps the user know what type of answer to select. (See Add Resource Information to a Variable or Dialog.)
- 6. At the **Properties** tab of the **Database Editor**, click the **Connection properties are defined by** drop-down button and choose **an ADO connection string**.
- 7. Click the **Edit** button next to the **Connection string** field. The **Data Link Properties** dialog box appears.

The **Data Link Properties** dialog box is a standard Windows dialog box designed to set up a connection string. If you have specific questions about items in this dialog box, click the **Help** button located in the lower-right corner of the dialog box.

- 8. Select the OLE DB provider you need from the list of providers. For example:
 - If you are using a Microsoft Access database, select the Microsoft Jet 4.0 provider.
 - If you are using a SQL Server, select the Microsoft Provider for SQL Server.
 - If you are using an Oracle database, select the Oracle Provider for OLE DB.
 - For all other integrations, refer to the documentation for your specific database for help in identifying the correct provider.
- 9. Click Next. HotDocs displays the Connection tab of the Data Link Properties dialog box.
- 10. Depending on which OLE DB provider you selected, enter the required information about the data source to which you are linking, including selecting the actual database file.
- 11. When finished entering this information, click **OK**. The **Database Editor** appears again, showing the ADO connection string that will link your template to the database. (This connection string is encrypted when it is saved in the component file, but it will always appear in the **Database Editor** as plain text.)
- 12. Click the **Field Map** tab. The window changes to show how fields in the table will be linked to variables in the template.
- 13. At the **Table name** drop-down list, select the table to which you want to connect. The information from that table appears in the **Field Name** column.

Once you have created the database component and associated a table with it, you must link variables in the template to fields in the database table. (See Link Variables to Database Fields). You also must designate a unique field in the table as the key field, which will allow HotDocs to remember which record the user selects during assembly. (See Remember Selected Records.)

Database connections in HotDocs 5 were maintained using a data presentation layer known as ODBC, or Open Database Connectivity. The ADO-based connection in HotDocs 11 allows you to continue using ODBC by using the OLE DB provider for ODBC, rather than a native OLE DB

provider for the specific database. Maintaining this type of connection may be useful when a native OLE DB provider may either be incompatible or unavailable for you to use. (See **Create a Database Component Using ODBC**.) However, it is recommended that, where possible, you update all existing ODBC-based connections to use a native OLE DB provider for the database program you are using. (See Convert ODBC-based Connections to ADO.)

You can save ADO connection information in a Microsoft Data Link (.UDL) file. This may be useful if you plan to distribute your templates and databases to users whose systems may not be configured the same as yours. Because the connection information is saved in a file separate from the component file, it will make it easier for users without component-editing capabilities to update their connections. See Save ADO Connection Information in a Separate File for details.

If your connection to the database requires you to enter a user name, password, or other options, you can enter them at the **Options** tab of the **Database Editor** instead of including them in the connection string. See Specify a User Name and Password for a Database.

Create a Database Component Using ODBC

In order for programs such as HotDocs to communicate with certain database programs, the database must understand the command HotDocs issues to retrieve data from it. Likewise, HotDocs must understand how to work with the data that gets retrieved from the database. This is accomplished by using a data presentation layer known as ADO, or ActiveX Data Objects. The functionality behind ADO allows HotDocs to communicate with a wide range of database programs using OLE DB providers for each specific database program. In essence, an OLE DB provider acts as a translator between the specific database and HotDocs—it reads information from the database and tells HotDocs how to interpret the data so HotDocs can use it.

Many database programs and/or clients include their own native OLE DB provider (see Create a Database Component Using ADO), but in some situations, an OLE DB provider may either be incompatible or unavailable for you to use. (Click here for a list of supported databases.) In such situations, you can use the OLE DB provider for ODBC (Open Database Connectivity). This type of provider allows you to link your template to the database using an ODBC data source.

To connect to a database using the OLE DB provider for ODBC, you must first make sure you have a database driver installed that is compliant with ODBC Level One or higher standards. Drivers are often installed when you install the database program. If the driver you need is not installed, you should contact the database program provider.

Once you know the correct driver is installed, you must create a data source, which sets up a connection between the database driver and the database so that you can retrieve information from it. You can create one data source to associate a single driver with several database tables, as long as the tables are associated with one database.

Then, once you set up the data source, you must create a database component that uses your data source. HotDocs will use the data source to link to the database table that contains your information. You can use Component Manager to create a database component.

To create an ODBC data source and link it to a database component

- 1. Click **Start > Settings > Control Panel** and navigate to the **Data Source Administrator** dialog box. How you do this depends on which operating system you are using:
 - Windows 2000: Click Settings > Control Panel. The Control Panel dialog box appears. Then double-click the Administrative Tools icon. The Administrative Tools dialog box appears. Finally, double-click the Data Sources (ODBC) icon. The Data Source Administrator dialog box appears.
 - Windows XP: Click Settings > Control Panel > Administrative Tools > Data Sources (ODBC). The ODBC Data Source Administrator dialog box appears.
 - Windows Vista and Windows 7: Click Start > Control Panel > Administrative Tools > Data Sources (ODBC). The ODBC Data Source Administrator dialog box appears.
- 2. With the **Data Source Administrator** dialog box displayed (and the **User DSN** or **System DSN** tab selected), click **Add**. The **Create New Data Source** dialog box appears.
- 3. Select the driver you need and click **Finish**. The **ODBC Setup** dialog box appears.
- 4. Type a unique name for the data source in the **Data Source Name** field.
- 5. If the **ODBC Setup** dialog box includes database version options, select the version you are using.
- 6. Specify the database (or the folder containing the database tables) you want to use. The procedure varies depending on the driver you selected.
- 7. At the **ODBC Setup** dialog box, click **OK** to return to the **Data Source Administrator** dialog box. The new data source is added to the list of data sources.
- 8. Click **OK** at the **Data Source Administrator** dialog box and close the **Control Panel** window (if it's open).
- 9. Open the template you want to connect to the database. (See Edit a Template.)
- 10. Click the **Component Manager** button in the HotDocs toolbar. The **Component Manager** window appears.
- 11. Click the **Components** drop-down button and select **Databases** from the list. The list changes to show only database components.
- 12. Click the **New Component** button. HotDocs displays the **Database Editor**.
- 13. Type a name for the component in the **Component name** field.
- 14. Optionally, perform the following tasks:
 - In the **Title** field, enter a title for the database component. The title will be used in the interview outline and dialog title bar.
 - In the **Prompt** field, type the information about the database table you want the user to see. This information will appear above the database table in the assembly window. (See Create a Prompt for a Variable.)

- Click the **Resource** tab and provide a resource option. A resource helps the user know what type answer to select. (See Add Resource Information to a Variable or Dialog.)
- 15. From the **Connection properties are defined by** drop-down list, select **an ODBC data source**.
- 16. From the **ODBC Data Source Name (DSN)** drop-down list, select the name of the data source you created earlier.
- 17. Click the **Field Map** tab. The window changes to show how fields in the table will be mapped to variables in the template.
- 18. At the **Table name** drop-down list, select the name of the table you want to use. The information from that table appears in the **Field Name** column.

Once you have created the database component and associated a table with it, you must link variables in the template to fields in the database table. (See Link Variables to Database Fields). You also must designate a unique field in the table as the key field, which will allow HotDocs to remember which record the user selects during assembly. (See Remember Selected Records.)

If your connection to the database requires you to enter a user name, password, or other options you do not wish to include in the ODBC data source, you can enter them at the **Options** tab of the **Database Editor**. See **Specify a User Name and Password for a Database**.

In order for HotDocs to remember which records your users have selected during assembly, you must assign a key field. For information on doing this, see **Remember Selected Records**.

Save ADO Connection Information in a Separate File

You can save ADO connection information in a Microsoft Data Link (.UDL) file. When you do this, HotDocs will look in this file (rather than the component file) for the connection information when it tries to access the database. This is useful if you plan to redistribute the template set to users who may need to edit the connection information, for example, because their database is located in a different folder or on a different server than the one you used to develop the template. Otherwise, storing this information in the component file would make it impossible for HotDocs Player users (who have no component-editing capabilities) to make the required changes.

To save an ADO connection string

- 1. Using the **Database Editor**, define the ADO-based connection string. (See Create a Database Component Using ADO.)
- 2. When finished, click the Save button next to the Connection string field. The Save Microsoft Data Link File dialog box appears.
- 3. Specify the location where you want to save the file, type a name for it, and click **Save**. The file is created.

- 4. At the **Database Editor**, click the **Connection properties are defined by** drop-down button and select **a Microsoft Data Link (.udl) file**.
- 5. Click the **Open** button next to the **Microsoft Data Link file** field and locate the data link file. The component now looks in this separate file for connection information.

To edit a data link file, you or your user can double-click the file in Windows Explorer and make the changes in the **Data Link Properties** dialog box.

Microsoft Data Link (.UDL) files are plain text files. If you use a data link file to store your connection information, you should be sure sensitive information, such as database user names and passwords, are not specified in the UDL file. Specify this information in the database component instead, where it will be safely encrypted. See Specify a User Name and Password for a Database for details.

Connect to Microsoft Excel using the Database Component

You can use Microsoft Excel as a readable data source in HotDocs Developer.

HotDocs does not support writebacks to Excel.

To connect to Microsoft Excel using the Database Component

- 1. Open an Excel document to use as a data source.
- 2. In the Excel document, define Names and Ranges for the data you wish to appear in the HotDocs interview. For example:

EmployeeDetails 🔻 💿 🏂 EmployeeID				
	А	В	С	D
1	EmployeeID	FirstName	Surname	
2	12	Eoin	McGuigan	
3	234	Bruce	Mathers	
4	42	Erika	Petrov	
5	345	Gillian	Smith	
6	45	Julianne	Fitzpatrick	
7	535	Simon	Lee	
8	64	Catherine	Larsen	
9				

In this example, the Name of the entire range of cells is *EmployeeDetails*. This Name appears as the name of the database table in HotDocs. The column headings *EmployeeID*, *FirstName*, *Surname* appear as the column headings in the database table. The data underneath the column headings appears as data in the table.

For more details on defining Names and Ranges in Excel, see the Excel help file.

 Create a new ODBC Data Source for Excel. Navigate to Administrative Tools > Data Sources (ODBC) > User DSN. Click Add. The Create New Data Source wizard appears.

If you are using a 64-bit operating system, you must create a 32-bit DSN. To do this, run the 32-bit version of the ODBC Data Source Administrator, which you can find in *C:\Windows\SysWoW64\Odbcad32.exe*. For more information, see http://msdn.microsoft.com/en-us/library/ms712362.aspx

- 4. Select the required Microsoft Excel driver from the driver list. Click **Finish**. The **ODBC Microsoft Excel Setup** dialog appears.
- 5. In the dialog, enter a new Data Source Name and Description. Click the **Select Workbook** button and navigate to the Excel document from Step 1. Click **OK**. Click **OK** to close the **Excel Setup** dialog. Close the **ODBC Data Source Administrator** dialog.
- 6. Open HotDocs. Right-click on the template in which you wish the Excel data to appear and select **Component Manager**.
- 7. In the Component Manager, create a new **Database** component. In the Database Component Editor:
 - Enter a Component name
 - In the **Connection Properties are defined by** drop-down list, select **an ODBC data source**.
 - In the **Command type** drop-down list, select **Simple**.
 - In the **ODBC Data Source Name (DSN)** drop-down list, select the Data Source created in Step 4.
 - Navigate to the **Field Map** tab. In the **Table name** drop-down list, select the required table. Link HotDocs variables to the required database columns and make any other changes to the database component.
 - Click **Test.** An interview window appears displaying the new database component. Check the correct data appears in the component. Once you are finished, close the interview window.
 - Click **OK** to save and close the database component. Close the component manager.

An **ADO data source** can also be used to connect to Excel. See **Create a Database Component Using ADO** for more details on using ADO data sources.

Edit a Database Component

You can edit database components using Component Manager.

To edit a database component

- 1. At the template, click the **Component Manager** button. The **Component Manager** window appears. (See Open and Close Component Manager.)
- 2. Select the database component from the list of components and click the **dit Component** button. The **Database Editor** appears. (To show only database components in the variable list, click the **Components** drop-down button and select **Databases**.)
- 3. Make changes to the component as necessary. For example, you can:
 - Link variables in your template to fields in the database table. (See Link Variables to Database Fields.)
 - Create a filter to control the number of records that will be retrieved from the database table. (See Limit the Number of Database Records from Which a User Chooses.)
 - Sort the data into logical lists for the end user. (See Sort Records in a Database Table.)
 - Cause HotDocs to remember which records you have selected for subsequent assemblies. (See Remember Selected Records.)
 - Specify other database component properties. (See Change Database Component Properties.)

In order for HotDocs to remember which records your users have selected during assembly, you must assign a key field. For information on doing this, see **Remember Selected Records**.

Link Variables to Database Fields

Once you connect your template to a database, you can select a table in the database and link its fields to variables in your template. These links allow HotDocs to retrieve information from the database table each time the template is used to assemble a document.

When linking variables to fields, data types must match. For example, if the field type is text, you must link to either a Text variable or a Multiple Choice variable (since Multiple Choice variables are text values).

During assembly, HotDocs processes a template from top to bottom. When HotDocs encounters a variable that you have linked to a field in a database table, HotDocs first checks to see whether the list of records for that database table has been displayed yet. If it has not, HotDocs displays it so the user can select one or more records.

Database components are designed to link to only one database table. That means that even though you may see several tables listed in the **Table name** drop-down list, you can choose only one table for the component.

To create links between variables and database fields

- 1. Open the database component you want to link to the database table. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Field Map** tab. The window changes to show the fields in the table as well as the variables in the template.
- 3. In the **Linked Variable** column, click the drop-down button that corresponds to the database field to which you want to link.
- 4. Select the variable.
- 5. Optionally, to show both linked and unlinked variables in the **Linked Variable** drop-down lists, clear **Show unlinked variables only**. Be aware, however, that if you select a variable that is already linked, HotDocs will unlink it in order to link it to the new field. Each variable in the template can be linked to only one field at a time. (In these lists, HotDocs uses brackets to show the variable is linked, for example, *[Employee Name]*.)
- 6. Repeat this process for every variable that can be answered using data from the database table.

In order for HotDocs to remember which records your users have selected during assembly, you must assign a key field. For information on doing this, see **Remember Selected Records**.

In HotDocs, NULL values retrieved from a database are considered unanswered while empty string values ("") are considered answered, but empty.

When HotDocs queries the database, it retrieves every record from the table, displays them for the user, and then disconnects from the database. Because of this, you should use a filter on your database component to minimize the number of records that are initially retrieved. (If filtering is not an option but you still want to avoid retrieving all records at once, you can click the **Options** tab of the **Database Editor** and clear **Use disconnected (client) record set**. Be aware, however, that clearing this option can impact the speed with which the user can scroll through and filter large data sets. It also prohibits the user from sorting the data.) (See **Choose a Database Cursor** for more details.)

Remember Selected Records

During assembly, when HotDocs connects to a database, it displays a table of records. Once a user selects a record and moves to another dialog, HotDocs can remember from where in the database table the record was selected so that if the user revisits the table during the interview, the user can see his or her selection.

For HotDocs to operate this way, you must make one of the fields in the database component the key field. A key field contains information that is unique to a record—for example, an invoice number or a

customer identification number. Without a key field, HotDocs has no way to remember from where in the table the record was retrieved and therefore, cannot retain this information once the user moves to a new dialog. This may be confusing to a user who reviews a database record selection—even though answers may appear in the assembled document, the interview shows that no record has been selected.

Selecting a key field also makes it easier to reuse an answer file that contains selected database records, for the reasons explained above.

If you are connecting to a database table using a native OLE DB provider, HotDocs recognizes key fields and automatically assigns them in the database component. (You can always choose a different key field, if necessary.) However, HotDocs does not recognize existing key fields when you connect to the database using the OLE DB provider for ODBC, or when your database component is connected to a database view. You must manually identify the key using the Database Editor.

The field you designate as the key field must always have a non-null value. This applies even in cases where you are defining multiple key fields—if you specify multiple key columns (because no single column uniquely identifies each row), then none of the columns you select should be nullable.

To identify a field as a key field

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Field Map** tab. The window changes to show a list of database fields and their related template variables.
- 3. In the **Key** column, select the field next to the field you want to designate as the key.

If you think you may need an exact replica of a document assembled from a template that includes a database component, be sure to save a copy of the document. If you try to reassemble the document later, even if you use the same answer file, the document may not come out exactly the same because the information stored in the connected database may have changed.

If no single field in the table or view can uniquely identify a selected record, choose multiple keys, making sure the combination of information in those fields will be unique for every record.

Save Changed Answers Back to the Database

When linking variables in your template to fields in a database, you can allow users to edit answers that are retrieved from the database. At times, you may want to save changes the user makes back to the database. To allow this, you can select an option for the database component that allows answers to be saved back to the database record. HotDocs can either always save the answer back to the database, or it

can allow the user to choose whether the answer should be saved back. If changes should never be saved back to the database, you can disallow write-back for a specific variable.

Changed answers are saved back to the database when the user saves the answer file.

To save changes back to a database record

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Properties** tab. The window changes to show the main properties of the database component.
- 3. Select Allow database write-back.
- 4. Click the **Field Map** tab. The mapping spreadsheet appears. By default, all linked variables are set to never allow write-back.
- 5. For each variable's answer you want to save back to the database, click the **Write-back** dropdown button and choose the option you need:
 - **Never** keeps the answer from being written back to the database if it is changed.
 - **Always** always writes the answer back to the database if it is changed. The user will not be notified or prompted of the change.
 - **Ask** displays a dialog box that lets the user decide if the changed answer should be written back to the database.
- 6. Close the database component.
- 7. In the Component Manager add a new dialog component.
- 8. Add the variables, that you applied the write-back settings to in step five, to the new dialog component.
- 9. Name the dialog and close the dialog component editor.
- 10. In your template add an ASK instruction for the new dialog to have it appear in the interview. This will give you the opportunity to edit the answers retrieved for the write-back.
- 11. When finishing an interview, the user will need to save the answer file in order for these answers to be saved back to the database.

To allow users to change answers retrieved from the database, select **Treat linked variables as asked: Never** at the **Options** tab of the **Database Editor**.

The write-back is only available for existing recording in the database and cannot be used to add new records.

Change Database Component Properties

You can change certain properties of a database component. These properties control the way information is retrieved from a database, as well as how the user is able to select which answers will be used in the document.

Database component properties allow you to:

- Let users select multiple records.
- Let users sort and filter records.
- Automatically select records for the user.
- Control which variables are asked in the interview, even if the database answers them.
- Specify user names and passwords for accessing information in the database.

To change database component properties

- 1. Open a database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Options** tab. The window changes to show several component options.
- 3. Make changes as explained in the following table:

То	Do This
Allow the user to choose two or more records during assembly	Select Allow multiple selections. (Make sure the variables in the template are included in a REPEAT Database instruction so that all the answers can be merged correctly.) (See Generate a List of Records in an Assembled Document.)
Force HotDocs to merge a list of filtered answers from a database into the document without allowing the user to make the selection	Select Automatically select all filtered records . (See Merge Filtered Records into the Document Automatically.)
Have HotDocs check the database for new or updated data each time you assemble the document	Select Automatically refresh linked answers . (See Cause HotDocs to Update Selected Records Between Assemblies.)
Have HotDocs retrieve data from a database and then disconnect from it	Select Use disconnected (client) record set . (See Choose a Database Cursor.)
Give users the ability to sort columns in the database table in ascending or descending order	Select Allow sorting by end user . Users can then, during assembly, click column headings in the table and have the records sort in alphanumeric order. They can also control the order records will be merged into the assembled document.

Give users the ability to limit the number of records that appear in the table	Select Allow filtering by end user . This level of filtering does not affect any filters you have used to retrieve the data from the database—it only affects the records the user is viewing at the assembly window.
Change the way HotDocs treats variables that may potentially be answered by the database	Click the Treat linked variables as asked drop-down button and select one of the following options: Always (HotDocs considers linked variables as always asked—it will not ask them, even if the database does not contain an answer), Never (HotDocs considers linked variables as never asked—it will ask them, even if the database contains an answer), or Only if Answered from Database (HotDocs asks a linked variable if no answer for the variable exists in the database). (See Ask Variables Already Answered by the Database.)
Specify a user name and password so that information can be retrieved from the database	Type the user name and password in the User name and Password fields, respectively. This information is encrypted and saved in the component file. (See Specify a User Name and Password for a Database.)

In order for HotDocs to remember which records your users have selected during assembly, you must assign a key field. For information on doing this, see **Remember Selected Records**.

Use a Custom Command to Retrieve Data from a Database

Instead of linking variables to fields using HotDocs field mapping, advanced template developers or database administrators can link variables to a database using a custom command. A custom command lets you extract exactly the data you need from the database. It also may allow you to connect HotDocs to an OLE DB provider representing a non-traditional or non-relational data source.

In order to write a custom command, you must be familiar with the text command syntax of the particular OLE DB provider you are using for the database component. For most relational databases, this is SQL, or at least some variation of SQL. Some OLE DB providers also accept commands in a language or syntax other than SQL. In all cases, HotDocs should work with any OLE DB provider that accepts text commands which return recordsets. HotDocs, however, does not work with OLE DB providers that return records or streams. (Click here for a list of supported databases.)

You write a custom command at the **Command** tab of the **Database Editor**. (Instructions for getting this tab are included below.) This tab lists all the variables in the template to which you can refer in your command. The only modification HotDocs makes to the command string as it passes it through to the database is the substitution of answers for any HotDocs fields it finds in the command.

As you add HotDocs components to a custom command string, keep the following in mind:

- Custom commands should be written so that the returned recordset always has the same number of columns with the same column names.
- Only HotDocs fields that merge variable answers are permitted. Other fields (such as HotDocs scripting or instructions) will not work in database commands.
- When true/false values are merged into a command, an example format is required. When date values are merged into a command, an example format is recommended.
- For HotDocs variables to be mapped to database fields at design-time, HotDocs must execute the custom command to determine what the column names and types are that it returns. Because no answers are available to merge at design-time, HotDocs uses the following guidelines to prepare the command for execution at design-time:
 - If the merge field is a Text variable, Multiple Choice variable or text computation, the merge field is simply replaced by an empty text string.
 - If the merge field is a Number variable or number computation, it is replaced by a zero (0) when the command is issued. If the merge field is a True/False variable or a true/false computation, it is replaced by the false portion of its example format.
 - If the merge field is a Date variable or date computation, it is replaced by its example format if one is specified; if no format is specified, it is replaced by a generic date in the format expected by the OLE DB provider, as specified in the file *Providers.ini*. (Click here for more information.)

While it might be possible to merge a text computation that would vary the number of columns, or the column names or types, that were returned by a command, doing so will prevent HotDocs from properly linking those fields to HotDocs variables. Therefore, HotDocs answers merged into custom commands should only be used to limit or order the results, not to change the scope of the command.

To create a database component using a custom command

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. From the **Command type** drop-down list (found on the **Properties** tab), select **User-defined** (SQL or custom).
- 3. Click the **Command** tab and type the custom command, following the guidelines outlined earlier. Use the command to retrieve the fields from the table. (Click here for an example of a custom command.)

Once the command script is correct and HotDocs has executed the command, the contents of the **Field Map** tab are updated to show both the fields in the database and the variables in the template. You can then link your variables to fields. See Link Variables to Database Fields.

Choose Which Database Fields Appear During Assembly

By default, when you connect your database component to a database table, HotDocs automatically selects every displayable field in the table for display during assembly, even if some of the fields aren't directly linked to variables in the template. (HotDocs automatically omits fields which cannot easily be displayed, such as binary fields.) However, you can choose to display only certain fields.

Also, you can change the order of the database columns as well as the widths of the columns.

To display only selected fields

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Field Map** tab. The window changes to show the options for linking template variables to database fields.
- 3. In the **Display** column, clear the check boxes next to the fields you don't want to appear during assembly. Notice the fields are removed from the list of fields in the **Field display order** list.
- 4. Complete any of the following tasks:

То	Do This
Change the order in which the columns are presented	In the Field display order field, select a field name and drag it to a different location.
Make the column widths larger than the space taken by the field name	Enter the number of units in the Width column. This number must be larger than the number of characters in the column name.
Make the column widths larger than the space taken by the field name	Click Test . HotDocs displays the database table in the test assembly window.

If you are using a custom command, you control the order fields appear in the table using the command script—not by rearranging fields in the **Field display order** field. See Use a Custom Command to Retrieve Data from a Database for details.

In order for HotDocs to remember which records your users have selected during assembly, you must assign a key field. For information on doing this, see **Remember Selected Records**.

When specifying the column width, please note that one unit is about equal to the width of the character **5**.

Cause HotDocs to Update Selected Records Between Assemblies

You can select an option that allows answers retrieved from a database to be current when it is merged into an assembled document. This allows data in a table to be updated without the user having to manually re-select the record or do something else that causes HotDocs to update the connection.

This update happens whenever a user starts a new assembly and then views the database selection dialog. However, if answers in the underlying database change during the interview, those changes will not be reflected in the interview or the document. When this option is cleared (which it is, by default), linked answers will only be updated when the user makes some change to the database table—for example adding, removing, or moving a selected row. Additionally, the data is updated when the database returns nothing or only one record, or when the database component has the **Automatically select all filtered records** option selected.

To update selected records between assemblies

- 1. Open the existing database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Options** tab. The window changes to show several options for the component.
- 3. Select Automatically refresh linked answers.

Limit the Number of Database Records from Which a User Chooses

By default, when HotDocs presents a table of records for the user, every record in the table is displayed. Depending on the size of the table, this may mean the table with which the user is working may be very large. You can limit the number of records that appear, however, by filtering the list of records to include only those relevant to the user.

For example, if you are retrieving customer information from a database, you can create a filter that displays only those customers located in a particular state.

A filter is created by setting up one or more comparisons between the value in a database field and a variable or other value. Only the records that meet the comparison(s) will be displayed.

To filter certain records from the database table

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the Sort & Filter tab. The window changes to show sorting and filtering options.
- 3. Click the **Filter by Field** drop-down button and select the database field you want to use as a filter.

- 4. Click the **Comparison** drop-down button and select the type of comparison operator you want to use.
- 5. Click the **Compared to** drop-down button and select a value to which you want to compare the field. (This list of values matches the data type of the field you are using as a filter. If you selected **IS EMPTY** or **IS NOT EMPTY** as a comparison operator, you do not need to select a value here.)
- 6. Optionally, include another condition for the filter by selecting **AND** or **OR** from the first dropdown list of the second row and completing the condition, just as you did earlier.

Each filter can have up to five comparison conditions, with each condition connected to the preceding condition by the keyword **AND** or **OR**. When two conditions are connected by **AND**, they both must return true in order for a record to be selected. If two conditions are connected by **OR**, only one of the conditions must be true in order for a record to be selected. If the filter has multiple conditions, some connected by **AND** and others by **OR**, the conditions connected by **AND** will be required first, followed by the conditions connected by **OR**.

7. Optionally, click **Test** to test the filter. You will see a list of records as it will be presented to the user during assembly. (See Test Individual Variables.)

In addition to creating a filter yourself, you can allow the user to create his or her own filter during assembly. Filtering options will appear in the interview for the user to select. To do this, select **Allow filtering by end user** at the **Options** tab.

You can use a variable in a filter when you want to make the filter dependent on information the user enters during assembly. For example, you could create a filter to display only those customers located in a particular state, and users could specify different states each time they assemble a document.

If you have used multiple filters, you can remove all of them by clicking **Clear All Filters**. To remove a single filter, click that specific filter's **Field** drop-down button and select the blank line at the top.

If you are filtering a repeated database component, you can also assign filtering instructions at the **REPEAT Field** dialog box. For information about doing this, see **Tips on Filtering and Sorting Repeated Database Components**.

Specify a User Name and Password for a Database

Some databases require users to provide a user name and password before connecting to the database. If this is the case for your database, you can provide the information at the **Options** tab of the **Database**

Editor. Connection strings, user names, and passwords are saved and encrypted in the component file (for safety purposes).

To specify a user name and password at the Options tab of the Database Editor

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Options** tab. The window changes to show several database component options.
- 3. In the **Connection Options** group, type the user name in the **User name** field.
- 4. Type the password in the **Password** field.

You can choose to store the username and password in the connection string (by entering them at the **Data Link Properties** dialog box). However, ADO connection strings are always displayed in plain text in the Database Editor, so anyone with component editing capabilities will be able to view the database password in plain text. (Passwords entered at the **Options** tab of the **Database Editor** are masked, though.)

If you must supply a database password for an Access database, you specify it at the **All** tab of the **Data Link Properties** dialog box. (You can access this dialog box by clicking the **Edit** button at the **Properties** tab of the **Database Editor.**)

Sort Records in a Database Table

To make it easier for users to find records in the database table, you can sort the records by any field in the table, either in ascending (*A to Z*, *1 to 9*) or descending (*Z to A*, *9 to 1*) order. You can also sort a list at two levels. For example, you could sort a client list first by state and then within the state by city.

You can also specify an option that allows users to sort the records during the interview. They can sort on any field in the table and have the contents arranged in ascending or descending order (as described above).

To sort the records for the user

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Sort & Filter** tab. The window changes to show sorting and filtering options.
- 3. Click the first **Sort by** drop-down button and select the field on which you want to sort.
- 4. Click the second **Sort by** drop-down button and select **Ascending** or **Descending**.

5. Optionally, to sort on a second level, select a field at the **Then by** drop-down list and then choose **Ascending** or **Descending**.

You can let users sort their own records. To allow this, select **Allow sorting by end user** at the **Options** tab of the **Database Editor**.

Generate a List of Records in an Assembled Document

By default, users can merge answers from one selected database record during assembly. However, there may be times when choosing multiple records would be necessary, for example, to produce a list of answers in a document.

To generate a list of answers that are retrieved from a database, you must repeat the database component that will provide the answers. You must also specify a database component property that allows the user to select multiple records from the table.

To assemble a single document that contains a list of records

- Select the text (including variables, instructions, and so forth) you want repeated using information from different records and click the REPEAT Field button. The REPEAT Field dialog box appears.
- 2. Select **REPEAT Database**.
- 3. Click the **Database** drop-down button and select the database component you want to use.
- 4. Click the **dit Component** button to open the database component for editing. The **Database Editor** appears.
- 5. Click the **Options** tab and select **Allow multiple selections**.
- 6. Click **OK** at the **Database Editor**. The **REPEAT Field** dialog box appears again.
- 7. Optionally, click **Show Advanced** and choose sorting or filtering options. (See Tips on Filtering and Sorting Repeated Database Components for information about doing this.)
- 8. Click **OK**. The REPEAT instruction is inserted into the template.

Tips on Filtering and Sorting Repeated Database Components

When you REPEAT a database component, you can specify filtering and sorting instructions at two places—at the **Database Editor** and at the **REPEAT Field** dialog box. Generally speaking, instructions

specified at the **Database Editor** control how the database table appears during the interview, while instructions specified at the **REPEAT Field** dialog box control how the data is merged into the assembled document. The following discusses in greater detail these relationships.

Filtering Repeated Database Components

When you assign a filter at the **Database Editor** (at the **Sort & Filter** tab), you are limiting the records from which the user may choose during the interview. When you assign a filter at the **REPEAT Field** dialog box, however, you are limiting the answers that are actually merged into the assembled document.

In most situations, assigning a filter at the database component should be adequate. One situation where it might be useful to create an additional filter at the **REPEAT Field** dialog box is if you want the user to select answers from one table but you want the answers merged into different lists in the document.

Sorting Repeated Database Components

Similarly, when you assign sorting instructions at the **Database Editor** (at the **Sort & Filter** tab), HotDocs forces the records in the table to appear in that order during the interview. If you also select **Allow sorting by end user** at the **Options** tab of the **Database Editor**, the table initially is presented to users using your sorting order, but users can then sort the records how they want. If users select multiple records, they can then rearrange the order the answers are merged into the assembled document.

When you assign sorting instructions at the **REPEAT Field** dialog box, HotDocs will merge the answers into the assembled document in the order you specify.

Where possible, you should always use the **Database Editor** to assign sorting instructions for a repeated database. One situation where it might be useful to assign sorting instructions at the **REPEAT Field** dialog box would be if you need to insert a list of answers at multiple places in the document but need the list to be arranged differently in each location. For example, perhaps in one location, you would need the list to be sorted based on *Invoice Number*, but in another place the list needs to be sorted by *Company Name*. You could use the sorting instructions at the **REPEAT Field** dialog box to help you create these different lists. Make sure, however, that you clear **Allow sorting by end user** at the **Options** tab of the **Database Editor**.

Merge Filtered Records into the Document Automatically

You can have HotDocs automatically select all filtered records and set the linked repeated variables accordingly. This can be useful if you've designed a filter to automatically retrieve this information from the database and you don't need the user to make any selections.

To merge all filtered records into the document during assembly

- 1. Repeat the variables you want answered by the database. (See Generate a List of Records in an Assembled Document.)
- 2. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 3. Click the **Options** tab. The window changes to show several database component options.
- 4. Select Allow multiple selections and then select Automatically select all filtered records.

Now, when your user assembles a document, any records that meet the filter condition are automatically selected and used to set linked variables—the table of records will not appear during assembly.

Ask Variables Already Answered by the Database

When variables in a template are answered using data from a database, HotDocs considers the variables to be answered and, by default, won't ask them again. You can, however, have HotDocs ask the variables, even though answers have already been provided. This may be useful if you want the user to review or change the answers supplied by the database. Then, if you choose to allow it, users can save any changes they've made to the data back to the database. (See Save Changed Answers Back to the Database for details.)

Similarly, if a variable requires an answer, but no answer for the variable exists in the database, HotDocs can ask the variable and allow the user to provide the answer.

To control which linked variables are asked during the interview

- 1. Open the database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Options** tab. The window changes to show several database component options.
- 3. Click the **Treat linked variables as asked** drop-down button and select one of the following options:
 - Choose **Never** to have HotDocs never consider linked variables as asked—it will ask them elsewhere in the interview, even if the database contains an answer.
 - Choose **Always** to have HotDocs always consider linked variables as asked—it will not ask them elsewhere in the interview, even if the database does not contain an answer.
 - Choose **Only if Answered from Database** to have HotDocs ask a linked variable if no answer for the variable exists in the database.

Select Related Database Records

Sometimes your template may link to more than one database table. If both tables have at least one field in common (for example, both tables have a field named *CLIENT_ID*), you can create a filter based on this association that will allow users, during assembly, to select a record from one table and have the corresponding records from the other table automatically selected. (This common field is sometimes referred to as a foreign key.)

For example, perhaps you want to assemble an invoice letter for a specific client whose payment is past due. Specifically, you want to select an unpaid invoice record from the INVOICE table and automatically have the associated information about the client selected from the CLIENT table. This data should then be merged into the assembled document. To do this, you must create both database components, link the fields in each table to variables in the template, and assign a filter to the CLIENT table. The filter uses the value of CUSTOMER_ID (which has already been answered by the INVOICE table) and returns the record from the CLIENT table that matches that value.

To select related records in another table

- 1. Create database components for two tables in your database. (See Create a Database Component Using ADO and Link Variables to Database Fields.)
- 2. Open the database component for the database table from which you want the user to choose a record. (See Edit a Database Component.)
- 3. Click the Field Map tab and link the related (or common) field to a variable in the template.

If no variable exists, create it by clicking the **Linked Variable** cell and then clicking the ***New Component** button. HotDocs automatically creates a variable that matches—as closely as possible—the properties of the field. The variable is created, but not inserted into the template, and the field is linked to it.

- 4. Open the other database component for editing. (This is the database from which you want to select related records.) The **Database Editor** appears.
- 5. Click the **Sort & Filter** tab. The window changes to show sorting and filtering options.
- 6. Click the **Filter by Field** drop-down button and select the related field.
- 7. Click the **Comparison** drop-down button and select **EQUAL** from the list of comparison operators.
- 8. Click the **Compared to** drop-down button and select the variable that corresponds to the related fields.
- 9. At each **Database Editor**, click **OK** to close both windows

If you are retrieving answers from multiple records, you can have HotDocs automatically select all the records that meet the filtered condition and merge them into the document. To do this, open the database component that contains the filter (this is the database component you don't want displayed during assembly) and select **Automatically select all filtered records**. (See Merge Filtered Records into the Document Automatically for more details on using this option.)

See Custom Command Sample for an example of a custom command that actually joins two database tables.

Control When Your Record Selection Screen Appears

When HotDocs creates an interview for the user, it reads through the template and displays dialogs based on the fields it finds in the template text. If a field contains a variable that is linked to a database component, HotDocs displays the record selection screen of that database component instead.

If you want the record selection screen to appear at a certain place in the template, you can insert an ASK Database instruction into the text of the template where you want the database table to be displayed.

To insert an ASK Database instruction

- 1. Position the cursor at the point in the template where you want the table to be asked (for example, at the top of the template).
- 2. Click the **ASK Field** button. The **ASK Field** dialog box appears.
- 3. Select **ASK Database**.
- 4. Select a database table from the **Database** drop-down list, or click the **Edit Component** button to create a new database component. (See Create a Database Component Using ADO and Link Variables to Database Fields.)
- 5. Click **OK**. The **ASK Database** instruction is inserted in the template.

You can group several ASK instructions in one computation script and use that computation to control the order questions are asked in the interview. (See Put ASK Instructions in a Computation Variable.)

Choose a Database Cursor

When HotDocs connects to the database and identifies the information it needs to retrieve, it uses what is called a cursor. Just as a cursor in a word processor document allows you to move around and select text in a document, a database cursor allows HotDocs to scroll through and select records from a database table.

A database component can use one of two types of cursors when retrieving data from a database table: client cursors and server cursors. By default, HotDocs uses a client cursor; however, you can choose to have it use a server cursor. The following is a discussion about the advantages of each.

Client Cursors

A client cursor retrieves all the data in the result set (possibly across the network) to the user's machine, and manipulates the data there. The advantages of this are:

- End users will have fast scrolling and full ability to change sort order.
- The data is only pulled across the network one time.
- Server resources (connections) are freed as soon as the data is retrieved. No server connection is maintained while the user is interacting with the database component, selecting records. This has minimal impact on server resources.

The primary disadvantage of client cursors is that for very large result sets, the initial loading of the data across a network consumes bandwidth and can be time consuming. That is why, in HotDocs, it is recommended that you always use filters to minimize the size of the result sets from which users will choose. (See Limit the Number of Database Records from Which a User Chooses.)

Server Cursors

Unlike a client cursor, a server cursor opens the connection to the database and leaves it open until the user leaves the database selection dialog. This may be desirable in a few circumstances, such as when a large result set is required and the initial performance of a client cursor is prohibitive. However, the disadvantages of a server cursor are:

- User scrolling can be slower, as only the displayed data is retrieved from the database at one time.
- A connection to the database server must be maintained the whole time the user is selecting records from the database. Also, network traffic may be impacted if the user scrolls through the data repetitively.
- The user cannot change the sort order of the displayed results.
- If the user applies filters to very large result sets, performance and network bandwidth usage can be negatively impacted.

If you're accessing a database that is located on your local machine, there may not be a huge performance difference between client and server cursors. The biggest difference in this case is that server cursors will not allow you to change the sort order of the data you are viewing during the interview.

Because of these limitations, it is recommended you use client cursors—unless you have a specific reason to choose otherwise.

Once you decide which type of cursor best fits your needs, you can specify this option at the **Options** tab of the **Database Editor**.
To specify which type of cursor you will use

- 1. Open the existing database component for editing. (See Edit a Database Component.) The **Database Editor** appears.
- 2. Click the **Options** tab. The window changes to show several options for the component.
- 3. Choose one of the following options:
 - Select Use disconnected (client) record set to use a client cursor.
 - Clear Use disconnected (client) record set to use a server cursor.

Using Dot Codes to Format Text and Insert Special Characters

Introduction: Dot Codes

Working with Plain Text

When working with plain text, either in variable prompts, dialog element text, plain-text resources, or in answers generated by computation scripts, you frequently want to assign font properties to the text, such as bolding, italicizing, and underlining. Additionally, you may want to punctuate and capitalize characters in plain text as well as insert special characters, such as hyphens and other punctuation marks.

To format text, insert dot code fields in your prompts, text, or scripts. During assembly, these dot codes will be processed and will apply whatever formatting you specify to the text.

Unfortunately, dot codes that format answer text in a form document cannot be processed because the text is in plain-text format and cannot allow formatting, such as bold and italics. To explain, think of when you've worked on a plain-text document in Notepad. You can insert characters like line breaks, section symbols, and punctuation in a Notepad document, but you cannot make character-level formatting changes. HotDocs uses this same plain text in form document answers. Dot codes that insert characters (such as line breaks, section symbols, and so forth), however, are processed without any problems in forms.

Punctuating Non-Repeated Lists

Dot codes can also be used to merge punctuation for non-repeated lists in the template. Using a series of dot codes, you can define the punctuation mark that should be used to separate items in the list as well as define the conjunction (*for, and, nor, but, or, yet, so*) used for the list. This is useful when you have separate answers that are being merged conditionally in list format.

Full List of Dot Codes

The following tables list all of the dot codes that are available in HotDocs:

Character Format:

Name	Dot Code	What it Does
Bold	«.b»/«.be»	Applies bold formatting to the enclosed word or group of words
Hidden	«.h»/«.he»	Applies the hidden text property to a word or group of words

		The dot code «.h»/«.he» is not allowed in prompts, dialog titles, or resources.
Italic	«.i»/«.ie»	Applies italic formatting to a word or group of words
Stike-through	«.x»/«.xe»	Crosses out a word or group of words
Underline	«.u»/«.ue»	Applies underline formatting to a word or group of words
Size	«.z»/«.ze»	Increases or decreases the font size for a word or group of words by a specified percentage.
		The dot code «.z +Size» is a percent increase in the current size. The dot code, «.z -Size» is a percent decrease in the current size. (When you select this option, HotDocs displays a dialog box asking you to specify these properties.)
All Caps	«.a»/«.ae»	Capitalizes every letter in the selected text
Capitalize Letter	«.C»	Capitalizes the word immediately following the code
Leading Caps	«.l»/«.le»	Capitalizes the first letter of each word in the enclosed text
Small Caps	«.s»/«.se»	Capitalizes each lowercase letter in the enclosed text and displays them at a smaller font size.
		For example, "The Quick Brown Fox" would be come "THE QUICK BROWN Fox".
		The dot code «.s»/«.se» is only fully supported in JavaScript interviews. In Silverlight interview all text is capitalized and displayed at the smaller font size, including existing capital letters. In desktop interviews this dot code has no effect.

Character Insertion:

Name	Dot Code	What it Does
An	«.an»	Inserts <i>a</i> or <i>an</i> , depending on whether the word that follows begins with a consonant or a vowel
Left Apostrophe	«.la»	Inserts a left apostrophe
Right Apostrophe	«.ra»	Inserts a right apostrophe
Left Quote	«.lq»	Inserts a left quotation mark
Right Quote	«.rq»	Inserts a right quotation mark

Paragraph Symbol	«.ps»	Inserts a paragraph character
Section Symbol	«.SS»	Inserts a section symbol character
Tab Character	«.tc»	Inserts a tab character
Line Break	«.lb»	Inserts a line break
Paragraph Mark	«.pm»	Inserts a paragraph mark
Page Break	«.pb»	Inserts a page break
Non-breaking Space	«.ns»	Inserts a non-breaking space character
Non-breaking Hyphen	«.nh»	Inserts a non-breaking hyphen character
Optional Hyphen	«.oh»	Inserts an optional hyphen character

Sentence Punctuation:

Name	Dot Code	What it Does
Period	«.»	Inserts a period if no other punctuation precedes it
Colon	«:»	Inserts a colon if no other punctuation precedes it
Exclamation Mark	«!»	Inserts an exclamation point if no other punctuation precedes it
Question Mark	«?»	Inserts a question mark if no other punctuation precedes it
Comma	«,»	Inserts a comma if no other punctuation precedes it
Semicolon	«;»	Inserts a semicolon if no other punctuation precedes it

List Punctuation:

Name	Dot Code	What it Does
Punctuation Setup	«.p "style"»«.pe»	Identifies the beginning of a punctuated list and assigns the punctuation format to it. The closing dot code identifies where the punctuated list ends.
Punctuation Item	«.p»	Used within the Punctuation Setup and identifies where the punctuation character should be inserted

Web Link:

Using Dot Codes to Format Text and Insert Special Characters

Name	Dot Code	What It Does
Web Link Dot Code	«.w "URL"»/«.we»	Inserts a Web link Enter the link text between the two dot codes

Change Font Properties of Text

Dot codes that change the character format (bold, italic, font size, etc.) do not work in prompts for True/False and Multiple Choice variables that are set to appear as buttons or check boxes.

Frequently, you use a computation script to generate an answer that must be merged in the document. Because answers generated by a computation script are in plain text (which can't be formatted at the Script Editor), you can insert formatting dot codes in the plain text. These dot codes will be processed and the formatting will be applied once the answer is merged in the document.

Additionally, you can use dot codes to assign formatting to variable prompts and dialog element text. This allows questions in the interview to be formatted according to your needs. Specifically, using dot codes, you can apply font properties (such as bolding and italicizing) to answers. You can also specify font sizes and capitalization.

When formatting sections of text using dot codes, you must first highlight the text you are formatting before you assign the dot code. This surrounds the text with beginning and ending codes that instruct HotDocs where the formatting starts and stops. Additionally, when you insert multiple dot codes for a single block of text, the codes must be nested, meaning the first opening dot code you assign must be the last closing dot code. For example, *this is an example of «.b»«.i»nesting«.ie»«.be»*.

To apply formatting to plain text

- 1. Either edit a Computation variable (See Customize a Computation Variable), or edit a prompt or dialog element text (see Create a Prompt for a Variable and Add Text to Your Dialogs).
- 2. Highlight the text you want to format and right-click. A shortcut menu appears.
- Choose Character Format > Format (where Format represents one of the dot codes described in the following table). (WordPerfect users: To insert the dot codes, manually type the codes as you see them in the table below.)

Dot Code	What It Does
«.b» (Bold)	Applies bold formatting to a word or group of words.
«.be» (Bold End)	

«.h» (Hidden)	Applies the hidden text property to a word or group of words.
«.he» (Hidden End)	
The hidden property is not supported in WordPerfect templates.	
«.i» (Italics)	Applies italic formatting to a word or group of words.
«.ie» (Italics End)	
«.u» (Underline)	Applies underline formatting to a word or group of
«.ue» (Underline End	words.
«.x» (Strike-through)	Crosses out a word or group of words.
«.xe» (Strike-through End)	
«.z» (Font size)	Increases or decreases the font size for a word or group of words by a specified percentage. The dot code.
«.ze» (Font size End)	+ Size », is a percent increase in the current size. The dot code, «.z -Size » is a percent decrease in the current size.
The font size options are not	(When you select this option, HotDocs displays a dialog
supported in WordPerfect	box asking you to specify these properties.)
templates.	

The following is an example of a computation script that uses dot codes. The account status is bolded:

IF DAYS FROM(Purchase Date, TODAY) > 60
"Your account is «.b»past due«.be»."
ELSE
"Your account is «.b»current«.be». Thank you."

END IF

Likewise, this is an example of a variable's plain-text resource that uses dot codes. Here, the name of a required form is italicized:

To get free service of the court's orders without paying a fee, you must fill out and file the «.i»Request and Order for Free Service of Restraining Order«.ie».

When you type double angle brackets (<< >>), HotDocs converts them to chevrons (« »). You can manually enter the dot code by typing the brackets with the format code between them.

Insert a Web Link

You can use a Web link dot code (**«.w "URL"»/«.we**») to add a Web address to a variable prompt or dialog element text. The user will be able to click this link during the interview and a separate browser window (displaying the specified Web page) will appear.

To include a Web address in a prompt or dialog text element

- 1. Edit a prompt or dialog text element. (see Create a Prompt for a Variable and Add Text to Your Dialogs.)
- 2. Place your cursor where you want the link to appear and right-click. A shortcut menu appears.
- 3. Choose Web Link. The Web Link dialog box appears.
- In the URL field, enter the complete Web address for the page you want the hyperlink to open. (Make sure the URL is entered correctly. Include http:// or https:// or ftp://, depending on what type of URL it is.)
- 5. Enter the text you want the user to see for the link in the **Link text** field.

To include a Web address in the document text

Using a .docx template you can also insert a Web link into the body text of a document and have it work as a hyperlink after assembly.

- 1. Open a .docx template for editing. (see Edit a Template.)
- 2. Select the text you wish to use as the link text for the Web Link and right-click. A shortcut menu appears.
- 3. Select Character Formatting and then Web Link. The Web Link dialog box appears.
- In the URL field, enter the complete Web address for the page you want the hyperlink to open. (Make sure the URL is entered correctly. Include http:// or https:// or ftp://, depending on what type of URL it is.)

Before creating the Web link, you can select existing text in the prompt or dialog element text and that text will be used as the **Link text**.

If you don't know the exact URL for the Web page, click the **Browse Web** button next the **URL** field to search for the Web page.

You can include a link that opens an e-mail message rather than a Web address. To do this, enter **mailto:**, followed by the email address you want included, in the **URL** box. (For example, enter **mailto:publications@hotdocs.com**.)

Add Punctuation and Capitalization to Sentences

Sometimes you need answers in the assembled document to be capitalized and punctuated correctly, but you can't anticipate how users will enter the answer. To ensure text in the document is correct, you can use dot codes to merge the correct capitalization and punctuation.

For example, say the user enters a description but doesn't include the end punctuation in the answer. Using a dot code, you can merge the punctuation in the document if the user doesn't. (If the user does include the punctuation, the dot code won't merge anything.) The following script demonstrates this:

The property, which is located at «Property Address», is described as «Property Description»«.»

To add punctuation and capitalization dot codes

- 1. Either edit a Computation variable (See Customize a Computation Variable), or insert your cursor in the template text where you want to insert the dot code.
- To insert a punctuation mark, right-click to display the shortcut menu and choose Sentence Punctuation > Punctuation Mark (where Punctuation Mark represents one of the dot codes described in the following table. (WordPerfect users: To insert the dot codes, manually type the codes as you see them in the table below.)

Dot Code	What It Does
«.»	Inserts a period if no other punctuation precedes it.
«,»	Inserts a comma if no other punctuation precedes it.
«;»	Inserts a semicolon if no other punctuation precedes it.
« : »	Inserts a colon if no other punctuation precedes it.
«!»	Inserts an exclamation point if no other punctuation precedes it.
«?»	Inserts a question mark if no other punctuation precedes it.

 To change the capitalization of text, right-click to display the shortcut menu and choose Character Format > Capitalization (where Capitalization represents one of the dot codes described in the following table:

Dot Code	What It Does
«.a» (All Caps)	Capitalizes every letter in the selected text.
«.ae» (All Caps End)	
«.c» (Capitalize Letter)	Capitalizes the first letter of the word immediately following the code.
«.l» (Leading Caps)	Capitalizes the first letter of each word in the selected text.
«.le» (Leading Caps End)	
«.s» (Small Caps) «.se» (Small Caps End)	Capitalizes each letter in the selected text and displays the text at a smaller font size.

When you type double angle brackets (<< >>), HotDocs converts them to chevrons (« »). You can manually enter the dot code by typing the brackets with the format code between them.

Insert Characters in Text Strings

When working with plain text in both computation results and in variable prompts, dialog element text, and resources, you frequently need to insert a character not typically supported by the plain-text character set. Using dot codes, you can insert these characters so that when the answer is merged in the document, the dot codes are processed and the sentence is punctuated correctly.

For example, if you are merging a literal text string in a computation script and the text needs smart quotation marks (and not straight quotation marks), you can insert the dot codes that merge the correct marks.

To insert special characters in plain text

- 1. Either edit a Computation variable (See Customize a Computation Variable) or edit a prompt or dialog text element. (see Create a Prompt for a Variable and Add Text to Your Dialogs.)
- 2. Place your cursor where you want the character and right-click. A shortcut menu appears.

3. Choose **Character Insertion > Character** (where **Character** represents one of the dot codes described in the following table). (**WordPerfect users:** To insert the dot codes, manually type the codes as you see them in the table below.)

Dot Code	What It Does
«.an» (An)	Inserts <i>a</i> or <i>an</i> , depending on whether the word that follows begins with a consonant or a vowel. For example:
	«Employee Name», «.an»«Job Title»
	becomes
	Jack Carey, a paralegal
	or
	Bonnie Millet, an attorney
«.la» (Left Apostrophe)	Inserts a left or right apostrophe.
«.ra» (Right Apostrophe)	
«.lq» (Left Quote)	Inserts a left or right quotation mark.
«.rq» (Right Quote)	
«.ps» (Paragraph Symbol ¶)	Inserts a paragraph character.
«.ss» (Section Symbol §)	Inserts a section symbol character.
«.tc» (Tab Character)	Inserts a tab character.
«.lb» (Line Break)	Inserts a line break.
«.pm» (Paragraph Mark)	Inserts a paragraph end.
«.pb» (Page Break)	Inserts a page break.
«.ns» (Non-breaking Space)	Inserts a non-breaking space character.
«.nh» (Non-breaking Hyphen)	Inserts a non-breaking hyphen character.
«.oh» (Optional Hyphen)	Inserts an optional hyphen character.

When you type double angle brackets (<< >>), HotDocs converts them to chevrons (« »). You can manually enter the dot code by typing the brackets with the format code between them.

Punctuate Non-Repeated Lists of Answers

Sometimes you create lists of answers in a template without using a REPEAT instruction. Often, you need a way to punctuate the list, depending on answers that are merged. For example, say you need to list the client name, the client's spouse's name (if the client is married), and the children's names (if there are children). Such a list could look like this in the assembled document:

The client's family consists of Tim Anderson, spouse Lori Anderson, and children Jessica Anderson, Aubrey Anderson, and Lance Anderson.

However, not all clients will be married, nor will all clients have children. Because of this, you need some way to punctuate the sentence, based on which answers are added to the list. You can accomplish this using a series of dot codes that punctuate a sentence (including adding a conjunction between items in the list), depending on where answers are merged in the document.

To punctuate a non-repeated list

- 1. Edit the template. (See Edit a Template.)
- In the template text where you want to merge the list punctuation, right-click and choose List
 Punctuation > Punctuation from the Word shortcut menu (where Punctuation represents the
 mark you want to insert, based on the following table). (WordPerfect users: To insert the dot
 codes, manually type the codes as you see them in the table below.)

Dot Code	What It Does
«.p "style"» (Punctuation Format)	Identifies the beginning of a punctuated list and assigns the punctuation format.
«.p» (Punctuation)	Identifies the spot where punctuation character should be inserted.
«.pe» (Punctuation End)	Identifies the end of the punctuated list.

The following is an example of how punctuation dot codes are used in the text of a template. The first dot code («.p "a, b, and c"») indicates the start of the list (and specifies the punctuation and conjunction that should be used). Each item that is added to the list is marked by a punctuation code («.p») which indicates where the punctuation mark should be merged. The end punctuation mark («.pe») indicates the end of the list:

«.p "a, b, and c"»The client's family consists of «Client Name»«.p»«IF Client is Married»spouse «Spouse Name»«.p»«END IF»«IF Client Has Children»children «REPEAT Children:a, b, and c»«Child Name»«END REPEAT»«.p»«END IF»«.pe».

In this example, there are really two lists—first, the list of family members (client, spouse, and children) and then second, the list of individual children. While dot codes are used to punctuate this first list, the repeat format for the REPEAT instruction punctuates the list of children. However, you can use dot codes

to punctuate the items in a REPEAT instruction, as well. The following script demonstrates how to nest these dot codes:

«.p "a, b, and c"»The client's family consists of «Client Name»«.p»«IF Client is Married»spouse «Spouse Name»«.p»«END IF»«IF Client Has Children»«.p "a, b, and c"»children «REPEAT Children»«Child Name»«.p»«END REPEAT»«.pe»«.p»«END IF»«.pe».

Testing Automation

Testing HotDocs Automation

Introduction: Test HotDocs Templates

Test Variables, Scripts, and Dialogs

As you create variables and computation scripts in your templates, you can test them to make sure they produce the desired result. Likewise, you can test dialogs to see how the information will be presented to users during document assembly.

As you test these different components, HotDocs displays the HotDocs test assembly window. If you find that you must edit the variable, script, or dialog, you can leave the assembly window open, bring the component editor to the front, and make the required changes. Then, to continue testing, you can click the **Update** button to review your changes. (Note, however, that these changes aren't saved to the component file until you click **OK** or **Save** at the component editor.)

Test Assemble Documents

At any point during template development, you can test all or part of the template to make sure it is assembling correctly. Test assembling a document can help you pinpoint trouble spots in the template without requiring you to close the template and assemble it from the library.

When you test assemble a document, HotDocs displays the same HotDocs assembly window users will see. At any time during test assembly, you can return to the template, make the desired changes, and test assemble the document again. The test assembly window will adjust to show your changes.

Use the Test Panel

To help you test, you can use the HotDocs Test Panel. This feature lets you examine variable usage, including whether variables referenced in the interview are actually used in the document (and vice versa). Features of the Test Panel also let you check to make sure you've properly asked variables and set values for variables in the template. Finally, during a test assembly, you can link from the **Document Preview** tab of the test assembly window back to a specific place in the template.

Resolve Problems in a Template or Script

As you test, you may find problems with your automation. For example, HotDocs may display a syntax error or warning you must fix before the template or script can be properly tested. To help you resolve syntax errors, HotDocs displays error messages that contain information about the specific error as well as a command for going to the error in the template.

Additionally, perhaps a test assembly produces a result you don't expect. To help you step through a template or script in order to determine why you're receiving unexpected results, HotDocs provides the HotDocs Debugger.

Test Individual Variables

When you're creating a variable or an IF instruction, you can test it to see if it's working the way you want. Even if the variable is linked to a custom dialog, HotDocs still tests it individually. You can test variables by clicking the **Test** button at the variable editor.

To test a variable or True/False expression

- 1. Edit a variable or IF instruction. (See Edit a Variable or Edit an IF Instruction or Expression.)
- 2. Click **Test** at the variable editor. HotDocs displays a test assembly window.
- 3. Enter an answer for the variable.
- 4. Optionally, if you assigned a resource to the variable, view the **Resource** pane. (The **Resource** pane appears below the dialog pane. If it is not visible, select it at the **View** menu.)
- 5. Click the **Result** tab to see the answer that would be merged in the assembled document.
- 6. Optionally, place your cursor in the answer field and click the **Edit Component** button in the assembly window toolbar. The variable editor comes to the front so you can make changes. Click **Update** to update the window. To permanently save the changes, click **OK** at the variable editor. (See Simultaneously Edit a Template and Test Assemble a Document.)
- 7. Click **Close** (File menu) to close the test assembly window.

You can save your test answers in a test answer file. See Use a Test Answer File.

You can have HotDocs arrange the test assembly window with other open windows. Adjust the height of the test assembly window and then click the **Arrange** button.

Test a Computation Script

Often, as you write calculations or scripts, you want to make sure they produce the desired result. You can test computations and scripts as you write them. When you do this, HotDocs displays a test dialog in an interview where you can enter the information that is needed to produce a result.

To test a script

- 1. At the **Computation Editor**, enter your computation or script.
- 2. When finished, click **Test**. HotDocs displays a test assembly window.
- 3. Enter any required information and click the **Result** tab. HotDocs displays the computed answer.
- 4. **(Optional)** To further edit the variable, leave the test assembly window open and return to the Computation Editor to make the changes. Click **Update** at the **Computation Editor** to update the test assembly window. To permanently save the changes, click **OK** at the **Computation Editor**.
- 5. Click **File**, then **Close** to close the test assembly window.

You can save your test answers in a test answer file.

When you test a computation script, if HotDocs finds any **syntax errors** in the script, it displays a warning message and then places your cursor as close to the error as possible in the **Script** box. You must fix the error before HotDocs allows you to save the script.

You can have HotDocs arrange the test assembly window with other open windows. Adjust the height of the test assembly window and then click the **Arrange** button.

Test a Custom Dialog

You can test a custom dialog to see how the dialog contents will appear during an interview, including any scripting you may have assigned. When you test assemble a dialog, HotDocs displays a test assembly window so you can see the dialog exactly as the user will see it.

To test a custom dialog

- 1. Edit the dialog you want to test. (See Edit a Custom Dialog.)
- 2. At the dialog, click **Test**. HotDocs opens a test assembly window, which shows the different components of the dialog you are editing.
- 3. Make sure variable prompts, dialog elements, and check boxes appear correctly.
- 4. Optionally, select the dialog icon in the interview outline and click the **Edit Component** button in the assembly window toolbar. The Dialog Editor comes to the front so you can make changes. Click **Update** to update the window. To permanently save changes, click **OK** at the **Dialog Editor**. (See Simultaneously Edit a Template and Test Assemble a Document.)
- 5. If you've created a dialog script, type information in the answer fields to make sure the script is updating the dialog correctly.

You can save your test answers in a test answer file. See Use a Test Answer File.

You can have HotDocs arrange the test assembly window with other open windows. Adjust the height of the test assembly window and then click the **Arrange** button.

Test Assemble a Document

When you're automating a template, you can test assemble it at any time to see if the document assembles the way you want. When testing a text document, you can test assemble all or just part of it. When testing a form template, you can test using direct-fill mode as well as using an interview (if one has been specified).

To test assemble a text template

- 1. At the template, click the **Assemble** button. HotDocs opens a test assembly window.
- 2. Answer questions as they are presented.
- 3. Click the **Document Preview** tab to see the assembled document, just as the user will.
- 4. Choose **Close** (**File** menu) to close the test assembly window, or switch windows to return to the template.

To test assemble a form template

- 1. At the form template, click the **Assemble** button. HotDocs opens a test assembly window.
- 2. If you have created a custom interview, you can view the template's variables and dialogs in the left pane (or interview outline) and the corresponding answer fields in the right pane (or dialog pane). Answer questions as they are presented.
- 3. Click the Form Document tab to see the assembled document, just as the user will.
- 4. Optionally, click or tab through the fields in the **Form Document** tab to fill the form directly.
- 5. Choose **Close** (**File** menu) to close the test assembly window, or switch windows to return to the template.

You can save your test answers in a test answer file. See Use a Test Answer File.

As you complete the interview, you can move between dialogs by clicking dialog icons in the interview outline, by clicking the **Next** or **Previous** buttons in the navigation bar, or by pressing **Alt+N** or **Alt+P**.

To test a template for online assembly, press the **Shift** key while clicking the **Test Assemble** button. (See View an Interview in a Web Browser.)

You can have HotDocs arrange the test assembly window with other open windows. Adjust the height of the test assembly window and then click the **Arrange** button.

When you test assemble a document from a template, if HotDocs finds any syntax errors in the template, it displays a warning message, which allows you to go to the place in the template

where the error occurs. You must fix the error before HotDocs will let you continue. (See Understand and Resolve Syntax Errors in a Template or Script.)

In Microsoft Word, you can also test assemble a document either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Test Assemble**, or by right-clicking in the template and choosing **Test Assemble** from the shortcut menu.

If you receive an error message that includes a field number, you can quickly go to that field in the template. To do this, click the **\blacksquare**+**Go To Field** command. See **Go** to a **Specific** Field in the **Template** for details.

Use a Test Answer File

As you test variables, scripts, dialogs, and templates, you can save the answers you use in a special test answer file. When you do this, these answers will be used each time you test, and you will not be prompted to save answers at the end of each test.

To save your answers in a special test answer file

- 1. Test assemble a variable, dialog, or template by entering answers for each question.
- 2. When finished, close the assembly window. HotDocs asks if you want to save your answers.
- 3. Click Save or Save As. The Save Answer File dialog box appears.
- 4. In the File name field, type Test Answer File.
- 5. In the **Title** field, enter a title or accept the suggestion HotDocs makes.
- 6. Click **OK**.

Now, whenever you test assemble in HotDocs, this answer file will be used and you will not be prompted to save your answers. To use a different answer file, click either the **New Answers** button or the **Open Answers** button in the test assembly window toolbar. If you use an existing answer file other than *Test Answer File*, HotDocs will use this answer file for each test assembly, but you will be prompted to save any changes you have made to the file when you close the assembly window.

Simultaneously Edit a Template and Test Assemble a Document

You can test a variable, dialog, or script to make sure the information is being processed correctly. When you don't get the results you expect, you can leave the test assembly window open as you make the necessary changes at the component editor for the component you are testing. You can then click the

Update button and HotDocs refreshes the assembly window to show your changes. Changes aren't saved to the component file until you click **OK** at the component editor.

To update the contents of a test assembly window with changes

- 1. At the template you are editing, test assemble a variable, dialog, computation script, or template.
- 2. Place your cursor in an answer field (or select a dialog in the interview outline) and click the **Edit Component** button. HotDocs displays the component editor for that specific component.
- 3. Make any desired changes to a component or the template text and click **Update**. HotDocs makes the changes in the test assembly window.
- 4. Repeat this process until you see the desired result.
- 5. When you are finished, close the test assembly window and click **OK** at the component editor. The changes are saved to the component file.

You can have HotDocs arrange the test assembly window with other open windows. Adjust the height of the test assembly window and then click the **Arrange** button.

Understand and Resolve Syntax Errors in a Template or Script

When HotDocs encounters a component or instruction in a template or script that it doesn't recognize or understand, it generates and displays a syntax error message. This message lists the component or components HotDocs was processing when it encountered the error and allows you to go directly to that point in the template or script so you can fix the problem. (See Understand the HotDocs Scripting Language.)

Errors may be caused by:

- **Unrecognized variable names:** Any references to variables or other components must have a corresponding component in the component file. For example, you cannot refer to the Text variable *Employee Name* in the template unless there is Text variable in the component file named *Employee Name*. This includes using unrecognized variable names in prompts, dialog element text, scripts, and so forth.
- **Incomplete instructions:** IF and REPEAT instructions used in the template or a script must have corresponding END IF and END REPEAT instructions. If HotDocs cannot find these instructions, it generates a syntax error when you assemble the document. (This includes test assembling from the template development window.) (You can have HotDocs match opening instructions with closing instructions. See Match Opening Instructions with Closing Instructions.)
- **Syntax errors in a script or expression:** Writing scripts in HotDocs requires you to use a specific "language" that HotDocs can recognize and then process. If you use the language incorrectly, or if you use words that are not part of the language, you will receive an error when you try to save your work.

HotDocs reports syntax errors when you try to test assemble a document and when you try to save a script that contains syntactical errors:

- When a syntax error occurs in a template, HotDocs displays a warning message that describes the
 error and displays the processing stack for the error. The stack shows, starting from the bottom,
 the components HotDocs was processing when it found the error. The Field denotes the variable
 or instruction field in the template where the error occurred, while Position denotes the character
 position within the field or script.
- To resolve such a syntax error, select an entry in the processing stack and click Go To Error. (Usually, you should select the topmost entry in the stack.) HotDocs takes you to the reference point so you can make the change. (When you click Go To Error, HotDocs continues to display the error message in a pop-up window, which you can leave open until you have corrected the error. When finished, click Close at the message box.)

You can click the **≣.Go to Field** button in the HotDocs toolbar to quickly move to a variable or instruction field containing an error.

• When a syntax error occurs in a script, HotDocs displays an error message and then moves the cursor as close to the error in the script as it can.

For more information on writing scripts, see Understand the HotDocs Scripting Language, Introduction: Instruction and Expression Models, and Customize a Computation Variable.

Using HotDocs Debugger

Introduction: Debugging Templates

When test assembling templates, you may find that some scripting or automation in your template isn't producing the result you expect. Perhaps you have created a computation script that is producing an answer you think is wrong. To help you troubleshoot situations like this, you can test your templates, resources, or scripts in Debug mode. You do this by inserting a DEBUG instruction in the template, resource, or script you think may be causing the unexpected result. Then, you test it using an answer file. When HotDocs processes the DEBUG instruction, it brings up the HotDocs Debugger so you can step through the script line by line, examining the components and any answers that may be causing the problem. The Debugger allows you to observe how a script is being processed or how a document is being assembled.

While debugging your templates, resources, or scripts, there are several options you can use to help you diagnose the problem. For example, you can add variables to a "watch list," which allows you to track how answers to certain variables change depending on how other questions are answered in the interview.

The Debugger also displays the processing stack which shows, starting from the bottom, the templates and components HotDocs is processing at the field or line you are currently examining. For example, if you are debugging a computation script in an inserted template, the processing stack would include the name of the parent template, the name of the inserted template, and the name of the Computation variable. Viewing the stack can help you understand what part of the interview you're viewing as well as provide the path for how you got to it.

You can use the information you gather from debugging to correct problems with automation.

At a Glance: The Debugger Window

At a Glance Debugger Window

🐑 HotDocs Debugger - Demo	o Employment Ag	greement.rtf					×
A़∓≣ Step Into 📮 Step Over	🟝 Step Out 🛛	Continue	 Stop 	0			
		«DEBUG»					-
		B					
	EMPLOY	MENTAGR	EEMEN	т			
This Agreement, by and between Hobble Creek Publishing and «Employee Name», is made and entered into this «Agreement Date:third day of June, 1990». The parties recite that:				d			
A. Hobble Creek Publishing is engaged in the business of publishing books and magazines, and maintains business premises at 1265 W. Canyon Pkwy, Mapleton, UT 84210.					d		
B. «Employee Name» is willing to be employed by Hobble Creek Publishing, and Hobble Creek							
B. «Employee Name» is w	liling to be emp	loyed by Hobi	ble Creek H	Publis	hing, and H	lobble Creek	
B. «Employee Name» is w	liling to be emp	loyed by Hobi	ble Creek H	Publis	hing, and H	lobble Creek	
B. «Employee Name» is w Generating an interview	illing to be emp	loyed by Hobi	ble Creek H	Publis	hing, and H	lobble Creek	•
B. «Employee Name» is w Generating an interview Variable		r	ble Creek H	Publis	hing, and H	lobble Creek	•
B. «Employee Name» is w ✓ Generating an interview Variable Employee Name 	/ Answe	r wered)	ble Creek H	Publis T	hing, and H ype ext	lobble Creek	•
B. «Employee Name» is w Generating an interview Variable Employee Name	/ Answe	r wered)	ble Creek H	T	hing, and H	lobble Creek	
B. «Employee Name» is w Generating an interview Variable Employee Name	Answe	r wered) G	ble Creek H	T T	hing, and H	lobble Creek	
B. «Employee Name» is w Generating an interview Variable Employee Name Default Watch List Custor	Ming to be empiried of the empirication of the	r wered) G Processing Sta	ck	T T	hing, and H	lobble Creek	

After inserting a **DEBUG** instruction in your template or script you can view the **Debugger Window** by clicking the **Test Assemble** button on the HotDocs ribbon and then the **Debug** button at the **Assembly Window** toolbar.

At the top of the Debugger Window is the Debugger toolbar A where you can find the following options:

- **Step Into:** Causes the Debugger to step into the current field or line of script to show how HotDocs has processed it using the answers you have provided.
- **Step Over:** Causes the HotDocs Debugger to not step through any computation or subtemplate referenced in the field unless it encounters another DEBUG instruction.
- ***Step Out:** Causes the HotDocs Debugger to move you out of the current level of debugging and move you to the next item on the processing stack.

- **Continue:** Instructs HotDocs to continue processing the interview until it either finds another explicit DEBUG instruction (which will cause the Debugger to appear again) or it finishes processing.
- **Stop:** Causes the HotDocs Debugger to close and returns you to the test assembly window. Also causes HotDocs to disable the Debugger. To enable it again, click the Enable Debugging button.
- **We Help:** Opens the relevant page of the HotDocs Help File.

Below that is the preview pane B where you can see HotDocs Step through the template or script and highlight in red the variable you are currently viewing. The bottom half of the window c can display the **Default Watch List**, **Custom Watch List**, or **Processing Stack** depending on which tab is selected.

To learn more about using the debugger window follow the links below:

- Introduction: Debugging Templates
- Insert Debugging Instructions in Templates and Scripts
- Step Through a Template or Script

Insert Debugging Instructions in Templates and Scripts

To debug your template or script, you must first insert a DEBUG instruction in the template, resource, or script. Then, you must test assemble the template (or test the resource or script). Finally, you must enable debugging in the test assembly window.

To debug your template or script using a certain set of answers, first test assemble the template using those answers and then save your answers in a test answer file. (See Use a Test Answer File.) Once you have done this, then complete the steps below.

You can insert a DEBUG instruction:

- In a text template.
- In a form template.
- In a script.
- In a resource.

To insert a DEBUG instruction

- 1. Open the text or form template for editing. (See Edit a Template.)
- 2. Complete the following steps, depending on where you need to insert the DEBUG instruction:

• If you are inserting a DEBUG instruction in a *text template*, place your cursor where you want to begin debugging. Click the **HotDocs** menu and choose **Other Field** from the list of options. Choose **DEBUG** from the **Field type** drop-down list . Click **OK**.

For WordPerfect Users: The **HotDocs** menu does not appear in WordPerfect. To insert a debug statement, manually type «DEBUG» into the template.

- If you are inserting a DEBUG instruction in a *form template*, create a field on the form where you want to begin debugging. Select the field and click the Area Field Properties button. In the Variable field, type DEBUG. (Close the Field Properties dialog box when you are finished.)
- If you are inserting a DEBUG instruction in a *script*, edit the computation or dialog whose script you want to debug. Insert the **DEBUG** instruction at the place in the script where you want to begin debugging.
- If you are inserting a DEBUG instruction in a resource, edit the component whose resource you want to debug. Insert the **«DEBUG»** instruction at the place in the resource text where to want to begin debugging.
- 3. Click the **Test Assemble** button. (If debugging a resource or script, click the **Test** button.) The test assembly window appears, along with the **HotDocs Debugger** window, which stops at the field following the DEBUG instruction. (If the Debugger window does not appear, make sure debugging is enabled by clicking the **Enable Debugging** button in the assembly window toolbar.)
- 4. Step through the template, resource, or script using the Debugger. (See Step Through a Template or Script.)

Step Through a Template or Script

Once you have inserted a DEBUG instruction in a script, a resource, or in the template, you can begin stepping through the template, resource, or script to determine where your automation is causing unexpected results. The HotDocs Debugger includes several options for doing this.

Before debugging, however, you should understand how HotDocs processes the template, resource, or script:

When you first test assemble a template, resource, or script, HotDocs processes (or runs through) the entire template, resource, or script to create the interview, which it displays in the test assembly window. As you answer questions in the interview, some answers may cause HotDocs to reprocess the script so it can update the interview. (For example, answering a true/false question may gray or ungray other variables in a dialog. In order for these changes to happen, HotDocs must reprocess the template.) Depending on the complexity of your template, HotDocs may reprocess the template, resource, or script many times during a test assembly.

During a test, if HotDocs encounters a DEBUG instruction, it displays the HotDocs Debugger, which simply shows you how the template, resource, or script is being processed at that specific field or line using any answers that have been entered. This allows you to see how variables and instructions are being used to manipulate text in the document or data in the script or resource. Such analysis will help you understand why you may be seeing unexpected results in your test.

At any time, you can close the Debugger and return to the test assembly window. However, as long as you are in debugging mode, any time HotDocs processes a DEBUG instruction, the HotDocs Debugger will be redisplayed. This will happen each time you make a change to answers in the interview that cause it to be reprocessed.

Debugging mode happens when you have a DEBUG instruction in your script or template and the **Enable Debugging** button is selected (pressed down) at the test assembly window.

To debug your template, resource, or script using a certain set of answers, first test assemble and then save your answers in a test answer file. (See Use a Test Answer File.) Then debug your template. (See Insert Debugging Instructions in Templates and Scripts.)

To step through a template or script

- 1. Insert a DEBUG instruction in the script, resource or template and then test assemble the template. (See Insert Debugging Instructions in Templates and Scripts.)
- 2. When the HotDocs Debugger window appears, complete any of the following tasks:

То	Do This
Understand why the Debugger window is showing	Read the explanation at the top of the tabbed pane.
Examine either each field in the template or each line in the script	Click Step Into . The Debugger steps into that field or line to show how HotDocs has processed it using the answers you have provided, specifically:
	 If you are in an INSERT instruction, the Debugger will step into the inserted clause, clause library, or template and show you how that file is being processed.
	 If you are in a Computation variable field, the Debugger will step into the computation and show you how the script is being processed.
	 If you are in a regular (non-computed) variable field, the Debugger will move you to the next field or line.
	As you step into components and instructions, you will notice that you are adding these levels of processing to the processing stack (which is a sequential list of templates and components you are processing).

Ignore a field or line in a script	Click Step Over . The Debugger will not step through any computations or subtemplates referenced in the field unless it encounters another explicit DEBUG instruction.
	For example, if you step over a computation field but the computation script includes a DEBUG instruction, HotDocs will stop at the DEBUG instruction in the computation script.
Move to the next item on the processing stack	Click EStep Out . The Debugger will move you out of the current level of debugging and move you to the next item down on the processing stack (which is a sequential list of templates and components you are processing).
	For example, if you are debugging a computation in an inserted template, clicking Estep Out will move you out of the computation and back to the inserted template. If you click Estep Out again, the Debugger will return you to the parent template.
Stop processing any current DEBUG instructions	Click Continue . HotDocs continues processing the interview until it either finds another explicit DEBUG instruction (which will cause the Debugger to appear again) or it finishes processing the template.
	Changing answers in the interview may cause HotDocs to reprocess the interview, thus triggering another debug session. To keep this from happening, close the Debugger window and either remove the DEBUG instruction from the template or script, or disable the Debugger. (To do this, click the Enable Debugging button in the test assembly window toolbar.)
Cancel the debugging session	Click Stop . HotDocs closes the Debugger window and disables the Debugger. You are returned to the test assembly window.
	You can start the Debugger again by clicking the Enable Debugging button in the test assembly window toolbar.
View the current field or line of script and any answers used in	Click the Default Watch List tab, which shows a list of variables referenced in the field or script as well as any answers for the variables.
processing it	As HotDocs processes REPEAT instructions, it displays the value for the current iteration (or index) in the Answer column. As you step through the REPEAT instruction, you can watch the Debugger increment this index. The Debugger also displays the REPEAT COUNTER.
Track the answers for specific variables as	Click the Custom Watch List tab. See Add Variables to the Debugger Watch List for details.

the Debugger steps through the interview or script	
View the processing stack, or path of execution	Click the Processing Stack tab. The processing stack shows the sequential list of templates and components you are processing as well as the reason why a certain template or component was added to the stack. This can help you determine how HotDocs came to be processing the current field or line. As you step into components, those components are added to the top of the stack.
Keep HotDocs from debugging the template or script without removing the DEBUG instruction	Click Stop at the Debugger window. This closes and disables the Debugger window. (To start another debugging session, click the Enable Debugging button. HotDocs reprocesses the interview or script, which forces HotDocs to process the DEBUG instruction, thus causing the Debugger to open again.)

At a Glance: The Add or Remove Variables dialog box

emo Employment Agreement.cmp - Add o	or Remo	ve Variables		? <mark>-</mark> ×
Variables:		Variables to <u>w</u> atch:		
All Variables 🗛 👻]			
 Agreement Date Annual Salary Company Representative Employee Gender Employee Name Employee to Complete Trial Period Employee to Receive Paid Seminar Days Employment Status A f Full-time Annual Salary Hourly Salary Job Description Job Title Number of Seminar Days Part-time Annual Salary Number of Vacation Days Part-time Annual Salary X Probation End Date Signature Date Start Date 	G		D	
Eind:]		ж	Cancel

After clicking the **Custom Watch List** tab at the **Debugger window**, right click and select **Add or Remove Variables**, then the **Add or Remove Variables** dialog box appear. Using this dialog you can edit which variables are on your watch list.

At the top of the left hand side of the dialog is a drop-down list \underline{A} , from this you can select the type of variables that will appear in the variable list \underline{B} below.

To add a variable to the watch list you first click on it in the variable list then click the arrow button \Box in the center. To remove a variable from the watch list you click on it in the watch list D and click the arrow in the center to move it back.

You can also search the variable list using the **Find** field **E** at the bottom of the dialog.

To learn more about using the watch list follow the links below:

• Add Variables to the Debugger Watch List

Add Variables to the Debugger Watch List

You can have the Debugger monitor how answers to specific variables change as you step through the interview or script. You do this by adding variables to a custom watch list.

When you add variables to the watch list, they will be associated with the list until you manually remove them. Also, the custom watch list contains variables for only the current component file. In other words, if you are debugging a template that has an inserted template, you can actually have two watch lists—one for variables in the parent template and one for variables in the inserted template.

To add variables to a watch list

- 1. At the **Debugger** window, click the **Custom Watch List** tab. The tab changes to show the list of specific variables you want to watch.
- 2. In the **Watch** pane, right-click and choose **Add or Remove Variables** from the shortcut menu. The **Add or Remove Variables** dialog box appears, showing a list of all the variables in the current component file. (You can filter this list by clicking the **Variables** drop-down button and choosing a variable type.)
- Select a variable and click the Add Variables button. The variable is added to the Variables to watch list.
- 4. Add as many variables as you want to monitor and click **OK** when you are finished. The variables are added to the **Custom Watch List** tab.

To remove a variable from the custom watch list, select the variable and click the **Remove Variables** button. You can also select a variable at the **Watch** pane, right-click, and choose **Remove Selected Variable** from the shortcut menu.

Using the Text Panel

Introduction: Use the Test Panel

You can display logistical information about the template you are testing in the Test Panel. This includes viewing a list of variables that are asked in the interview but not used in the document (and vice versa), and identifying situations where you've improperly asked or set variable values. Finally, it includes linking from the **Document Preview** tab of the assembly window back to a specific place in a Microsoft Word template.

Once you open the Test Panel, you can leave it displayed as you work in the test assembly window. Changes you make in the interview are automatically reflected in the Test Panel.

Validate Variable Usage in a Template

One key to good template automation is making sure the variables asked in the interview are both relevant to the interview and to the document. Specifically, you must ensure you're not asking questions in the interview that aren't used in the document, and vice versa. If such a situation exists, the interview and/or document may be incorrect.

You can use the Test Panel to examine which variables are asked in the interview (but not used in the document) as well as see which variables are not asked in the interview but are used in the document.

Variables listed in these tables appear based on the current answer set. For example, if a variable is included in conditional text and the condition has not yet been resolved, that variable will appear in the **Variables asked in interview but not referred to in document** list. Additionally, if a variable is merged in the document but it is set to **Don't ask automatically** and it's not specifically asked in a dialog, it will appear in the **Variables referred to in document but not asked in interview** list.

You can use this tool with both text and form templates.

To view variable usage

- 1. Test assemble the document. (See Test Assemble a Document.)
- 2. At the test assembly window, click the &**Test Panel** button. The **Test Panel** window appears.
- 3. Click the **Variable Usage** tab. The view changes to show the validation options.
- 4. Select Enable variable usage tracking. This generates the variable lists.
- 5. Review the usage lists and make any necessary corrections to the template or interview script.
- 6. Optionally, to review the variable's properties, including seeing which dialog links to it, doubleclick the variable.

At a Glance: The Test Panel Window

🗞 Test Panel 🔹 💌
Variable Usage Warnings Go to Template
A Enable variable usage tracking (selecting this slows interviews)
Variables asked in interview but not referred to in document
6
Variables referred to in document but not asked in interview
G

When test assembling a document you can open the **Test Panel** window by clicking on the \mathcal{L} **Test Panel** button of the **Assembly** window tool bar. When it opens you will see the **Variable Usage** tab.

At the top of the tab you have a check box A that you can click to enable variable usage tracking to see if there are any issues with the variables that have or haven't been included in the interview.

The first list of variables B will show any variables that are in the interview but have not been used in the document, making them unnecessary questions. The second list of variables c will show any variables that are in the document but have not been asked in the interview, meaning they will remain unanswered on the final assembled document.

At the top of the dialog there are two other tabs next to **Variable Usage**. You can click on the **Warning** tab to view a list of warnings in the interview and go to them, you can also set options on what causes HotDocs to generate a warning.

You can click on the **Go To Template** tab to choose whether or not to enable the go to template function.

To learn more about using the test panel window follow the links below:

• Introduction: Use the Test Panel

- Validate Variable Usage in a Template
- Check Interviews for Improper Scripting
- Move From the Document to the Template During Testing

Check Interviews for Improper Scripting

When assembling documents in HotDocs, HotDocs displays an interview outline in the assembly window. This outline is dynamic, in that as users enter answers, HotDocs processes the answers and updates the outline as necessary. Because of this, some scripting may cause the interview to process incorrectly.

Potential problems include SET instructions that have been used incorrectly, and variables that have been asked or referred to incorrectly. During testing, HotDocs identifies these problems and displays the relevant warnings in the **Warnings** tab of the **Test Panel**. You can go directly from a warning to the problem in the template and make changes.

Many of these warnings pertain to variables being asked incorrectly in the interview. To understand the circumstances under which a variable or dialog is asked, it is important to note that when the **Ask automatically** property is selected for a variable, HotDocs will ask the variable 1) when the answer is used (for example, tested in an IF instruction or merged into text) and 2) when the variable hasn't been asked before in the interview, either by itself or as part of a dialog. Additionally, a dialog is asked automatically when 1) the dialog has not been asked before in the interview and 2) when a variable to which the dialog is linked is asked automatically. Also, a dialog will be asked automatically when it is used in a REPEAT instruction.

See Warnings Tab Descriptions for a description of each warning.

To test a template for correctly produced interviews

- 1. At the template you are editing, click the **Test Assemble** button. HotDocs displays the test assembly window.
- 2. Answer questions in a way that causes the interview to update differently based on your answers.
- 3. Click the &**Test Panel** button. The **Test Panel** window appears.
- 4. Click the **Warnings** tab. HotDocs displays a window that contains items for which you can check, as well as a list of actual problems that were found when HotDocs processed the interview.
- 5. Select a warning from the **Warnings in interview** list and click **Go to Warning**. HotDocs takes you to the place in the template where the potential problem exists. If the problem is in a component, HotDocs opens the component editor so you can make the changes.

If you receive an error that includes a field number, you can quickly go to that specific field in the template by clicking the \blacksquare -Go To Field button in the HotDocs Edit toolbar. See Go to a Specific Field in the Template for details.

Move From the Document to the Template During Testing

The Go to Template command is supported in Microsoft Word templates only.

When you are test assembling a document, you sometimes need to return to the template to see how automation in the template affects the portion of the assembled document you are reviewing. Rather than manually search for that section of text in the template, you can use the **Go To Template** command in the Test Panel to move from a specific place in the assembled test document to the template.

When using this command, you must test assemble the entire document. The command will not work properly if you test assemble just a portion of the template.

To go to a specific location in the template from the test assembly window

- 1. Test assemble the document. (See Test Assemble a Document.)
- 2. At the test assembly window, click the &**Test Panel** button. The **Test Panel** window appears.
- 3. Click the **Go to Template** tab. The view changes to show the options available.
- 4. Select Enable Go to Template.
- 5. Click the **Document Preview** tab to view the assembled document.
- 6. Place your cursor in the document where you want to review the corresponding automation and click **Go to Template**. The template window comes to the front and inserts your cursor at the specified location.

Selecting **Enable Go to Template** inserts several hidden field markers in the assembled document that allow HotDocs to find the correct location in the template when you click **Go to Template**. (To see these markers, you can right-click in the assembled document and choose **Show Codes**.) If you send a test assembled document to Word with this option enabled, the field markers will appear (albeit hidden) in the new Word document. If you don't want these hidden markers in the document, clear this option before sending the document to Word. (Of course, these markers will not appear in a document assembled regularly.)

Publishing Templates

Publishing HotDocs Templates

Introduction: Publish HotDocs Templates

Publishing is a convenient way to distribute template files to others. When you publish a template, the related files (component files, clause library files, and so forth) are automatically compressed with the template file into one file.

There are four types of files you can publish, including:

- **Standard template files.** You can publish a regular template file and copy all the related files to the correct location. You can also apply protection options to the files.
- **Auto-install files.** You can publish a template or template set that, when accessed by the user, creates a new library of templates or updates an existing library for the user.
- **Auto-assemble files.** You can publish a template or template set that will automatically assemble when the user accesses the file.
- **HotDocs Server files.** You can publish the files needed to assemble documents using HotDocs Server.

All of these files can be uploaded to a Web server using your own custom uploading method. (See Introduction: Upload HotDocs Files to a Web Server.)

The HotDocs publishing process offers several ways for you to protect your published files. Also, at the end of the process, you can save your publishing preferences in a HotDocs publish settings (.HDP) file. These files are saved to your HotDocs *Publish* folder and can be used the next time you publish your templates.

At a Glance: Publishing Options (The Publishing Wizard)

Publishing Wizard	? 💌
Publishing Options	
A Publish as	
Standard template files	
Individual auto-install files (HDI)	
Individual auto-assemble files (HDA)	
Template set in a single auto-install file (HDI)	
Template set in a single auto-assemble file (HDA)	
Template files for use with HotDocs Server	
Inglude templates used in INSERT and ASSEMBLE instructions Make published files compatible with HotDocs 2006-2008 Local <u>f</u> older for published files: C:\Users\HolliCooper.HOTDOCS\DocumCs\HotDocs\Templates	•
Web destination for published files:	
None	
E Load Settings < Back Next >	Cancel

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

In the **Publishing Options** Dialog you will first be able to choose from a list of 6 options A:

- **Standard Templates:** Publishes the selected templates as standard template files. The template and component files will be copied to the same folder and protected with any protection options you select.
- Individual auto-install files (HDI): Compresses and publishes the selected template, related files, and HotDocs library as individual auto-install (HDI) files. When HotDocs is passed an auto-install file and the library already exists in the library folder, HotDocs either updates the existing content or creates new library entries with the contents of the .HDI.
- Individual auto-assemble files (HDA): Compresses and publishes the selected template or templates as individual auto-assemble (HDA) files. When HotDocs is passed an auto-assemble file, it "unpacks" all the files and starts assembling. When it finishes, HotDocs deletes the files.
- **Template set in a single auto-install file (HDI):** Compresses and publishes multiple templates, related files, and HotDocs library as an auto-install (HDI) file. When HotDocs is passed an auto-install file and the library already exists in the library folder, HotDocs either updates the existing content or creates new library entries with the contents of the .HDI.
- **Template set in a single auto-assemble file (HDA):** Compresses and publishes multiple templates as one auto-assemble (HDA) file. When HotDocs is passed an auto-assemble file, it decompresses all the files and starts assembling each document as if the user had selected multiple templates at the library. When it finishes, HotDocs deletes the files.
- **Template files for use with HotDocs Server:** Publishes the HotDocs Server files required for assembling a template on a server. The files generated include the template file (.WPT, .HPT, .DOCX, or .RTF), the component file (.CMP), the template manifest (.manifest.xml), and any additional files (typically graphics) as necessary. Before publishing, you must enable the template for use with HotDocs Server. To do this, make the change at Component File Properties.

Below this list are two check boxes **B**. Check the first box to have HotDocs include templates used in INSERT and ASSEMBLE instructions and check the second box to make the published files compatible with HotDocs 2006-2008.

Next there is a text field G where you can enter the file path for the folder where you would like to save the published templates. You can click on the arrow at the right of the field to see a drop down list of recently used folders or you can use the Rowse button to the right to navigate to the folder you need.

Below this is a drop-down list **D** where you can choose a web destination if you are uploading the published files to the web. You can also use the **SECTION DESTINATION** button to the right to open the **Web Destination** dialog box and select a new URL.

At the bottom left of this dialog is the Load Settings button **E**. If you click on this you can navigate to previously saved publishing settings (see Save Publishing Settings).

You can move along to the next dialog by clicking the **Next** button.

To learn more about publishing options follow the links below:

- Introduction: Publish HotDocs Templates
- Publish a Template as a Standard File
- Publish Templates as Auto-Assemble Files (HDA)
- Publish Templates as Auto-Install Files (HDI)
- Publish Templates For Use With HotDocs Server

At a Glance: Published Template Set (The Publishing Wizard)

Publishing Wizard			? 💌
Published Template Set —			
File name for published HDI file:			
	A		
Template set title:			
	0		
	< <u>B</u> ack	Next >	Cancel

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

In the **Published Template Set** dialog you can select options for saving your published library.

In the top text field $\frac{A}{A}$ you need to type in a name for the published file and in the bottom text field $\frac{B}{B}$ you need to type in a title for the template set.

You can move along to the next dialog by clicking the **Next** button.

To learn more about publishing template sets by following the links below:

- Publish Templates as Auto-Assemble Files (HDA)
- Publish Templates as Auto-Install Files (HDI)
- Automatically Update Published Template Sets

Publish a Template as a Standard File

You can publish a template as a standard template file to distribute among other users, including other users who will access it from the Internet or intranet. (See Upload Templates to a Web Server.) Publishing a template ensures that all associated files are copied to the same folder. It also allows you to protect the files.

You can also publish multiple template files as standard files; however, when you publish multiple templates, they will not be grouped as a set (as they are if you publish an auto-install or auto-assemble file.)

Additionally, you can publish your templates for use with HotDocs 2006-2008. This allows you to keep your templates backwards compatible. If you select this option, HotDocs opens and saves the component file in that version.

To publish templates as standard template files

- At the HotDocs library, select the template or templates you want to publish and click the Publishing Wizard button. The Publishing Wizard dialog box appears, displaying the Publishing Options information.
- 2. Select **Standard template files** from the **Publish as** group.
- 3. If your template contains any INSERT or ASSEMBLE instructions, select **Include templates used in INSERT and ASSEMBLE instructions**.
- 4. If you want to make the published template(s) compatible with HotDocs 2006-2008, select **Make published files compatible with HotDocs 2006-2008**.
- 5. Click the **Browse** button next to the **Local folder for published files** field to specify the folder where you want HotDocs to save the published files.
- 6. Optionally, if you plan to upload the published files, click the **S Edit** button next to the **Web destination for published files** field and specify the URL of the server to which you want to upload. (When HotDocs finishes publishing the templates, it will start the uploading process. See Introduction: Upload HotDocs Files to a Web Server.)
- 7. Click Next. The File Security dialog box appears.
- Optionally, specify any security options and click Next. (See Protect Published Files for details.) The Additional Files dialog box appears.
- 9. If you want to include additional files (such as resource files), click **Add Files**, then select the desired files. (See Add Additional Files to a Published Set for details.)
- 10. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

If you publish multiple template sets, you can save the settings you specify during the publishing process so that the next time you publish, the options are already specified. (See Save Publishing Settings.)

Once these files are published to this temporary folder, you should move them to a more permanent location. If you publish additional templates or other files that use the same file names, HotDocs will overwrite the previously published files.

Publish Templates as Auto-Assemble Files (HDA)

A HotDocs auto-assemble (.HDA) file is a compressed file containing a template and its related files (component files, clause library files, and so forth.) When you assemble an auto-assemble file, HotDocs copies the template and related files into a temporary folder, then begins assembling. After assembly finishes, HotDocs deletes the files in the temporary folder. This is useful if you are distributing templates on the Internet, an intranet, or on a network and you don't want users to have copies on their local drives. (See Introduction: Upload HotDocs Files to a Web Server.)

If you select multiple library items to publish, you can either publish each item as its own auto-assemble file, or you can group items together in a template set. When an auto-assemble file contains a group of templates, assembly proceeds as if the user had selected multiple templates for assembly at the HotDocs library window.

Additionally, you can publish your templates for use with HotDocs 2006-2008. This allows you to keep your templates backwards compatible. If you select this option, HotDocs opens and saves the component file in that version.

To assemble an auto-assemble file, you can either pass the file to HotDocs using a command-line option, or you can add the auto-assemble file to a HotDocs library. If users access an auto-assemble file while browsing the Internet or an intranet, the browser will start HotDocs and pass it the auto-assemble file. If you are using a Web browser other than Internet Explorer, you may need to register HotDocs to work with the Web browser. (See Register HotDocs to Work with Web Browsers.)

To publish templates as auto-assemble files

- At the HotDocs library, select the template or templates you want to publish and click the Publishing Wizard button. The Publishing Wizard dialog box appears, showing the Publishing Options information.
- 2. Select one of the following from the **Publish as** group:
 - Individual auto-assemble files (HDA)
 - Template set in a single auto-assemble file (HDA)
- 3. If your template contains any INSERT or ASSEMBLE instructions, select **Include templates used in INSERT and ASSEMBLE instructions**.
- 4. If you want to make the published template(s) compatible with HotDocs 2006s-2008, select **Make** published files compatible with HotDocs 2006-2008.
- 5. Click the **Browse** button next to the **Local folder for published files** field to specify the folder where you want HotDocs to save the published files.
- 6. Optionally, if you plan to upload the published files, click the **Section Edit** button next to the **Web destination for published files** field and specify the URL of the server to which you want to

upload. (When HotDocs finishes publishing the templates, it will start the uploading process. See Introduction: Upload HotDocs Files to a Web Server.)

- 7. Click Next. The File Security dialog box appears.
- 8. Optionally, specify any security options and click **Next.** (See Protect Published Files for details.)(If publishing individual auto-assemble files, skip to step 9.)
- 9. Enter a name for the HDA file in the **File name for published HDA file** field, and then enter a title for the file in the **Template set title** field.
- 10. Click **Next**. The **Additional Files** dialog box appears.
- 11. If you want to include additional files (such as resource files), click **Add Files**, then select the desired files. (See Add Additional Files to a Published Set for details.)
- 12. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

If you publish multiple template sets, you can save the settings you specify during the publishing process so that the next time you publish, the options are already specified. (See Save Publishing Settings.)

Once these files are published to this temporary folder, you should move them to a more permanent location. If you publish additional templates or other files that use the same file names, HotDocs will overwrite the previously published files.

At a Glance: Auto Install Target Library (The Publishing Wizard)

🗋 Publishing Wizar	1	? 🛃
Auto-Install Tar	jet Library	
Library <u>fi</u> le name:		
PUB	A	
Library <u>t</u> itle:		
	в	
Library <u>d</u> escription:		
	G	*
		-
Update library title	and description even if library already on without installing templates	exists
Install templates f	or read-only access	
	< Back	ext > Cancel

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

In the Auto-Install Target Library dialog you can select options for saving your published library.

In the first text field \boxed{A} you can enter a file name for the library, in the second field \boxed{B} you can enter a title for the library (this will appear at the top of the template list in the when the library is opened). In the third field \boxed{C} you can enter an optional description of the library.

Below the text fields is a series of three check boxes **D**. Check the first box to over-write an existing library title or description, check the second box to update the library items without installing the templates and check the third box to install the templates for read-only access.

You can move along to the next dialog by clicking the **Next** button.

To learn more about setting options for the auto-install target library follow the links below:

- Publish Templates as Auto-Install Files (HDI)
- Create an HDI File for a CD-based Template Set

Publish Templates as Auto-Install Files (HDI)

A HotDocs auto-install (.HDI) file is a compressed file containing a library and one or more templates and their related files (component file, clause library file, and so forth). Auto-install files are a convenient way to distribute template sets to users, especially sets that must be updated periodically.

When you create an auto-install file, you specify a library file name. The first three characters of the file name are automatically assigned the characters, **PUB**, which sets a reference path keyword for the library. The first time a user installs a template set with that library file name, HotDocs suggests to the user that the library for the templates be installed in the *Template Sets* folder or the user's *Libraries* folder. HotDocs then asks where the templates should be installed. Any subsequent template sets with that same library file name will automatically get installed to the same folders. The paths for the templates in the library will appear as *^library file name\library file name\template file name*.

If you select multiple library items to publish, you can either publish each item as its own HDI file, or you can group items together in a template set.

Additionally, you can publish your templates for use with HotDocs 2006-2008. This allows you to keep your templates backwards compatible. If you select this option, HotDocs opens and saves the component file in that version.

To assemble an auto-install file, you can either pass the file to HotDocs using a command-line option, or you can add the auto-install file to a HotDocs library using the **Install Templates** command (**File** menu at the HotDocs library). If users open an auto-install file from the Internet or an intranet, the browser will start HotDocs and pass it the file.

When HotDocs is passed an auto-install file, it installs a library, which contains the templates you published. If the library already exists, HotDocs updates it, overwriting existing library items and adding new items. If the library doesn't already exist, HotDocs creates it.

To publish templates as HotDocs auto-install files

- At the HotDocs library, select the template or templates you want to publish and click the Publishing Wizard button. The Publishing Wizard dialog box appears, displaying the Publishing Options information.
- 2. Select one of the following from the **Publish as** group:
 - Individual auto-install files (HDI)
 - Template set in a single auto-install file (HDI)
- 3. If your template contains any INSERT or ASSEMBLE instructions, select **Include templates used in INSERT and ASSEMBLE instructions**.
- 4. If you want to make the published template(s) compatible with HotDocs 2006-2008, select **Make published files compatible with HotDocs 2006-2008**.

- 5. Click the Browse button next to the Local folder for published files field to specify the folder where you want HotDocs to save the published files.
- 6. Optionally, if you plan to upload the published files, click the **Section** Edit button next to the Web destination for published files field and specify the URL of the server to which you want to upload. (When HotDocs finishes publishing the templates, it will start the uploading process. See Introduction: Upload HotDocs Files to a Web Server.)
- 7. Click Next. The File Security dialog box appears.
- 8. Optionally, specify any security options and click **Next.** (See Protect Published Files.) (If publishing individual auto-install files, skip to step 9.)
- 9. At the **Published Template Set** dialog box, enter a name for the HDI file in the **File name for published HDI file** field, and then enter a title for the file in the **Template set** title field.
- 10. Click Next. The Auto-Install Target Library dialog box appears.
- 11. Enter a library file name, title, and optional description for the library that will be created and select any of the following options:
 - **Update library title and description even if library already exists** updates the current library title and description to match the information in the library that is being installed.
 - **Update library items without installing templates** points the items in the library to templates on a CD. (The templates on the CD must be saved in a first-level folder with the same name as the library base file name (including **PUB**).)
 - **Install templates for read-only access** installs the templates as read-only, which prohibits users from editing them.
- 12. Click **Next**. If you're publishing a template set in a single auto-install file, the **Target Library Shortcut** dialog box appears. (If you're publishing individual auto-install files, skip to step 13.)
- 13. Optionally, select **Install a shortcut in the user's Start menu** and specify any shortcut options. (See Create a Shortcut for an Auto-Install Library.)
- 14. Click Next. The Additional Files dialog box appears.
- 15. If you want to include additional files (such as resource files), click **Add Files**, then select the desired files. (See Add Additional Files to a Published Set for details.)
- 16. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

Once these files are published to this temporary folder, you should move them to a more permanent location. If you publish additional templates or other files that use the same file names, HotDocs will overwrite the previously published files.

When installing templates for read-only access, you should understand that this option simply marks the files as read-only, so feasibly, users could change the file properties and then edit the templates, as long as they have HotDocs Developer. To keep users from editing the template, specify the **Lock Component Files** option at the **File Security** dialog box.

When updating an existing published template set that is in binary format, if you use any foreign characters in any of the properties of the template set, HotDocs will convert the existing library to XML format so that it can display the characters.

When updating the library, HotDocs looks to see if each item in the auto-install file's library exists in the user's library. If folder and template titles don't match, HotDocs adds new items instead of updating existing items.

If you give your libraries file names that are unique to your company or product, you will reduce the chance of another publisher creating a template set with the same name.

If you publish multiple template sets, you can save the settings you specify during the publishing process so that the next time you publish, the options are already specified. (See Save Publishing Settings.)

You can have a custom splash screen appear when the user launches the library created by the auto-install file. (See Use a Custom Library Splash Screen.)

At a Glance: File Security (The Publishing Wizard)

Publishing Wizard	? <mark>×</mark>
File Security	
A cock component files to prevent editing	
B Set expiration date for component files	
G Register component files	
Passw <u>o</u> rd:	
Password protect published files	
Pas <u>s</u> word:	
Password protect documents sent to Microsoft Word	
P <u>a</u> ssword;	
< <u>Back</u> <u>N</u> ext >	Cancel

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

In the File Security dialog you can select the security options for your published library.

In the first check box \underline{A} you can choose to lock the component files of the published templates to prevent editing and in the second check box \underline{B} you can choose to set an expiration date for the component files.

If you tick the third check box C to register component files then the password field below ungrays and you can enter the password for the publisher key file. (You should have received this password from the HotDocs Corporation publisher key administrator who gave you the publisher key file.)

If you tick the fourth check box \mathbf{D} to choose to password protect the published files you can specify the password you would like the user to enter in the password field below. This is the same when you tick the fifth check box \mathbf{E} to password protect documents sent to Microsoft Word

You can move along to the next dialog by clicking the **Next** button.

To learn more about setting options for your file security follow the links below:

- Publish a Template as a Standard File
- Publish Templates as Auto-Assemble Files (HDA)
- Publish Templates as Auto-Install Files (HDI)
- Protect Published Files
- Publish and Register Templates for Use With HotDocs Player

At a Glance: Component File Expiration (The Publishing Wizard)

Publishing Wizard		? <mark>×</mark>
Component File E	×piration	
Expiration <u>d</u> ate:	A	
Warning period: 0	🚔 days 🖪	
Extension period: 0	🚔 days 🖸	
Expiration message:		
	D	
	< <u>B</u> ack <u>N</u> ext >	Cancel

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

If you choose to set an expiration date for your component files in the **File Security** dialog then the **Next** button would take you to the **Content File Expiration** dialog where you can set these options.

In the first text field A you can type the expiration date you would like to give the component files, or you can click on the E Calendar button to the right of the text field to select a date.

Below this text field are two number fields. In the first field **B** you can use the arrows or type a number to set the amount of days prior to the expiration date HotDocs will display the expiration message to warn the user. In the second field **C** you can use the arrows or type a number to set the amount of days after the expiration date that the user will still be able to use the templates.

In the large text field D you can type in an expiration message that the user will see along side the HotDocs warning showing the expiry date. By default, when a user attempts to use a template that is either going to expire or has already expired, HotDocs automatically displays a warning message that gives these details, including the expiration date. You can include additional information that will appear in this message box -- for example, contact information for obtaining an updated template or template set.

You can move along to the next dialog by clicking the **Next** button.

To learn more about setting the component file expiration options follow the link below:

• Protect Published Files

Protect Published Files

During the publishing process, you can specify options that control users' access to the files, set expiration dates for the files, and assign passwords to them.

To protect published files

- 1. At the **File Security** dialog box of the Publishing Wizard, select the desired protection options:
 - Lock component files to prevent editing assigns an encrypted password to the component file, which keeps the user from editing it.
 - Set expiration date for component files makes the published templates usable only until a certain date. Selecting this option causes the **Component File Expiration** dialog box to appear when you click **Next**. (See Step 2 for details.)
 - **Register component files** allows you to register templates for commercial use with HotDocs Player. (To do this, you must obtain a special publisher key (.HDK) file from your HotDocs sales representative. Once this file is obtained (including a password to use it), the publisher key must be copied to the HotDocs program folder. Then, when you publish your template set, these options will be available. The resulting published set can be used with Player. See Publish and Register Templates for Use With HotDocs Player.)

If your users upgrade to HotDocs Developer and then attempt to edit any published templates you have registered, those templates will no longer work with HotDocs Player. To prevent users from modifying component files, and thus unregistering them, you should lock the component files. (See above.)

- **Password protect published files** requires users to enter a password before assembling the auto-assemble file or installing the auto-install file. You specify this password in the **Password** field.
- **Password protect documents sent to Microsoft Word** requires users to unprotect the document before modifying assembled documents that have been sent to Microsoft Word. (They do this by entering the password you used to publish the file.) (This can include documents that were assembled from WordPerfect templates.) You specify this password in the **Password** field.
- 2. If you selected **Set expiration date for component files**, the **Component File Expiration** dialog box appears. Select the desired expiration options:

- **Expiration date** lets you specify the date on which you want the files to expire. Type the date in the field, or click the calendar icon next to the field to select the date.
- **Warning period** lets you warn users that their files are about to expire. Click the up or down arrows to specify the number of days this warning will appear before the template actually expires and can no longer be used.
- **Extension period** lets users continue to use the files for a specified amount of time after the template or template set has expired. Click the up or down arrows to specify the number of days.
- **Expiration message** lets you include a custom message about the files that are set to expire. Enter your message in the text field. (HotDocs automatically includes information about the expiration date.)

Publish and Register Templates for Use With HotDocs Player

You can register your templates so they can be used with HotDocs Player. To do this, you must first obtain a publishing license from HotDocs Corporation. You will then receive a publishing key file (register.hdk), which allows you to register your templates for use with HotDocs Player. Registration happens during the publishing process. A publishing license also allows you to redistribute HotDocs Player at no cost to your users.

To publish your templates for use with HotDocs Player

- 1. Copy **Register.hdk** to the HotDocs program folder (for example, C:*Program Files**HotDocs*).
- 2. At the template library, select the templates you want to publish and click the **Publishing Wizard** button.
- 3. At the **Publishing Options** page of the wizard, enter all the required information for the template set you are publishing and click **Next**. The **File Security** page appears.
- Select Register component files and enter the password for the publisher key file in the Password field. (You should have received this password from the HotDocs Corporation publisher key administrator who gave you the publisher key file.)
- 5. Continue with the publishing process

Your templates are now registered and can be used with HotDocs Player.

To distribute HotDocs Player, either your users can download a version from the HotDocs Web site, or you can include it in the installation package with your template sets.

At a Glance: Target Library Short-Cut (The Publishing Wizard)

Publishing Wizard	? <mark>×</mark>
Target Library Shortcut	
A 🔽 Install a shortcut in the user's Start menu	
B Prompt for which folder should contain the shortcut	
Icon to use:	
G	
Eolder to contain shortcut:	
Label for shortcut:	
3	
< <u>B</u> ack Next >	Cancel

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

In the **Target Library Shortcut** dialog you can select options for creating a shortcut for your published library.

You can choose to install a shortcut to the published library in the users **Start** menu by ticking the check box A at the top of the dialog. This ungrays the rest of the options.

Ticking the second check box **B** will have HotDocs ask the user to confirm which folder in the **Start** menu should contain the shortcut.

To choose a custom icon for the shortcut you can type the icon file path into the text field **G** or use the **Browse** button to navigate to the file you would like to use.

In the next text field \mathbf{D} you can enter a name for the folder that will contain the library shortcut and in the last text field \mathbf{E} you can type in the label for the shortcut.

You can move along to the next dialog by clicking the **Next** button.

To learn more about creating a target library shortcut follow the links below:

- Publish Templates as Auto-Install Files (HDI)
- Create a Shortcut for an Auto-Install Library

Create a Shortcut for an Auto-Install Library

When you publish a template set in a single auto-install file, you can have HotDocs create a shortcut in the user's Start menu. The shortcut is created when the user installs the file, and, depending on the options you specify, the user can specify the exact location for the shortcut. You can also include your own custom icon that will be used when the shortcut is created.

To add a shortcut (including a custom icon) to the Start menu

- 1. At the **Target Library Shortcut** dialog box (which appears during the publishing of a template set in a single auto-install file), select **Install a shortcut in the user's Start menu**.
- 2. Optionally, select any of the following options:
 - Select **Prompt for which folder should contain the shortcut** to have HotDocs prompt users to specify where the shortcut will be placed.
 - Click the Rowse button next to the **Icon to use** field and locate the icon (.ICO) file you want to assign to the library. (If you leave this field empty, HotDocs will use a default icon.)
 - Type a name in the **Folder to contain shortcut** field. HotDocs will create a folder in the **Start > Program Files** menu using this name, and will save the shortcut there. (If you don't specify a name, HotDocs creates the shortcut in the *HotDocs* program folder.)
 - Type a shortcut name in the **Label for shortcut** field. The label is what identifies the shortcut in the **Start** menu.
- 3. Click **Next** and finish the publishing process.

Create an HDI File for a CD-based Template Set

If you are distributing a template set on CD and you want the templates to stay on the CD instead of being installed, you can create an auto-install file that will create the template set library on the user's computer and point the library entries to the templates on the CD.

Users can install the library by clicking **Install Templates** (File menu at the HotDocs library window) and selecting the .HDI file.

To create an auto-install file for a CD-based template set

- 1. Publish the template set in a single .HDI file. (See Publish Templates as Auto-Install Files (HDI).)
- 2. At the **Auto-Install Target Library** dialog box, note the library file name (you will need this name in Step 4), and select **Update library items without installing templates**.
- 3. Finish publishing the template as you normally would.
- 4. When you put the templates and their associated files on the CD, put them in a first-level folder with the same name as the library base file name (including **PUB**) you specified in Step 2. Make sure all associated files and all inserted files are included on the CD.

At a Glance: Additional Files (The Publishing Wizard)

Publishing Wizard		? 💌
Additional Files		
	•	
	A	
Add Files Remove Files		
Save Settings	< Back Finish	Cancel
Encoccange		Concer

After opening **Publishing Wizard** from the toolbar (or **Tools** menu) in your HotDocs Library you will be presented with a series of dialogs containing different options for publishing templates and template sets.

In the **Additional Files** dialog you can add extra files that need to be published with the library.

In the main part of this dialog \boxed{A} you will see a list of any additional files already added to your library. To add or remove files you can use the buttons \boxed{B} at the bottom left of this field. For example, this can include a product splash screen bitmap you want to associate with an auto-install file.

As this is the last page of the **Publishing Wizard** you will see at the bottom left of the dialog is the **Save Settings** button . You can click on this to save the options you have been selecting as a **HotDocs Publish Settings File** so you can use it in the future to save time when publishing the same or similar libraries.

You can end the **Publishing Wizard** and start to publish the files by clicking the **Finish** button.

To learn more about adding additional files follow the links below:

- Publish a Template as a Standard File
- Publish Templates as Auto-Assemble Files (HDA)
- Publish Templates as Auto-Install Files (HDI)
- Add Files to a Published Set
- Save Publishing Settings
- Use a Custom Library Splash Screen

Add Files to a Published Set

As you are publishing templates and template sets, you may want to include additional files, such as answer files, resource files, or bitmap files used for custom library splash screens. You can include any of these files by adding them at the **Additional Files** dialog box.

To add files to a published template set

- 1. Publish your templates or your template set. (See Publish a Template as a Standard File, Publish Templates as Auto-Assemble Files (HDA), or Publish Templates as Auto-Install Files (HDI).)
- At the Additional Files dialog box, click Add Files. The Select Additional File(s) dialog box appears.
- 3. Browse to the folder that contains your additional files, select the files, and click **Select**. HotDocs lists the files in the field.
- 4. Optionally, select a file and click Remove Files to remove a file from the list.
- 5. Click **Finish** to finish the publishing process.

Save Publishing Settings

Any options you select during the publishing process can be saved to a published settings (.HDP) file and used again the next time you publish a template or set of templates.

To save the settings you used during publishing

- 1. Publish your templates or your template set. (See Publish a Template as a Standard File, Publish Templates as Auto-Assemble Files (HDA), or Publish Templates as Auto-Install Files (HDI).)
- 2. Before clicking **Finish** at the **Additional Files** dialog box, click **Save Settings**. HotDocs displays the **Save Publish Settings As** dialog box.
- 3. Type a name for the file in the **File name** field. HotDocs adds the extension **.HDP** to the file.
- 4. Click **Save**. The file is saved to the HotDocs *Publish* folder.

To use the publish settings file, at the first dialog box of the Publishing Wizard, **Publishing Options**, click **Load Settings**. The **Load Publish Settings** dialog box appears. Select the publish settings (.HDP) file and click **Load**.

Use a Custom Library Splash Screen

You can attach a custom splash screen to any HotDocs library. The splash screen is displayed each time the library is opened. Template set publishers use library splash screens to identify ownership of or contributors to the templates in a library.

To attach a splash screen to a library

- 1. Create a bitmap file (.BMP) of your splash screen, then save it with the same base name as your published library name. For example, the splash screen for the published library with the file name *PUBBusiness.hdl* would be *PUBBusiness.bmp*.
- Save the splash screen file to the same folder as the library file to which you want it associated. (If you are publishing an auto-install file, add the file at the **Additional Files** dialog box of the Publishing Wizard.)

When a user opens a library by clicking the **dHotDocs** button in the word processor toolbar, no splash screens are displayed.

Automatically Update Published Template Sets

For complete information on automatically updating your published template sets through HotDocs, see the HotDocs API section of the HotDocs Knowledge Base.

When you publish a template set, you can distribute a HotDocs update catalog (.HCAT) file along with your templates. HotDocs uses this file to automatically check for updates and notify users when updates are available for your template set. Catalog files contain a link to an RSS document ("feed") on your organization's Web server, which lists all of the available updates. When you need to update the template set, you add an item to the RSS document and users see the update the next time HotDocs checks the RSS feed.

To use this feature, you must create an update catalog (.HCAT) file that you distribute to each user and an update document (RSS feed) that you host on your Web server.

HotDocs Update Catalog Files

A HotDocs update catalog file is an XML file that contains information about the template set to be updated. The following table lists the main elements of a catalog file:

Element	Description
formatversion	A number that indicates the catalog file format. Currently, the only recognized value is 1 , but future versions of HotDocs may support additional values.
feedurl	A URL for the update document (RSS feed). For example, http://www.yourcompany.com/update.rss .
checkinterval	The number of days between each update check. If the interval is 0 , HotDocs will check for updates every time it starts.
publishdate	The date on which the template set was published. When HotDocs reads the RSS feed, it ignores any updates published prior to this date. The date must be in the following format: Sun, 03 Jun 1990 00:00:00 GMT .
title	The title of the template set. For example, Hobble Creek Real Estate Forms.
password	The password required to install HotDocs Auto-Install (.HDI) files for your template set. When HotDocs installs an update that requires a password, it uses the password found in this property. (If you set this property, you should use the same password each time you create an .HDI file to update your template set.)
enabled	Indicates whether HotDocs should check for updates for this template set (true) or not (false). Users can change this value at HotDocs Options (Tools > Options > Template Set Updates).

In addition to these elements, catalog files contain three lists of update items: **notinstalleditems**, **discardeditems**, and **installeditems**. HotDocs populates these lists and keeps them up to date automatically as it reads the RSS feed specified in the **feedurl** element. When you create an update catalog file for distribution with a template set, these lists are normally empty.

The following example shows the contents of a typical update catalog file without any updates.

Example HotDocs Update Catalog File

You can distribute the update catalog file in two ways. If you publish your template set as a HotDocs Auto-Install (.HDI) file, you can include the .HCAT file in the list of additional files. When HotDocs installs the template set, it automatically copies the .HCAT file to the Catalog Files folder specified in the user's **HotDocs Options** (**Tools > Options > File Locations > Catalog Files**). You can also manually copy the .HCAT file to the user's *Catalog Files* folder.

HotDocs must have read/write access to the .HCAT file. If HotDocs cannot access the folder or catalog file, the catalog file will be ignored.

HotDocs Update RSS Document

A HotDocs update RSS document is an XML file that conforms (with a few exceptions) to the RSS 2.0 specifications. It contains one channel with the following required elements:

Element	Description
title	The title of the template set. For example, Hobble Creek Real Estate Forms.
link	A URL for the template set. It may be a Web site that contains information about your company or template set. (HotDocs does not currently use this element, but it is included to conform to the RSS specifications.)
description	A description of the template set. (HotDocs does not currently use this element, but it is included to conform to the RSS specifications.)

The RSS channel also includes a list of individual update items, which contain the following elements:

Element	Description
title	The title of the update item. For example, if you update your template set monthly, the title may include the month in which the update is released. If the update fixes a problem with a particular template, the title may include the name of the template.
link	The URL for the update item. For file updates, this is the URL for a HotDocs Auto-Install (.HDI) file. For message updates, this is either the URL for a Web site you want users to visit, or it can be empty if you just want to display a simple message (see the description element below).
description	A description of the update item. If a message update does not include a link, HotDocs displays this description when the user applies the update.
guid	A string that uniquely identifies this update item. This must be different for each update item within your template set.
category	The type of item update. It can be one of the following two values:
	File Update: Installs an .HDI file.

Message: Displays a message or links to a Web page.**pubDate**The date on which the update item was published. When HotDocs reads the RSS
document, it ignores any items with a publication date earlier than the publication date in
the .HCAT file. The date must be in the following format: Sun, 03 Jun 1990 00:00:00
GMT.

You can also indicate if an item is required, which means that end users cannot discard (hide) the update. HotDocs does not force users to install the update, but it will appear in the list of updates every time HotDocs checks for updates until it has been installed. To make an update item required, add **required="true"** to the item tag (e.g., **<item required="true">**).

The following example shows an RSS document with three update items.

Example RSS Document

Publishing HotDocs Server Templates

HotDocs Server Overview

HotDocs Server is the server-based version of HotDocs that allows HotDocs interviews (data-gathering sessions) to be displayed by a standard Web browser and documents to be generated (assembled) on the web server, without requiring any special software to be installed on an end-user's computer. Generally speaking, when a user goes to your Web site and initiates assembly of a document, a request is made to the HotDocs Server engine, which is running on the server. HotDocs Server then sends an interview back to the user's browser, where it appears as part of a Web page. Once the required information has been entered, the user clicks a button that posts the answers (in XML format) back to HotDocs Server, where they can be merged into assembled documents..

The following diagram shows the assembly process:

Publishing Templates



The benefit of creating a Web application using HotDocs Server is that end users do not need HotDocs installed on their computers, since the interview is presented in a Web browser and document assembly happens on the server. For example, your human resources department may integrate its forms into its intranet site using HotDocs Server. Employees would then go to the intranet site, select which form to fill out, provide the required information, and then submit the form to the appropriate human resources representative—all from within a Web browser. Depending on project requirements, employees may also print copies of assembled documents, save copies of assembled documents to their hard drives, or (if the intranet site facilitates it) simply store assembled documents or answer files directly in the intranet app.

HotDocs Server may also be used to generate a document without an interactive interview. Likewise, HotDocs browser interviews may be used to gather information for storage or later use without immediate assembly of any documents at all.

HotDocs Server vs. Desktop HotDocs

The HotDocs family of products includes both desktop- and server-based products. The primary desktopbased product, HotDocs Developer, is used for creating HotDocs templates. HotDocs Server, on the other hand, is used only for running templates; it cannot be used to create them. Another difference between Server and desktop HotDocs is the user interface: desktop editions of HotDocs include word processor and other integrations for creating templates, and the "HotDocs Library" for managing templates, answer files, and more. The user interface presented directly by HotDocs Server, at least from an end user's perspective, is limited to displaying interviews in the Web browser. All other interaction with the user is accomplished by a custom host application, which you must build and maintain. The advantage with this

approach is that you can expose your users to only as much user interface as they need to perform their document assembly tasks.

Despite their differences, desktop HotDocs still plays an important role in relation to HotDocs Server: it is the tool used to create (and help test) all content (templates) that will eventually be deployed in HotDocs Server.

If you plan to primarily develop templates for HotDocs Server then you can use the HotDocs Options to enable every newly created template for HotDocs Server by default.

Template Features Not Allowed in HotDocs Server or Cloud Services Interviews

There are some template building features that are specific to HotDocs Server and HotDocs Cloud Services. Other features work differently in desktop interviews than in Server and Cloud Services. The following template features are NOT compatible with browser-based deployments. Using any one of these features in a template prevents you from uploading that template to HotDocs Server or HotDocs Cloud Services. (An error message appears when you try to test the template in a browser or enable the template for use with Cloud Services or HotDocs Server.)

Feature	Description
Clauses and clause libraries	Although clauses and clause libraries are not supported in HotDocs Server, you can use inserted templates to achieve similar results. For example, you can use the answer for a True/False variable to determine when to insert a template into the assembled document.
INSERT instructions used in Interview Templates	HotDocs Server cannot process INSERT instructions in Interview Templates.
LANGUAGE instructions used in expressions	LANGUAGE instructions are not allowed when used as part of an expression. However, you can merge the instruction directly in the template.
Personal Information variables	Personal Information variables require access to the Current User key of the Windows registry. Since HotDocs Server cannot access the user's registry, this feature is not allowed.
ANSWER FILE NAME variables	The ANSWER FILE NAME variable is used to merge the name of the answer file used in the assembled document. Because of the differences in the way server-based versions of HotDocs work with answer files, this is not allowed.
Database components	Database components created using HotDocs Developer cannot be used in HotDocs Server templates deployed to HotDocs Server or Cloud Services.

There are workarounds available for some of the disallowed features below.

PLAY "MACRO" instructions	This instruction plays a word processor macro after the document is assembled and either sent to the word processor, printed, or saved. Playing a macro requires access to a word processor, and since HotDocs Server assembles documents outside of the word processor, it cannot play macros.
Application link dialog elements and EXECUTE instructions	These instructions are used to start other applications on the user's computer. However, they are not allowed in HotDocs Server templates.
Automatic Paragraph Numbering	In JavaScript and Silverlight interviews, you cannot use PN numbering (e.g., «PN1». What is your name?) to automatically number the prompts. You can still use PN numbering in the text of the template itself.
Bookmarks	Bookmarks that have been inserted into a document by Word or WordPerfect cannot be used with HotDocs server.

Features that Work Differently in Desktop Interviews than in Server and Cloud Services Interviews

The following template features are allowed in templates deployed on HotDocs Server or HotDocs Cloud Services; however, they work differently in desktop interviews than in browser-based interviews. In some cases, the end result may be the same, but the difference between the desktop and browser interview user interfaces require you to change how you implement some features. There are also some features that are ignored when they are used in templates enabled for HotDocs Server (browser-based) deployment. Unlike those features not supported in browser-based interviews, the features listed below do not cause any errors.

Feature	Description
Answer sources	In desktop HotDocs, you can specify an answer source (a list of answers from which a user can select one) for a dialog. Users open the list when answering a question in a dialog and pick an existing answer, instead of entering one manually. However, when presenting interviews in a web browser, only "CURRENT ANSWER FILE" answer sources are supported; others are ignored (the Select button does not appear on the dialog).
Automatically selecting a Multiple Choice option when the variable is unanswered	In HotDocs Developer, at the Multiple Choice Variable Editor , you can click the Options tab and specify which option should be automatically selected if the variable is unanswered when displayed. Although these "default" options are ignored in browser-based

	interviews, you can still achieve the same results by using the DEFAULT instruction in your template.
Text patterns	Text patterns are supported in both desktop and browser-based interviews, but the way answers are formatted as they are entered during the interview is slightly different in each version.
Example formats	HotDocs fields embedded in variable prompts, dialog element text, resource text, etc., within browser interviews, do not yet support the full range of format examples supported by desktop HotDocs. If a format example is encountered that is not recognized, the variable's answer will be merged using a reasonable default format.
Rows to display	If you alter the amount of spreadsheet Rows to display in the interview, at the Dialog Editor, the change will not be carried over to the browser-based interview after publishing.
Web link dialog elements	You can use Web link dialog element to add a button to a dialog that links to a database. However, this works only in browser-based interviews. Web links that launch such databases do not appear in desktop interviews.
Image dialog elements	The following restrictions apply to images displayed in dialogs:
	 .JPG and .PNG images are supported in all browser-based interviews. In addition, JavaScript interviews allow .GIF images. Browser interviews do not support .BMP images.
	 Some rare PNG files (gray scale and 64-bit true color) may not be supported in Silverlight interviews.
Resources for variables and dialogs	Browser-based interviews only support Plain Text and URL resources. All other resource types (HTML Help , Windows Help , Folio Infobase , Custom Program) are ignored. (No resource is displayed for the dialog or variable.) In addition, browser-based interviews display all URL resources as hyperlinks that open up separate windows; the actual Web page does not appear in the resource pane.
	Interviews intended to be displayed on an Ipad should note that when the resource pane is activated the on-screen keyboard will be closed.
End of Interview dialog	HotDocs browser-based interviews do not include an <i>End of</i> <i>Interview</i> dialog. Selecting Hide End of Interview dialog at the Component File Properties dialog box (Assembly tab) has no effect on a browser-based interview.

Question and Answer summaries	Selecting Use variable names in summaries at the Component File Properties dialog box (Assembly tab) has no effect on a browser- based interview. (Browser-based interviews do not have an option for creating Question summaries, and Answer summaries always use variable prompts.)
"TypeHere" bookmarks	Since HotDocs Server and Cloud Services do not send assembled documents to the word processor, selecting Move to the "TypeHere" bookmark at the Component File Properties dialog box (Assembly tab) is not applicable.
Prevent users from using an answer file to assemble a document	Since answer file selection and processing is handled by your host application, the Do not use answer files option at the Component File Properties dialog box (Assembly tab) does not affect browser-based interviews.
Max WHILE iterations	HotDocs Server and Cloud Services do not honor the Maximum WHILE iterations value specified at the Component File Properties dialog box (Assembly tab). (You should be careful not to create an infinite WHILE loop in your templates.)
Max processing stack depth	HotDocs Server and Cloud Services do not honor the Maximum processing stack depth value specified at the Component File Properties dialog box (Assembly tab). (You should be careful to avoid infinitely recursing (processing) a computation.)
Accelerator keys in variable prompts	In desktop interviews, you can make a character in a variable prompt an accelerator key by typing an ampersand (&) character immediately before the letter you want as an accelerator. Browser- based interviews correctly remove the ampersand when the prompt is displayed in the interview, but does not make the following character an accelerator key.
SUM(COMPUTATION_VAR) expression	Using the SUM(COMPUTATION_VAR) expression to display information during the interview is not supported. You can use this expression as part of the text in the assembled document, but not in a dialog script or dialog element.
Dot Codes	The following dot codes are not supported in Silverlight interviews:
	• «.tc»: Tab character dot codes are ignored.
	• «.oh»: An optional hyphen is treated like a normal hyphen.
	 «.pb» and «.pm»: Page break and paragraph mark dot codes are both treated like line breaks.
	 «.w "URL"»/«.we»: New line characters, strike-through, and underline formatting is ignored within the text of a Web link.

Automatic Prompt Alignment	In Silverlight interviews, automatic prompt alignment works differently than it does in JavaScript or desktop interviews.
ASK NONE script in Interview computations	As well as not being asked in the interview, variables that have been included in the ASK NONE script in desktop HotDocs will be marked as having already been asked so any subsequent attempts to ASK them in the interview script will not work. This is not the case in browser interviews where ASK NONE will allow these variables to be asked again later in the interview script.
SPAN Instructions	When HotDocs assembles a template which contains a SPAN instruction, it checks the answer file to see if it contains an answer for that SPAN component. If there is an answer, the text within the SPAN instruction is replaced by the text in the answer file. This works whether you are assembling a document on the desktop or from a browser-based interview. The difference, however, is that desktop interviews provide an interface at the Document Preview tab where you can edit the text within a SPAN instruction. In browser-based interviews, there is no such user interface for editing SPAN text, which puts the usefulness of SPAN instructions into question in this case.

Enable Templates for Use With HotDocs Server

If you plan to primarily develop templates for HotDocs Server then you can use the HotDocs Options to enable every newly created template for HotDocs Server by default.

To enable templates for use with HotDocs Server

- 1. Open the template's component file for editing and click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 2. Click the HotDocs Server tab and select Enable template for use with HotDocs Server.
- 3. Select any of the following options:

Option	Description
Interview outline initially showing	This option causes the interview outline (list of dialogs in the interview) to appear when the interview is first displayed.
Allow user to hide/show interview outline	This option lets users show and hide the interview outline as needed.

Resource pane initially showing	This option causes the resource pane, which shows helpful information about questions in the interview, to appear when the interview is first displayed.
Allow user to hide/show resource pane	This option lets users show and hide the resource pane as needed.
Instant Update initially ON	This option causes HotDocs Server to automatically update the entire interview as users enter or change answers. If this option is turned off, HotDocs will only update the interview as needed. (Some complex interviews may require Instant Update to be turned off or response time may be slow.) The Instant Update option applies only to JavaScript interviews. In Silverlight, interviews are always instantly updated.
Allow user to turn Instant Update on/off	This option lets users control how frequently HotDocs updates the interview.
Single-page interview initially ON	This option causes the interview to appear as a single page when the interview is first displayed.
Allow user to turn single- page interview on/off	This option lets users switch between normal and single-page interviews.

When a template is enabled for use with HotDocs Server, HotDocs warns you if you try to use a feature that is not allowed in a HotDocs Server template. In addition, HotDocs Server-enabled templates can be tested in a Web browser to ensure that the interview appears as intended.

To view a JavaScript or Silverlight browser interview, use the **Template > Test in Browser** command at the library window. You can also view a JavaScript-based interview by pressing the **Shift** key as you click the **Assemble** button. To view the interview using Silverlight, press the **Ctrl** key as you click the **Assemble** button. (This also works while editing a template; you can press either the **Shift** key or the **Ctrl** key as you click the ***Test Assemble** button in your word processor or HotDocs Automator.)

It is recommended that template names are kept to less than 52 characters and don't use non-ASCII characters. This will reduce the possibility of the template name causing errors if it is enabled for use with HotDocs Server.

'/', '\', and '?' cannot be included in the template name when you are creating a template package for use in a HotDocs Cloud Services direct interview.

Viewing an Interview in a Web Browser

When you test assemble a template you plan to use with HotDocs Server, you can display the interview in a Web browser. This allows you to test its functionality and view its overall appearance.

Templates containing instructions to assemble another template (for example, «ASSEMBLE *SomeTemplateName*.rtf'») do not work when you use Test In Browser; however they do work when actually run in an independent browser instance.

To view an interview in a Web browser

- 1. At the HotDocs library window, select a template and ensure it is enabled for use with HotDocs Server. (See Enable Templates for Use with HotDocs Server.)
- 2. Select one of the following options from the **Template > Test in Browser** menu:
 - JavaScript: Displays a traditional JavaScript-based interview in a browser window.

Starting with HotDocs 11.1 JavaScript interviews have a slightly altered layout. You can minimize the Interview Outline to the left of the window and the Resource Pane to the right of the window. You can view either panel by clicking on the side bar and causing it to expand. You can click the arrow button on the top right of the panels to keep them open while you enter answers.

• **Silverlight**: Displays a Silverlight-based interview in a browser window. Unlike the traditional JavaScript-based interviews where the interview logic is written in an interpreted language (JavaScript), Silverlight-based interviews are compiled to machine code, which makes them much quicker at processing and updating answers (this is especially noticeable in long, complex interviews).

The Silverlight option is only enabled if you have the Microsoft Silverlight 3 Runtime, SDK, and Microsoft .NET Framework 3.5 installed on your computer. See System Requirements for details.

- 3. HotDocs displays the **Answer File** dialog box.
- 4. Select an answer file and click **OK**. A browser window appears and displays the interview.
- 5. Type any answers and click Finish. HotDocs closes the browser window and displays the assembled document in your default word processor or HotDocs Filler (for form documents).

You can also view a JavaScript-based interview by pressing the **Shift** key as you click the **Assemble** button. To view the interview using Silverlight, press the **Ctrl** key as you click the **Assemble** button. (This also works while editing a template; you can press either the **Shift** key or the **Ctrl** key as you click the **Test Assemble** button in your word processor or HotDocs Automator.)

'/', '\', and '?' cannot be included in the template name when you are creating a template package for use in a HotDocs Cloud Services direct interview.

Publish Templates For Use With HotDocs Server

To publish and upload templates for use with HotDocs Server

- At the HotDocs library, select the template or templates you want to publish and click the Publishing Wizard button. The Publishing Wizard dialog box appears, displaying the Publishing Options information.
- 2. Select **Template files for use with HotDocs Server** from the **Publish as** group
- 3. Click the **Browse** button next to the **Local folder for published files** field to specify the folder where you want HotDocs to save the published files. (Once these files are published to this temporary folder, you should move them to a more permanent location. If you publish additional templates or other files that use the same file names, HotDocs will overwrite the previously published files.)
- 4. Choose how you want to upload the templates to the server:
 - Automatically: Click the Edit button next to the Web destination for published files field to display the Web Destination dialog box where you can specify the URL and description ("friendly name") of the Publishing Form Page on the server to which you want to upload the files.
 - Manually: Leave the Web destination for published files field empty.
- 5. Click Next. The Additional Files dialog box appears.
- 6. Select any additional files you want to include with your published templates, and then click **Finish**. HotDocs finishes publishing your templates.

If you chose to automatically upload the templates, the Publishing Form Page appears where you can enter additional information about the templates you are uploading, and then upload that information and the templates to the server using the Publishing Destination Page.

If you chose to manually upload the templates, move the files from the Local folder for published files to the server. You can use an FTP client or other application, such as Microsoft FrontPage.

When a template is published for use with HotDocs Server, HotDocs creates an additional file required by HotDocs Server: the template manifest file (.manifest.xml). This file lists all of the variables used in a given template, as well as the dependencies the template has on other files.

If you upload the template automatically using the **Publishing Wizard**, these files are also uploaded to the server automatically. If you manually upload your template, you must remember to copy these files to the server.

'/', '\', and '?' cannot be included in the template name when you are creating a template package for use in a HotDocs Cloud Services direct interview.

What is a Publishing Form Page?

The Publishing Form Page contains an HTML form. With this form, you can provide fields or other options for the template developer to add more information about the files being uploaded, such as template titles and descriptions. You must follow certain criteria when creating the HTML form:

- The form must use the POST method and its encoding must be multipart/form-data. (HotDocs 2007 or earlier required that the form be named *HotDocsPubForm*, but this is no longer required.)
- Previously, the form ACTION was required to include an absolute URL path to the location on the server where you want to send the files, but starting with HotDocs 2008 Professional Edition, this is no longer a requirement.
- Any Select/Option fields (or drop-down/list/combo boxes) must have a VALUE attribute assigned. For example: <option value="choice1">choice1</option>.

Once the template developer provides the necessary information in this form, he or she submits the form and HotDocs adds the information to the published files. All the information is then uploaded (using HTTP POST) to the server. The following is sample code for the Publishing Form Page.

Example:

<form method="POST"

```
name="HotDocsPubForm"
action="http://www.company.com/hotdocsuploadproc.asp"
enctype="multipart/form-data">
Template Name: <input type="text" name="textinput" size="65">
<br>
<br>
Template Desc: <textarea rows="2" name="textareaDesc" cols="55">
Template Desc: <textarea rows="2" name="textareaDesc" cols="55">
Template Desc:: <textarea rows="2" name="textareaDesc" cols="55">
<br/>
</textarea><br>
<select size="1" name="dropdownbox">
<option value="choice1">cohice1</option>
<option value="choice2">cohice2</option>
</select>
</select>
</option value="choice2">submit"></option value="choice2"></option>
</select>
</option value="Submit"></option>
</option value="Submit"></option>
</option type="Submit"></option>
</option type="Submit"></option</p>
```

</form>

What is a Publishing Destination Page?

After the uploading process is complete, the Publishing Destination Page appears. This page accepts the HTTP POST from the Publishing Form Page and processes it by completing the actions specified by your project.

Once the information has been received on the server, the files from the upload must be decoded and handled. Several components are available to help with this process, including the Dundas Upload control, distributed by Dundas Software. The following is sample code for the Publishing Destination Page using the Dundas Upload control.

Example:

<%

dim dund
<pre>Set dund = Server.CreateObject("Dundas.Upload.2")</pre>
dund.SaveToMemory
response.Write "Data from form <hr/> "
dim objUploadedForm
For Each objUploadedForm in dund.Form
Response.Write objUploadedForm.tagname & "="
Response.Write objUploadedForm.Value & " "
Next
response.Write " Files from HotDocs <hr/> "
dim objUploadedFile
dim filename
dim filen
dund.ImpersonateUser "username", "password"

For Each objFile in dund.Files
 filen = dund.GetFileName (objFile.OriginalPath)
 filename = "c:\inetpub\companyname\temp\" &
 objFile.SaveAs filename
 Response.Write "<a ref='http://www.company.com/temp/"
 Response.Write filen &"'>"
 Response.Write filen &"'>"
 Response.Write objFile.TagName
 Response.Write "=" & objFile.OriginalPath
 Response.Write " : Size = " & objFile.Size
 Response.Write "

set dund = nothing

%>

Publishing Templates for HotDocs Cloud Services

Overview: Upload HotDocs Files to a HotDocs Cloud Services Host Application

Overview: Upload HotDocs Files to a HotDocs Cloud Services host application

HotDocs Cloud Services is an online "software-as-a-service" (SaaS) implementation of HotDocs that enables you to provide HotDocs interviews to end users through a web browser (without requiring you to install your own HotDocs Server). To deliver interviews, you fist upload your templates to a web application hosting HotDocs Cloud Services. Such a web application is called a HotDocs host application. HotDocs provides an "off-the-shelf" host application in HotDocs Document Services; however, anyone can build their own host application to leverage Cloud Services for delivering interviews to end users.

In order to upload HotDocs templates, URLs, or any other needed files for use with a Cloud Services host application, HotDocs Developer 10.2 and above includes an upload plugin to upload templates from a desktop template library to a host application.

Enabling a Template for Uploading to Cloud Services

If you want to enable a template for uploading to Cloud Services, you have two options:

- You can use the Component Manager to access the component file's properties
- You can use the Test in Browser feature.

Enabling a Template for Upload Using the Component Manager

You can enable a template from the HotDocs Server tab of the Component Files Property dialog.

To enable a template for uploading using the component manager:

- Open your template library in HotDocs Developer, select the template you want to upload to Cloud Services; then in the toolbar at the top of HotDocs Developer, click the **Component Manger** button.
- 2. On the left of the Component Manager, click the **Component Files Properties** button.
- 3. Click the HotDocs Server tab and select Enable template for use with HotDocs Server.

For a description of the various options made available when you enable the template, see the At a Glance document.

Enabling a Template for Upload Using Test in Browser

You can enable a template using the Test in Browser method by right-clicking a template.

To enable a template for uploading using test in browser:

- 1. In the HotDocs library, right-click a template; then click **Test in Browser** before clicking either **JavaScript** or **Silverlight**.
- 2. On the message that appears, click **Enable**; then on the Answer File dialog, click **Cancel**.

If the template has already been enabled, the Enable window does not appear; click **Cancel**.

Adding an Expiration Date to a Template for Use with Cloud Services

This option is only available for Cloud Services when uploading from HotDocs Developer 11 and above.

If you want to add an expiration date to a template that you intend to upload for use with Cloud Services, you need to use the Publishing Wizard.

To add an expiration date to a template for uploading to cloud services:

- 1. Open your template library in HotDocs Developer and select the template or templates you want to publish; then in the toolbar at the top of HotDocs Developer, click the **Publishing Wizard** button.
- 2. From the Publish as pane, select **Standard template files**.
- 3. Next to the Local folder for published files box, click **Browse** to specify a temporary folder where you want HotDocs to save the published files.

This cannot be the folder where the template is located, or the Publishing Wizard throws an error. You should further note that copying or republishing files to the same location while using the same file names overwrites the previously published files (which can cause problems, particularly if you overwrite your source template with a new one with an expiration date attached). For best practice, follow the sub steps below:

- a. Close the Publishing Wizard, then on the Properties tab of HotDocs Developer click the **URL link** that appears below File Path.
- b. In the Windows Explorer instance that appears, right-click in the file area and select New > Folder.
- c. Rename the new folder *With Expiration Date*; then double-click the **With Expiration Date** folder.
- d. At the top of Windows Explorer, copy the file path for the With Expiration Date folder.
- e. In HotDocs Developer select the template or templates you want to publish; then in the toolbar at the top of HotDocs Developer click the **Publishing Wizard** button.
- f. From the Publish as pane, select **Standard template files**.
- g. Paste the path into the Local folder for published files text box; then click Next.
- 4. Under File Security, select: Set expiration date for component files; then click Next.
- 5. To the right of the Expiration date field, click the **calendar icon**, choose an expiration date; then click **OK**.
- 6. If you want to include a warning or extension period, use the arrows to choose a value. If you want to provide an expiration message, type one in the **Expiration message** field; then click **Next**.
- 7. If you want to include any additional files in your template, click **Add Files**, locate the files (by browsing), and click **Select** before clicking **Finish**; otherwise, click **Finish** directly.
- 8. When publication is finished, click **OK**.
- 9. Open the temporary folder into which you published the template above; then at the top of Windows Explorer, copy the file path to that folder.
- 10. Next, HotDocs Developer, right-click the top element in your library; then click Add Folder.
- 11. In the **Title** text box, type *With Expiration Date*; then add any description you want and click **OK**.

- 12. Ensure the With Expiration Date folder you just created is open; then right-click the folder and select **Add Item**.
- 13. Paste the path you previously copied from Windows Explorer into the File name text box; then click the **Browse** button before double-clicking the **With Expiration Date** folder.
- 14. Click the **Files of type** drop-down arrow and select **All Files**; then double-click the template you wish to upload before clicking **OK**.
- 15. Next, enable the template for uploading to a Document Services (or a host application site) by doing the following:
 - a. In the With Expiration Date folder of your HotDocs library, right-click the template you just added; then click **Test in Browser** before clicking either **JavaScript** or **Silverlight**.
 - b. On the message that appears, click **Enable**; then on the Answer File dialog, click **Cancel**.

If this template has already been enabled, the Enable window does not appear; click **Cancel**.

16. Return to the library, right-click the template you just enabled; then click **Upload** before clicking the name of the upload site to which you want to upload the template package.

If the an error message is thrown stating "The Component File Has No Computation Designated as Its Interview," see the Error Handling section below.

Error Handling: "The Component File Has No Computation Designated as Its Interview" Error Message

- 1. On the error message, click **OK**.
- 2. Select the template in the With Expiration Date folder, then in the tool bar at the top of Developer, click **Component Manager**.
- 3. On the left of the Component Manager, click the **Component Files Properties** button.
- 4. Click the Interview tab; then select Use custom interview.
- 5. Click the **Interview component drop-down arrow**; then choose the computation component you want to use, or create a new computation component by clicking the **Edit Component** button.
- 6. Click **OK**; then close the **Component Manager** and retry uploading your template.

Publishing and Uploading Templates and Other Files for Use with Cloud Services

Before you can upload a template package you need to have an upload page in your host application.
Publishing a template is a seamless part of the process for uploading a template for use with Cloud Services. With the release of HotDocs 11, publishing a template means that a template package is both created and uploaded in a single operation.

You can publish and upload a template either to Document Services (an off-the-shelf host application) or to a third party host application web server. All you need is the URL of the destination and the credentials to access the server.

To publish and upload a template for use with cloud services:

1. Ensure you have enabled your template for uploading.

You do not need to enable non-template files (such as a PDF) or URLs prior to publishing them.

In the left hand (library) column of HotDocs Developer, right-click a template; then click Upload > Edit Upload Sites. If you have not yet added the site to which you want to upload your template, perform the sub steps that follow, otherwise proceed to the next primary step.

If you are using HotDocs Document Services you will need to download the configuration file from your site, see the Document Services help file for more information.

- a. Click **Add**; then type the **Site Name** of the host application site (whether Document Services, or a third party) and the corresponding **URL** location.
- b. Click **OK**; then click **Close**.
- 3. Right-click the template (or other file) you want to upload; then click **Upload** before clicking the **name of the upload site** to which you want to upload the template package.
- 4. Follow the instructions on the upload page that appears.

Using the Upload Plug-in to Save Template Packages Locally

If you decide you want to use a different method (say FTP) than an upload page to move your templates to a host application server or the Document Services server, you can still create a template package using the upload plug-in. You have two options for doing so, each with different advantages. You can:

- 1. Save a template package in the same directory as the template. This has the following advantages:
 - Saving locally is an option on the Upload menu.
 - The template package has the same base file name as the template, but has a .hdpkg (HotDocs Package) file extension. This, and the fact that package save location is in the same folder as where the template is stored make it clear which template the template package is associated with.

2. Save your template packages to a "local upload site." The has the advantage of producing the same output as when a package is uploaded to a host application web server.

Saving a Template Package in the Same Directory as the Template

When you want to save a template package locally, the simplest solution is to use the Save Locally option on the right-click Upload menu to save the template package in the same folder as the template.

To save a template package in the same directory as the template:

- In the left hand (library) column of HotDocs Developer, right-click a template; then click Upload > Save Locally.
- 2. On the Save Locally page, you can click the down arrow to see the file names of the files in the template package just created. When you are finished, click **Save Locally**.
- 3. To view the template package at its saved location, on the Properties tab of HotDocs Developer, click the link under **File path** and locate the template package. The template package has the same base file name as the template, but has a .hdpkg (HotDocs Package) file extension.

Saving a Template Package to a "Local" Upload Site

When you want to save a template package locally, but you want the same output as when you create a package to upload to a host application web server, you can save the package to a local upload site.

To save your template package to a local upload site:

- In the left hand (library) column of HotDocs Developer, right-click a template; then click Upload > Edit Upload Sites.
- 2. Click **Add**; then type a **Site Name** for your local storage area.
- 3. In the **URL** field, type the full path of an existing local directory (or shared network directory) where you want to save your template packages.
- 4. Click **OK;** then click **Close**.
- 5. Enable a template for uploading to your local upload site by doing the following:
 - a. Right-click a template; then click **Test in Browser** before clicking either **JavaScript** or **Silverlight**.
 - b. On the message that appears, click **Enable**; then on the Answer File dialog, click **Cancel**.

If this template has already been enabled, the Enable window does not appear; click **Cancel**.

- 6. Right-click the template you just enabled; then click **Upload** before clicking the **name of your local upload site**.
- 7. On the HotDocs Upload page, you can click the down arrow to see the file names of the files in the template package just created. When you are finished, click **Upload**; then click **OK**.

8. To view the template package at its saved location, open Windows Explorer and navigate to your local upload site. The template package has a .pkg file extension, and the manifest has a .xml extension. Both have the same GUID as their base file name.

Uploading Published Files

Introduction: Upload HotDocs Files to a Web Server

You can upload HotDocs template files to a Web server. These include standard template files, autoassemble files, auto-install files, and HotDocs Server files. You can do this by creating a custom Web publishing method, which allows you to use an HTML form to upload your template files. This method also allows you to specify and gather metadata about the files, such as titles and descriptions, before uploading the files to the server. This information can be integrated with a database or used with other custom programs or Web designs.

Cloud Services Overview

HotDocs Cloud Services is the cloud-based version of HotDocs that allows HotDocs interviews (datagathering sessions) to be displayed by a standard Web browser and documents to be generated (assembled) in the cloud, without requiring any special software to be installed on an end-user's computer. Generally speaking, when a user goes to your Web site and initiates assembly of a document, a request is made to HotDocs Cloud Services, which is running in the cloud. Cloud Services then sends an interview back to the user's browser, where it appears as part of a Web page. Once the required information has been entered, the user clicks a button that posts the answers (in XML format) back to HotDocs Cloud Services, where they can be merged into assembled documents.

The following diagram shows the assembly process:

Publishing Templates



The benefit of integrating a Web application with HotDocs Cloud Services is that end users do not need HotDocs installed on their computers, since the interview is presented in a Web browser and document assembly happens on the server. For example, your human resources department may integrate its forms into its intranet site using HotDocs Cloud Services. Employees would then go to the intranet site, select which form to fill out, provide the required information, and then submit the form to the appropriate human resources representative—all from within a Web browser. Depending on project requirements, employees may also print copies of assembled documents, save copies of assembled documents to their hard drives, or (if the intranet site facilitates it) simply store assembled documents or answer files directly in the intranet app.

HotDocs Cloud Services may also be used to generate a document without an interactive interview. Likewise, HotDocs browser interviews may be used to gather information for storage or later use without immediate assembly of any documents at all.

HotDocs Cloud Services vs. Desktop HotDocs

The HotDocs family of products includes both desktop- and server-based products. The primary desktopbased product, HotDocs Developer, is used for creating HotDocs templates. Cloud Services, on the other hand, is used only for running templates; it cannot be used to create them. Another difference between Cloud Services and desktop HotDocs is the user interface: desktop editions of HotDocs include word processor and other integrations for creating templates, and the "HotDocs Library" for managing templates, answer files, and more. The user interface presented by Cloud Services, at least from an end user's perspective, is limited to displaying interviews in the Web browser. All other interaction with the

user is accomplished by a custom host application, which you must build and maintain. The advantage with this approach is that you can expose your users to only as much user interface as they need to perform their document assembly tasks.

Despite their differences, desktop HotDocs still plays an important role in relation to HotDocs Cloud Services: it is the tool used to create (and help test) all content (templates) that will eventually be deployed in Cloud Services.

Create Custom HTML Documents for Template Uploading

Once you publish your templates or template sets, you can upload the files to a Web server. This may be useful if you are developing templates for HotDocs Server, or if you want to make your templates available on an Internet or intranet site.

Since Web servers are configured to handle data exchange differently, you must customize the uploading process to meet your own needs. This requires you (or your Web server administrator) to create a custom HTML form, called the *Publishing Form Page*. This form can gather information about the user, the template set, or any other information required by your project.

Once this form has been created, the template developer can specify its URL as a destination for the upload. (This is done at the **Publishing Options** dialog box of the **Publishing Wizard**.) When the publishing process finishes, the form appears and template developers can provide the information required to complete the uploading process. Then, once they have provided the information and submitted the form, HotDocs posts that information, along with the template or template set, to the server.

As the information and the files are uploaded to the server, the server must know how to manipulate the data being sent from HotDocs. This requires you to create a second HTML document, called the *Publishing Destination Page*. This page contains scripting that can accept the HTTP POST from the *Publishing Form Page*. It should then process the data and files contained in the upload by entering metadata about the templates in a database, and then by copying the published files to the correct server locations, or by performing any other number of services your project requires.

During the publishing process, HotDocs logs a report about the uploading process. This report may be useful if you experience problems uploading your templates. To view this report, open *0hdUploadDebug.htm*, located in the *Local Settings\Temp* folder (for example, *C:\Documents and Settings\<username>\Local Settings\Temp\0hdUploadDebug.htm*).

Publishing Form Page

The Publishing Form Page contains an HTML form. With this form, you can provide fields or other options for the template developer to add more information about the files being uploaded, such as template titles and descriptions.

You must follow certain criteria when creating the HTML form. First, you must name the form **HotDocsPubForm**. Second, the form ACTION must include an absolute URL path to the location on the server where you want to send the files. Last, any Select/Option fields (or drop-down/list/combo boxes) must have a VALUE attribute assigned, for example:

<option value="Option Text">Select the option</option>

Once the template developer provides the necessary information in this form, he or she submits the form and HotDocs adds the information to the published files. All the information is then uploaded (using HTTP POST) to the server.

Click here for an example of a Publishing Form Page.

Publishing Destination Page

After the uploading process is complete, the Publishing Destination page appears. This page accepts the HTTP POST from the Publishing Form page and then processes it by completing the actions specified by your project.

Once the information has been received on the server, the files from the upload must be decoded and handled. Several components are available to help with this process, including the Dundas Upload control, distributed by Dundas Software.

Click here for an example of Publishing Destination Page. (This page uses the Dundas Upload control.)

At a Glance: The Web Destination dialog box

Web Destination	?	x
URL:		_
	A	Q
Eriendly name:	0	_
		_
	OK	

Clicking on the **Web Destination** dialog box.

You can type the URL of the website you would like to upload your published files to in the first field \underline{A} or click the \underline{a} **Browse** button to navigate to the site.

Then in the second field **B** you can type a **Friendly Name** for the URL that will help you to identify it.

To learn more about uploading templates to the web click the link below:

• Upload Templates to a Web Server

Upload Templates to a Web Server

Once you finish publishing templates and template sets, you can upload the published files to a Web server. This might be useful if you are publishing templates to be used with HotDocs Server, or if you want your users to access files from an Internet or intranet site.

To upload template files to a Web server

- 1. Publish your templates or your template set. (See Publish a Template as a Standard File, Publish Templates as Auto-Assemble Files (HDA), or Publish Templates as Auto-Install Files (HDI).)
- 2. At the **Publishing Options** dialog box, click the **VED** Edit button. The **Web Destination** dialog box appears.
- 3. Type an absolute URL in the **URL** field, or click the **Q Browse** button to locate the URL.
- 4. Type a description of the Web site in the Friendly name field, and click OK.
- 5. HotDocs displays the **Publishing Options** dialog box again, where you can complete the publishing process.

When HotDocs finishes publishing your templates, it displays your customized Web page where you can enter additional information about the templates you are uploading. Provide the information required by your organization and submit it. That information, along with your templates, are uploaded to the server.

In order for template uploading to work, you must create two HTML pages to handle the data you are uploading. (See Create Custom HTML Documents For Template Uploading.)

Uploading Published Files to a Server

Introduction: Upload HotDocs Answer Files

You can specify options that let users upload the answers they use to assemble documents. You must designate a destination URL and provide a resource at that URL, such as a CGI script or HotDocs Server server, to process the answers.

HotDocs answers are not uploaded in a HotDocs answer file, they are uploaded as a single block of data using the HTTP POST method. The server receiving the data sees it as a POST submission from an HTML form.

A CGI script receiving the data should do three things:

- Get the size of the incoming data block from the CONTENT_LENGTH environment variable.
- Read that number of characters from standard input.
- Parse the data to extract the HotDocs answers.

This section does not provide the details of writing a CGI script or other program to process the answers. It does, however, tell you how to enable answer uploading for a template. It also provides the information you need to parse the uploaded data and extract the HotDocs answers.

Specify Options to Upload Answers

You can allow users to upload answers they use with certain templates to a Web server. To do this, you must specify uploading options at each template's component file.

If you plan on uploading answers, make sure your templates don't contain variables with names longer than 30 characters. HotDocs uploads the variable name along with the answer, and it can only upload 30 characters for the variable name.

To enable users to upload answers for a template

- 1. Edit the template's component file using Component Manager. (See Open and Close Component Manager.) The **Component Manager** window appears.
- 2. Click the **TProperties** button. The **Component File Properties** dialog box appears.
- 3. Click the Answer Upload tab.
- 4. Type the URL of the Web page to which you want the answers uploaded in the **Upload URL** field.
- 5. Type a description of the page to which the URL points in the **Friendly name** field.
- 6. Optionally, select **Upload answers without asking** to force the answers to always be uploaded, according to the instructions you have specified.

When users assemble documents that have answer uploading enabled, at the end of document assembly, HotDocs will either prompt them to upload their answers, or it will upload the answers automatically (depending on whether you select **Upload answers without asking**). If users choose to upload their answers later, they can select **Upload Answers** from the **Tools** menu of the HotDocs library.

Handle Data in Uploaded Answers

This topic covers information on writing the CGI script or other program to process the answers. It contains the information needed to parse the data to extract the HotDocs answers, which are uploaded as a single block of data using the HTTP POST method. The server receiving the data sees it as a POST submission from an HTML form.

Format of Answers

The answers are uploaded in a single string of data, with each answer separated from the next by an ampersand, like this:

Answer1&Answer2&Answer3

The answers are not necessarily in the order in which they were input during document assembly.

Each answer is in the format Name=Value. This results in a data block, like this:

```
Name1=Value1&Name2=Value2&Name3=Value3
```

The format for answer names and values is URL-encoded text, with numbers being represented by text. For example, the number 25 would be represented by the characters 2 and 5. (For more information on URL-encoded text, see URL-encoded Text.)

The ampersand character that separates HotDocs answers and the equal sign that separates names and values are not URL-encoded. The script needs to use these characters to delimit the answers, names, and values before it decodes the names and values.

Answer Names

All answer names are 31 characters long. (Names for repeated answers are longer, however. See Names of Repeated Answers.) The first character of the name is a capital letter code indicating the value type of the answer value (which is always URL-encoded text). This value type corresponds to the variable type of the HotDocs variable that gathered the answer.

Code	Value
т	Text
Ν	Number
D	Date
В	True/False (Boolean)
М	Multiple Choice

The other 30 characters of the answer name are the variable name of the variable that gathered the answer. If the variable's name is shorter than 30 characters, space characters are appended to the name to bring the total to 30.

The template developer needs to give you the complete answer names (value type code plus variable name).

Answer Values

The format of an answer value depends on the value type of the answer, as specified by the value type code in the answer name.

Туре	Format	
Text	Any text	
Number	Number characters with an optional decimal point	
Date	dd mm yyyy (fixed length of 10 characters)	
True/False	Either TRUE or FALSE	
Multiple Choice	Sections of text separated by vertical bars (for example, No Pets Covered Parking AC)	

Answer File History

Every collection of HotDocs answers uploaded to an Internet resource has an Answer File History answer. This answer contains the name of the template used to gather the answers and the date and time it was used. The answer name is:

T(ANSWER FILE HISTORY)

The answer value is a text string in the following format:

template_name : month_name, day, four_digit_year, hh:mm

The time is the local time of the machine on which the template was used and is expressed in 24-hour format. The following is an example of an Answer File History answer (after URL-decoding):

```
T(ANSWER FILE HISTORY) =MyTemplate : April 9, 2002, 18:15
```

You can use HotDocs Server to process uploaded answers. Click here for an example of an ASP page that processes uploaded answers using HotDocs Server. (Contact your HotDocs sales representative for information on obtaining a HotDocs Server license.)

When the upload answers script executes and the server encounters an error, it sends an error report back to HotDocs. This error report is saved in a file called *uploadOutput.htm*. It is usually stored in your default temporary folder.

HotDocs Models

Introduction: Create HotDocs Models

As you draft documents for your clients, you frequently modify existing documents by replacing the changeable text in the document (such as names, dates, and figures) with the current client's information. This process can be time-consuming and error-prone, especially because you must go through this process for each new client or matter.

One way you can solve this problem is to create a HotDocs template from the document. This process requires someone with HotDocs knowledge to replace changeable text with variables, conditions, and scripts. Depending on how complex the template is, sometimes the automated template can be difficult for non-HotDocs developers to read and understand.

Another option is to instead create a *HotDocs Model*. HotDocs Models are ideal for subject matter experts who want to review a template without having to learn or install HotDocs on their local machine. The variable and instructions are clearly visible in the document text and can be color-coded to differentiate between types (see Apply Color to Markup). This makes the entire template process easily accessible for the non-HotDocs user and allows them to edit the HotDocs Model before passing it back to their HotDocs developer.

Once you have a HotDocs Model, you can use it with HotDocs to generate documents for your clients. If you are using HotDocs Developer, you can also convert your HotDocs Models to HotDocs template format so they can be added to your collection of templates.

For example, say you have a will that could include different provisions for different inheritance scenarios. Some provisions should be included if you're drafting a will for one type of client, but those same provisions should be left out for other types of clients. With a HotDocs Model, you keep all of the different text variations in a single document, rather than storing the text in separate documents. Then you can mark up the document so that only the correct versions are used, depending on answers the user gives.

As you mark up the HotDocs Model directly in Microsoft Word, your documents must be in DOCX, DOC or RTF format.

Creating a Simple HotDocs Model

Basic Markup

A HotDocs Model is an exemplar document used as the basis for drafting documents of the same type for specific clients. The markup used in HotDocs Models to identify text that varies from one client to another provides an unambiguous description of how a HotDocs Model should be used to draft client-specific documents.

This topic describes only the basics of Markup. For a complete description, see the section of this help file entitled "Creating Full HotDocs Models".

The process of marking up a HotDocs Model can be broken into several steps, as follows:

- 1. Mark Variable Text
- 2. Mark Conditional Text
- 3. Mark Repeated Text
- 4. Mark Special Characters
- 5. Hide Markup Clutter
- 6. Apply Color to Markup

Next Topic 🕨

Previous Topic Next Topic

Mark Variable Text

You can replace names, dates, amounts, and other variable text throughout a document with variable fields. A variable field consists of a name and type, separated by a semicolon and enclosed in square brackets. For example, the document text:

The client, Jane Doe, hereby rescinds all previous claims.

is marked up as:

The client, [ClientName;te], hereby rescinds all previous claims.

In this example, **ClientName** is the variable name and **te** (which stands for text) is the type.

Variable names:

- Can include letters, numbers, and underscores but they cannot contain spaces or other special characters.
- Are case-sensitive (meaning that GiftAmount and giftamount are different names).
- Must not be composed entirely of capital letters.
- Must be 50 characters or less in length.

In general, a variable name consists of at least two words that identify the subject and the aspect of the field. Best practice is to capitalize the first letter of each word in a variable name, regardless of its part of speech. Some examples:

ClientCity	Client is the subject and City is the aspect.	
ClientZipCode	Client is the subject and ZipCode is the aspect.	
AgreementEffectiveDate	Agreement is the subject and EffectiveDate is the aspect.	
TermOfAgreement	Agreement is the subject and TermOf is the aspect.	

Variable field types are specified using the first two letters of the word that represents the type:

- **te** for text fields
- **nu** for number fields
- **da** for date fields

When no type is specified in a field (for example, [ClientName]), text is assumed as the type.

Variable fields may optionally include a format following the field type. For example, if you want the client's name to be inserted into the document in all uppercase letters, use:

The client, [ClientName;te;format=upper], hereby rescinds all previous claims.

Such markup would produce the following in a completed document:

The client, ANNA JAMES, hereby rescinds all previous claims.

In the example field above, notice the semicolon between the field type and the word **format**. Two commonly used formats are:

- upper forces alphabetic characters to uppercase
- **alpha** spells out a number instead of using digits

For a complete list of Markup formats, see Format Answers in the Document.

Previous Topic Next Topic

◀ Previous Topic Next Topic ▶

Mark Conditional Text

Text in a document is conditional if it is included only when certain conditions are met. To mark conditional text, enclose the text in square brackets. Immediately following the opening bracket, type the

word **if** followed by a colon. After the colon, type a condition name followed by another colon. For example:

[if:ClientIsMarried:I give to my spouse all interest in property, both real and personal, including buildings, fixtures and improvements to such property.]

A condition name is an affirmative statement specifying the conditions under which the conditional text is included in the document. All of the rules that apply to variable field names apply to condition names.

When conditional text consists of a complete paragraph, it is usually desirable to include the paragraph mark as part of the conditional text. To do this, type the closing bracket after the paragraph mark, such that it appears at the beginning of the following paragraph. For example:

```
My name is [ClientName;te]. ¶
[if:ClientIsMarried:I am married to [SpouseName;te]. ¶
][if:ClientHasChildren:My children are ¶ [ChildrenNamesAndBirthdates;te]. ¶
]Except as specifically noted in this document … ¶
```

Conditional text fields may be "nested" within other conditional text fields. For example:

[if:ClientIsMarried:I give to my spouse all interest in property, both real and personal, including buildings

[if:GiftsIncludeFixtures:, fixtures,] and improvements to the property.]

Conditional text fields may include one or more alternatives, only one of which will be included in the final document. To specify alternatives, introduce each alternative with a forward slash (/) character followed by **elseif** or **else**. Use **elseif** if the alternative is conditional. Use **else** if the alternative is conditionless, meaning that it is included in the final document when no other alternative is included.

Here is an example showing several conditional alternatives followed by a conditionless alternative:

```
I, [ClientName;te;format=upper], direct my personal representative to see that
my body is [if:ClientToBeBuried:buried in the
[ClientBurialCemetery;te]/elseif:ClientToBeCremated: cremated and my ashes
scattered whithersoever my personal representative shall
direct/elseif:ClientBodyToMedicine:donated to
[ClientBodyInstitution;te]/else:disposed of as my personal representative shall
direct].
```

A conditionless alternative (else) is not required, but if present, it must follow all conditional alternatives.

When a HotDocs Model includes large sections of conditional text or nested conditional text, it can be difficult to find the beginning and ending of conditional text fields. If you have installed the HotDocs

Markup Tools, use the **Select Field** button to easily see where the conditional text field begins and ends. (Click once to select or highlight the immediate condition, and click twice to view the entire condition.)



◀ Previous Topic Next Topic ▶

Mark Repeated Text

Repeated text is text that can appear in a document two or more times in succession. Repeated text can consist of anything from a few words to an entire paragraph. To mark repeated text, enclose a single instance of the text in square brackets. Immediately following the opening bracket, type the word **repeat** followed by a colon (:). After the colon, type a repeat dialog name followed by a semi-colon and the list of variables or text you want included in the repeat instruction, separated by forward slashes.

The following is an example of repeated in-line text:

```
My children are: [repeat:ChildList;format="a; b; and c";contents=ChildName/ChildBirthDate:[ChildName], born [ChildBirthDate;da]].
```

In this example, the entire paragraph is repeated:

```
Specific Gifts
[repeat:SpecificGiftList;contents=SpecificGiftRecipientRelationship/SpecificGif
tRecipientName/SpecificGiftItem: To my [SpecificGiftRecipientRelationship],
[SpecificGiftRecipientName], I give [SpecificGiftItem].
```

]

Best practice for a repeat dialog is to have the dialog name consist of at least two words, the first of which identifies the subject of the list and to use the word "List" as the second part of a repeat dialog name, although this is not mandatory. In the example above the repeat dialog name is "SpecificGiftsList" where "SpecificGifts" is the subject and "List" is used to help remind the user that this is a repeated dialog. All of the rules that apply to variable names apply to repeat dialog names.

◀ Previous Topic Next Topic ▶

◀ Previous Topic Next Topic ▶

Mark Special Characters

As you are marking up your document, you may come across characters in your document text that also have a purpose in Markup. Specifically, Markup attaches special meaning to square bracket, forward slash, and backslash characters. If any of these characters are to appear as literal text in the final document, they must be marked in the HotDocs Model to show that they are just text, rather than part of the Markup.

To mark a special character, type a backslash before it in the document. So:

- [is marked as \[
-] is marked as \]
- / is marked as \/
- \is marked as \\

For example, this text in a HotDocs Model:

```
For a complete description, see H:\\Publications\\Complete Description.doc
\[revised 2006\]
```

appears like this in the final document:

```
For a complete description, see H:\Publications\Complete Description.doc [revised 2006]
```

Here are two reasons for marking special characters in HotDocs Models: First, macros used to manipulate Markup in HotDocs Models rely on the proper use of special characters. Unmarked special characters appear to be part of the Markup, causing incorrect macro behavior. Second, document drafting applications (such as HotDocs) that accept HotDocs Models as input cannot operate properly if special characters are not marked.

◀ Previous Topic Next Topic ▶

Previous Topic Next Topic

Hide Markup Clutter

To make a HotDocs Model easier to read, you can hide the type and format information for variable fields and the condition information for conditional text fields. With this "clutter" hidden, this conditional example:

```
I, [ClientName;te;format=upper], direct my personal representative to see that
my body is [if:ClientToBeBuried:buried in the
[ClientBurialCemetery;te]/elseif:ClientToBeCremated: cremated and my ashes
scattered whithersoever my personal representative shall
direct/elseif:ClientBodyToMedicine:donated to
[ClientBodyInstitution;te]/else:disposed of as my personal representative shall
direct].
```

becomes:

```
I, [ClientName], direct my personal representative to see that my body is
[buried in the [ClientBurialCemetery]/cremated and my ashes scattered
whithersoever my personal representative shall direct/donated to
[ClientBodyInstitution]/disposed of as my personal representative shall
direct].
```

Notice that in variable fields, markup from the colon up to the closing bracket is hidden. In conditional text fields, markup after the opening bracket or forward slash up to the first character of the document text is hidden.

To hide markup clutter using the built-in features of Word, select (or highlight) the text to be hidden, choose **Format** on the Word menu bar, choose **Font** in the drop-down menu, and select **Hidden**. To show hidden markup clutter for review or editing, click the **Show/Hide 1** button on the Word toolbar.

If you have installed the HotDocs Markup Tools, you can first click the Apply Clutter button on the toolbar to apply the hidden attribute to markup clutter across the entire HotDocs Model. Then, to easily show or re-hide the clutter you have hidden, click the Hide/Show Clutter button.

Using the Hide/Show Clutter button instead of Show/Hide 1 has the advantage that it shows hidden markup clutter without showing Word formatting marks which can make the HotDocs Model difficult to read.

Previous Topic Next Topic

Previous Topic

Apply Color to Markup

HotDocs Models often use blue to identify variable fields, green to identify conditional text fields, and maroon to identify repeat fields. While color is not mandatory, it is much easier to see markup in a HotDocs Model when color has been applied.

To apply color using the built-in features of Word, select (or highlight) the text to be colored, choose **Format** on the Word menu bar, choose **Font** in the drop-down menu, and select the font color. Or, after selecting text, click the **Font Color** button on the Word toolbar.

If you have installed the HotDocs Markup Tools, you can click the **Apply Color** button on the HotDocs Markup Toolbar to apply the standard markup colors used in this document to all variable fields and conditional text fields in a HotDocs Model.

Previous Topic

Creating Full HotDocs Models

1. Preparing Your Documents

Identify a HotDocs Model

A HotDocs Model is one in which the text of the document changes, depending on the user. Frequently, a HotDocs Model also contains variations of text, which you can change, add, or remove, depending on the user. Examples of HotDocs Models include wills, legal filings, real estate contracts, and so forth.

For example, say you have a will that could include different provisions for different inheritance scenarios. Some provisions should be included if you're drafting a will for one type of client, but those same provisions should be left out for other types of clients. With a HotDocs Model, you keep all of the different text variations in a single document, rather than storing the text in separate documents. Then you can markup the document so that only the correct versions are used, depending on answers the user gives.

You mark up HotDocs Models directly in Microsoft Word so your documents must be in DOCX, DOC, or RTF format.

The Markup described in this help file requires the use of square bracket, forward slash, and backslash characters. If a HotDocs Model includes any of these special characters, you must mark them to show they are intended to appear as literal text in the document rather than as markup characters. See Mark Special Characters for instructions on doing this.

Next Topic 🕨

◀ Previous Topic Next Topic ▶

Mark Special Characters

As you are marking up your document, you may come across characters in your document text that also have a purpose in Markup. Specifically, Markup attaches special meaning to square bracket, forward slash, and backslash characters. If any of these characters are to appear as literal text in the final document, they must be marked in the HotDocs Model to show that they are just text, rather than part of the Markup.

To mark a special character, type a backslash before it in the document. So:

[is marked as \[] is marked as \] / is marked as \/

```
\ is marked as \
```

For example, this text in a HotDocs Model:

```
For a complete description, see H:\\Publications\\Complete Description.doc
\[revised 2006\]
```

appears like this in the final document:

```
For a complete description, see H:\Publications\Complete Description.doc [revised 2006]
```

Here are two reasons for marking special characters in HotDocs Models: First, macros used to manipulate Markup in HotDocs Models rely on the proper use of special characters. Unmarked special characters appear to be part of the Markup, causing incorrect macro behavior. Second, document drafting applications (such as HotDocs) that accept HotDocs Models as input cannot operate properly if special characters are not marked.

```
◀ Previous Topic Next Topic ▶
```

◀ Previous Topic Next Topic ▶

Rules for Naming Variable Fields

As you name your variables, you must follow these rules:

- Names can include letters, numbers, and underscores but they cannot contain spaces or other special characters.
- Names must begin with a letter.
- Names are case-sensitive (meaning that GiftAmount and giftamount are two different names).
- Names must not be composed entirely of capital letters.
- Names must be 50 characters or less in length.

By default, when you mark up a HotDocs Model, you are required to follow the naming rules described above—particularly by using only letters, numbers, and underscores in field names. If you haven't followed these rules, when you attempt to draft or assemble a HotDocs Model, HotDocs will generate an error message and stop the assembly process. When automating templates in HotDocs, however, you can include spaces and non-alphanumeric characters in variable names. This discrepancy between naming standards becomes problematic when you convert a HotDocs Template to a HotDocs Model and then try to assemble the HotDocs Model. In order to remain compatible, you can define a setting either in HotDocs Options or in the HotDocs Model that allows you to use non-standard names (or names that have spaces and non-alphabetic characters) for fields. See Define Settings for a HotDocs Model for details.

A variable field name typically consists of at least two words that identify the subject and the aspect of the field. Some examples are:

ClientCity	Client is the subject and City is the aspect.	
ClientZipCode	Client is the subject and ZipCode is the aspect.	
AgreementEffectiveDate	Agreement is the subject and EffectiveDate is the aspect.	
TermOfAgreement	Agreement is the subject and TermOf is the aspect.	

While you can use any scheme you choose for naming variables, it's recommended that you capitalize the first letter of each word in a variable field name, regardless of its part of speech, as is shown in the examples above. One reason for doing this is because during a HotDocs interview, HotDocs displays the questions from the document so the user can answer them. If no prompt is specified for a variable in the markup, HotDocs can generate default prompts based on this scheme, adding spaces where each word is capitalized. So, for example, the variable field **ClientName** will automatically be assigned the prompt **Client name**.

◀ Previous Topic Next Topic ▶

Previous Topic Go to Section 2. Marking Up the Document

Steps for Creating a HotDocs Model

The following instructions define, in simple terms, the steps you go through to create a HotDocs Model. The instructions include links to topics that describe the process in further detail.

The basic steps for creating a HotDocs Model include:

- Marking variable fields.
- Marking conditional text.
- Marking repeated text.
- Grouping variables in dialogs and creating an interview.

To create a HotDocs Model with variable fields

- 1. Open the document in Microsoft Word.
- 2. Identify the text that will change, depending on the user. For example, find all names, dates, descriptions, and so forth.
- 3. Replace the text with variable fields. For example, replace all references to the client's name with a markup field, like **[ClientName;te]**. (See Mark Variable Text.)

You can use Word's **Find and Replace** feature to find multiple instances of the same word in the document.

4. Specify any additional information about the variables in tables at the end of the document. (See Define Field Properties.)

To mark conditional text in the document

- 1. Identify the sections of the document that are conditional.
- 2. Mark these sections using any combination of **IF**, **Else IF**, or **Else** fields. (See Mark Conditional Text.)
- 3. Optionally, hide the conditions in the fields to make the markup easier to read. (See Manage Markup Fields in the Document.)

To mark repeated text in the document

- 1. Identify the sections of the document that must be repeated.
- 2. Mark these sections using **REPEAT** fields. (See Mark Repeated Text.)
- 3. Define the dialogs for the variable fields you need repeated. (Make sure you also define repeat styles for the dialogs.) (See Group Variables in Dialogs.)
- 4. Optionally, hide the repeat field properties to make the markup easier to read. (See Manage Markup Fields in the Document.)

To group variables into dialogs

- 1. In the tables section of the document, create a **Dialogs** table.
- 2. List the variables you want included in each dialog, in the order you want the variables asked, in the **Contents** column of the table.
- 3. Optionally, create an Interview table and define the order you want the dialogs asked in the interview. (See Define the Order of Dialogs in the Interview.)

Once your documents are marked up, you can either assemble them using HotDocs, or you can convert them to HotDocs template format. For instructions on doing this, please see Create HotDocs Template from Model or Assemble a HotDocs Model.

You can also define settings that control how the HotDocs Model is assembled. For details, see Define Settings for a HotDocs Model.

For examples of how to mark up a HotDocs Model, see Simple Markup Example (Contract), Simple Markup Example (Agreement), and Complex Markup Example with Tables (Last Will and Testament).

Previous Topic Go to Section 2. Marking Up the Document

2. Markup the Document

Mark Variable Text

When marking up a HotDocs Model, you can identify text that changes in the document, depending on the user. If you are careful to use the Markup described in this help file, you can convert a HotDocs Model to HotDocs template format, or you can assemble the document directly in HotDocs, while still maintaining a reviewable copy of the document.

As you mark up the HotDocs Model, you replace names, dates, amounts, and other client-specific or matter-specific text with variable fields.

There are six different field types:

Field Property	Name Description	
text (or te)	Used for text fields	
number (or nu)	Used for number fields (monetary amounts, figures, etc.)	
date (or da)	Used for date fields	
true/false (or tf) Used for true/false fields (text that is represented by yes/no or true, answers, or text that is merged whether a condition is true or false)		
multiple choice (or mc) Used for predefined options, such as gender. (See Customize a Mu Choice Variable for other information about multiple choice fields.		
computation (or co)	Used for computation fields (values that are calculated, etc.)	

Variable field types can be specified using either the full type name (like **text**, **number**, **date**, **true/false**, **multiple choice**, and **computation**) or the first two letters of the word that represents the type (like **te**, **nu**, **da**, **tf**, **mc**, and **co**, respectively). Field types are case-insensitive, but it is recommended you specify them using lowercase letters.

See Rules for Naming Variable Fields for an explanation of how to name your variables

At its most basic, a variable field should include a name and a field type; however, it can also include other properties, such as a format that defines how the answer should appear in the final document.

A field is enclosed in square brackets. The field name appears first, followed immediately by a semicolon and then the field type. Any other properties you need to assign to the field may appear after the type and must be separated by semi-colons. (You can also define these properties in a variable table. See Define Field Properties for details.)

For example, the following text:

The client, JANE DOE, hereby rescinds all previous claims.

would be marked like this:

The client, [ClientName;te;format=upper], hereby rescinds all previous claims.

In this example, **ClientName** is the variable field name and **te** (which stands for **text**) is the type. The **format** should be **upper** (which stands for **uppercase**).

You can assign additional properties to a field. For a list of these properties, see Define Field Properties.

See Simple Markup Example (Contract), Simple Markup Example (Agreement), and Complex Markup Example with Tables (Last Will and Testament) for examples of how to mark up a HotDocs Model.

◀ Previous Topic Next Topic ▶

◀ Previous Topic Next Topic ▶

Format Answers in the Document

As you mark up the HotDocs Model, you may want an answer to appear in a special format. For example, you may want to spell out a number (twenty-five vs. 25) or you may want to use the long form of a date (such as 14th day of March, 2007). To do this, you can assign a format property to a variable. When the document is assembled, the answer will be formatted according to the markup.

To format an answer, you first define the format property, followed by an equal sign and the specific format you want to use.

For example, the following marked up date:

```
[AgreementDate;da;format="dth day of Mn, YYYY"]
```

would merge the following answer in the document:

6th day of November, 2008

The type of format you assign depends on the type of markup field you're inserting. See the tables below to view the available formats:

Text Formats

Property Name	Description	Defined As	How It Formats the Answer
upper	All letters upper case	format=upper	THIS IS THE ANSWER.
lower	All letters lower case	format=lower	this is the answer.
title	First letter of each word uppercase	format=title	This Is The Answer.

First letter of first word upper case sentence

format=sentence This is the answer.

The sentence format capitalizes the first letter of the first word in the answer, regardless of the answer's placement in the paragraph or document.

Number Formats

Property Name	Description	Defined As	How It Formats the Answer
alpha	Case-sensitive, spelled out number	format=alpha	nine
		format=Alpha	Nine
		format=Alpha	NINE
ordinal	Case-sensitive, spelled out ordinal number	format=ordinal	ninth
		format=Ordinal	Ninth
		format=ORDINAL	NINTH
09	Number including a leading 0 if the	format=09	04
	user enters one in the answer		78
9	Simple numeral	format=9	98
			7,952
9 1/8	Fraction	format="9 1/8"	2 1/4
9,999.00	Monetary amount with cents (even if	format=9,999.00	9,216.00
	no cents are entered)		9,216.92
9.9	Whole numeral with optional decimal places	format=9.9	87.1254
9999	Numeral with no thousands separator	format=9999	12875
9th	Ordinal number	format=9th	23rd
Nine Dollars and Twelve cents	Spelled out monetary amount	format=Nine Dollars and Twelve cents	Five Dollars and Thirty-Six Cents

ninth	Spelled out ordinal number	format=ninth	thirteenth
		format=Ninth	Thirteenth
		format=NINTH	THIRTEENTH

Date Formats

HotDocs 2008 and Earlier Users

Property Name	Description	Defined as	How It Formats the Answer
d	Numeric day	format=d	1
m	Numeric month	format=m	1
У	Numeric year (4 digits)	format=y	2008
dd	Two-digit numeric day	format=dd	01
mm	Two-digit numeric month	format=mm	01
уу	Two-digit numeric year	format=yy	08
уууу	Four-digit numeric year	format=yyyy	2008
dth	Case-sensitive, numeric ordinal day	format=dth	1st
		format=dTH	1ST
dy	Case-sensitive, spelled out day	format=dy	first
		format=Dy	First
		format=DY	FIRST
mn	Case-sensitive, spelled out month	format=mn	january
		format=Mn	January
		format=MN	JANUARY
yr	Case-sensitive, spelled out year	format=yr	two thousand eight
		format=Yr	Two Thousand Eight

		format=YR	TWO THOUSAND EIGHT
wd	Case-sensitive, spelled out weekday	format=wd	monday
		format=Wd	Monday
		format=WD	MONDAY
mnt	Case-sensitive month abbreviation	format=mnt	jan
		format=Mnt	Jan
		format=MNT	JAN
wdy	Case-sensitive weekday abbreviation	format=wdy	mon
		format=Wdy	Mon
		format=WDY	MON

HotDocs 2009 and Later Users

Property Name	Description	Defined as	How It Formats the Answer
d	Numeric day	format=d	1
dd	Two-digit numeric day	format=dd	01
ddd	Case-sensitive weekday abbreviation	format=ddd	mon
		format=Ddd	Mon
		format=DDD	MON
dddd	Case-sensitive spelled-out weekday	format=dddd	monday
		format=Dddd	Monday
		format=DDDD	MONDAY
dth	Case-sensitive numeric ordinal day	format=dth	1st

		format=dTH	1ST
dy	Case-sensitive spelled-out day	format=dy	first
		format=Dy	First
		format=DY	FIRST
m	Numeric month	format=m	1
mm	Two-digit numeric month	format=mm	01
mmm	Case-sensitive month abbreviation	format=mmm	jan
		format=Mmm	Jan
		format=MMM	JAN
mmmm	Case-sensitive spelled-out month	format=mmmm	january
		format=Mmmm	January
		format=MMMM	JANUARY
у	Numeric year	format=y	2009
уу	Two-digit numeric year	format=yy	09
уууу	Four-digit numeric year	format=yyyy	2009
yr	Case-sensitive spelled-out year	format=yr	two thousand nine
		format=Yr	Two Thousand Nine
		format=YR	TWO THOUSAND NINE

True/False Formats

Property Name	Description	Defined as	How It Formats the Answer
true/false	If the answer is true, merges text to the left of the forward slash; if false, merges the text to the right of the slash	format="true/false"	True text False text

yes/no	If the answer is true, merges text to the left of the forward slash; if false, merges the text to the right of the slash	format="yes/no"	True text False text
truetext/falsetext	If the answer is true, merges text to the left of the forward slash; if false, merges the text to the right of the slash	format="truetext/falsetext"	True text False text

List Formats

Property Name	Description	Defined as	How It Formats the Answer
a, and b	Comma after the first in the series, even if there are only two items in the series	format=a, and b	apples, and oranges
a, b	Leaves out the conjunction	format=a, b	apples, oranges, cherries
a, b and c	Leaves out the comma before the final item in the series; items in list are not capitalized	format=a, b and c	apples, oranges and cherries
A, b and c	Leaves out the comma before the final item in the series; first item in list is capitalized	format=A, b and c	Apples, oranges and cherries
A, B and C	Leaves out the comma before the final item in the series; items are capitalized	format=A, B and C	Apples, Oranges and Cherries
A, B AND C	Leaves out the comma before the final item in the series; items are capitalized; conjunction is upper case	format=A, B AND C	Apples, Oranges AND Cherries
a, b or c	Uses the conjunction or to separate items in the list	format=a, b or c	apples, oranges or cherries
a, b, and c	Includes the last comma in the series	format=a, b, and c	apples, oranges, and cherries
a; b; and c	Uses semi-colons to separate items in the list	format=a; b; and c	apples; oranges; and cherries

Previous Topic Next Topic

◀ Previous Topic Next Topic ▶

Define Field Properties

When marking up a document, there are two aspects of the markup you must be aware of—how the answer will appear in the final document and how the answer will appear in the HotDocs interview used to create the final document. You can define these properties in two different places—directly in the field or in properties tables at the end of the document.

This topic includes the following sections:

- Creating Properties Tables for Storing Properties
- Specifying How Answers Are Merged in the Final Document
- Specifying How Questions Appear During the Interview

Creating Properties Tables for Storing Properties

When assigning multiple properties to a variable, rather than include the properties directly in the field, you can include them in a properties table. Using a properties table lets you specify multiple properties for each of your variables without cluttering the actual text of the document.

Properties tables are Word tables that are located at the end of the document, after a special **[EndDocument]** marker. While you can format the look of the table any way you choose (for example, by making text in the table bold or by shading cells), the actual content of the table must be organized a specific way:

- The first row of the table must identify the variable type. (In the example below, this is Text Variables.) Nothing else should be specified in this row.
- The second row of the table lists the most common properties you assign to all your variables (for example, prompts). This row must include the **Name** column, but all other columns are optional and can represent the properties you assign most frequently to your variables. If there are more properties for some variables than can be displayed reasonably in the table, you can include an **Additional** column, where you list the additional properties. (Separate the properties in this column using a semi-colon.)
- The third and all following rows list the individual variables with their associated property values.

Properties are separated by semi-colons. Properties are identified in name/value pair combinations. For example, if you want to create a number variable, you would define the following properties:

```
format=9,999.00;currency=$;decimal=2
```

When you use variable tables, you must specify the place in the document where the document text ends and the tables begin. You do this using the **[EndDocument]** field marker. It appears after the last section of document text but before the first variable table.

Here is an example of an [EndDocument] marker, followed by a variable table:

[EndDocument]

TEXT VARIABLES

Name	Title	Prompt	Additional
EmployeeName	Name of Employee	Enter the employee's name	format=upper; resource="All employees must submit to a background check to verify their eligibility to work for Hobble Creek Publishing."
JobTitle	Job Title	Enter the employee's job title	
JobDescription	Job Description	Complete the following sentence: Job duties shall include:	height=3

When you assemble the HotDocs Model using HotDocs (or when you create a HotDocs template from the HotDocs Model), HotDocs looks for variable tables and uses the information in them to create and set properties for HotDocs variables. If no tables are found, HotDocs uses the properties defined in the markup fields of the document.

See Complex Markup Example with Tables (Last Will and Testament) for an example of a HotDocs Model that uses variable tables.

Specify How Answers Are Merged in the Final Document

You can assign properties to a variable that will affect how a user's answer will appear in the final document. For example, you can assign a format for the answer (such as uppercase or spelled out), whether the answer should break across lines of text or not, and how the field should look if the user chooses not to answer it, just to name a few.

These properties can be defined in one of two places—in the variable field (described just below) or in a properties table (described earlier).

Properties are separated by semi-colons and are identified in name/value pair combinations. For example, the following would represent a text variable and how it should be formatted in the document:

[ClientName;te;format=upper;nonbreak=yes]

Specifically, this markup field will merge the client's name in uppercase, and it will be non-breaking (meaning the name won't break across lines in the document).

When a property is assigned at the field level, it will apply to that field only. If a property is defined in a table, it will be used for all variable references. Where a property is defined both in the field and in a table, the field-level property will take precedence.

The following table describes all of the available field properties with their associated names and values. Default property values are in parentheses. When a property is not specified in a variable field, the default value is assumed. You can include as many properties as necessary, as long as you separate each property with a semi-colon.

Property Name	Description	Value
type	This specifies the variable type	(text) or (te) number or nu date or da true/false or tf multiple choice or mc computation or co
format (text fields)	This is the format for text fields, including text variables, multiple choice variables, and computation variables that produce a text answer. It specifies how the answer will be formatted in the final document.	(none)
, ,		When you don't specify a format, text is formatted as the user enters it.
		upper
		lower
		title
		sentence
format (number	This is the format for number fields, including number variables and computation variables that produce a number answer. It specifies how the answer will be formatted in the final document.	(none)
fields)		9,999.00
		Nine
		9th
		etc.
format (date fields)	This is the format for date fields, including date variables and computation variables that produce a date answer. It specifies how the answer will be formatted in the final document.	(none)
(dute fields)		d/m/yy
		dd/mm/yyyy
		Mn, D, Y

HotDocs Models

		etc.
format (true/false	This is the format for true/false fields, including true/false variables and	(none)
fields)	computation variables that result in true/false. It specifies how the answer will	true/false
	be formatted in the final document. If the value is true, the text to the left of the "/"	yes/no
	will be merged. If the value is false, the text to the right of the "/" will be merged.	truetext/falsetext
format (multiple	This is the list format, which is used with multiple-select multiple choice variables. It	For single-select options:
choice fields)	specifies how the answer will be formatted in the final document.	upper
		lower
		title
		sentence
		For multiple-select options:
		A, B and C
		a; b; and c
		etc.
nonbreak	This property can be used to keep users'	(no)
	final document.	yes
unanswered	This property specifies the text that will be merged into the final document when the variable is unanswered.	any text
font	This property specifies the font that will be	(default font used in document)
	the final document.	font name
comment	This is a comment for the field. It does not affect assembly and will not be merged into	(none)
com	the final document. It is used to make in- line notes in the document.	any text

For the full value lists and explanations see Format Answers in the Document

Specify How Questions Appear During the Interview

If you assemble the document using HotDocs, the fields you create in the document will appear as questions in a HotDocs interview where you will enter your answers. As such, you can define how the questions will be formatted during this interview.

Again, properties are separated by semi-colons. They are identified in name/value pair combinations, like this:

format=9,999.00;currency=\$;decimal=2

The following table describes properties common to all variable fields. Subsequent tables then discusses variable-specific properties:

Common Field Properties

Property Name	Description	Value
title	Designates an alternate name for the variable, which is used in the HotDocs interview outline.	any text
prompt	Specifies the text that guides users in knowing how to answer the question during the HotDocs interview.	any text
	If you specify a prompt, it will be used as the interview "question." If you do not, HotDocs will attempt to generate a prompt based on the field name. It does this by inserting a space wherever it finds a capital letter. So, for example, the field name ClientAddress would generate the prompt Client address .	
resource	Provides additional help text for answering the question. This text appears in the resource pane of the HotDocs assembly window	any text
irrelevant (or irrel)	Specifies how a variable that isn't used in the document (for example, because it's conditioned and the condition resolves to false) should be treated—whether it's grayed out, hidden, or always shown.	(gray) hide
		show
ask	Specifies whether a variable or dialog is asked automatically during the interview.	(yes)
		no
save	Specifies whether the answer the user enters can be saved to an answer file at the end of an interview.	(yes)

		no
warn	Causes HotDocs to display a warning if the variable is left unanswered during the	(yes)
		no

Text Fields

Property Name	Description	Value
height	Determines the height of the answer field that appears in the interview, allowing it to show more than a single line of text	(1)
		Any number between 1 and 12
maximum	Determines how many characters can be used in the answer	(as many as the user enters, up to 15,000)
(or max)		
pattern Detern	Determines a specific pattern that will be used for entering the answer (such as a telephone number or time of day)	(none)
		(999) 999-9999
		999-99-9999
		etc.
enter	Determines whether a new paragraph is created when the user presses the Enter key in a multi-line field. (By	(break)
	default, pressing Enter simply starts a new line in the same paragraph.)	paragraph

Number Fields

Property Name	Description	Value
minimum (or min)	Specifies the minimum value for the answer	(0)
,		Any number
maximum	Specifies the maximum value for the answer	(0)
	Any number	
--------------------------------------------	------------------------------------------------------------------------------------------	
Specifies the maximum value for the answer	(0)	
	Any number between 0 and 7	
Specifies the maximum value for the answer	(none)	
	\$	
	£	
	€	
	etc.	
	Specifies the maximum value for the answer Specifies the maximum value for the answer	

True/False Fields

Property Name	Description	Value
style	Determines whether the Yes/No prompts for the variable appear on the same line	(row)
		column

Multiple Choice Fields

Property Name	Description	Value
options	Specifies the options of the multiple choice variable. (You are required to include one or more options with all multiple choice variables.)	Option1/Option2/Option3
optionprompts	Specifies the text that can be used to identify the options (perhaps because the options aren't descriptive enough)	Prompt1/Prompt2/Prompt3
merge	Specifies the text that will be merged in the final document if the user chooses the correlating option during the interview	Merge1/Merge2/Merge3
optionresources	Specifies the resource text for each option you've defined	Resource1/Resource2/Resource3
select	Specifies whether the user can choose one option or multiple options	(single)

HotDocs Models

	multiple
Determines whether the user can enter an "other" option for single-select options	(no)
	yes
Determines whether the user can select a "none of the above" option for multiple-select options	(no)
	yes
Specifies how the options will be presented to users in the interview	(dropdown)
	(grid)
	column
	list
	The default option depends on whether the field is set to multiple selection or single selection.
	Determines whether the user can enter an "other" option for single-select options Determines whether the user can select a "none of the above" option for multiple-select options Specifies how the options will be presented to users in the interview

Computation Fields

Property Name	Description	Value
script	This is the formula or calculation (written in HotDocs scripting language) that determines the value for the answer. It is a required property for computation fields.	A valid HotDocs script
merge	Specifies text that can be merged if the computation script generates a true or false value	Text This property can be used when the Computation field generates a true/false value.

Dialogs

Property	Description	Value	
Name			

title	Specifies the text that is used to represent the dialog in the assembly window title bar and in the interview outline	Any text
	If you do not specify a title, HotDocs will attempt to generate a title, based on the dialog name. It does this by inserting a space wherever it finds a capital letter. So, for example, the dialog name ClientInformation would generate the title Client Information .	
style	Specifies whether the dialog is asked once, as a series of dialogs, as a spreadsheet, or as a spreadsheet on the parent dialog.	(regular)
		repeated
		spreadsheet
		ssonparent
contents	Lists the variables that are to be included in the dialog. (The order in which you list these variables will be the order they are asked in	Variable1/Variable2/
(required)	the dialog.)	Variable3
group	Groups all True/False variables so they are represented either by check boxes (which allow users to choose multiple options) or	(none)
	option buttons (which allow users to choose only one option).	single
		multiple
none	Specifies whether a grouped list of True/False variables includes a None of the Above option.	(no)
		yes
label	Specifies the text that will be used to identify the entire group of answers in a repeated list	Any text
		This option can only be used when the dialog's style is set to repeated
rows	Specifies the number of rows that are visible in a spreadsheet	Any number
	dialog. (This does not affect the number of answers a user can enter—just how many rows of the spreadsheet are visible at a given time.)	This option can only be used when the dialog's style is set to Spreadsheet

prompt	Shows the prompt that will be used if the dialog is inserted in a parent dialog Any text	Any text
irrelevant	Causes the dialog to be hidden when all the variables in the dialog are inactive	(hide)
		show

Language Fields

Property Name	Description	Value
decimal	Indicates the character that will be used to show the decimal character in a non-English number field	Any character (usually a comma)
grouping	Indicates the character that will be used to show the thousands separator in a non-English number field	Any character (usually a period)

Insert Fields

Property Name	Description	Value
keep	When inserting another document in the parent model, lets you keep the header and/or footer from the inserted document	(none)
		header
		footer
		both
grouping	Indicates the character that will be used to show the thousands separator in a non-English number field	Any character (usually a period)

See Simple Markup Example (Contract), Simple Markup Example (Agreement), and Complex Markup Example with Tables (Last Will and Testament) for examples of how to mark up a HotDocs Model.

You can have HotDocs define default prompts and titles for fields in the HotDocs Model. See Define Settings for a HotDocs Model for details.

◀Previous Topic Next Topic ▶

Previous Topic Next Topic

Mark Conditional Text

Text in a document is conditional if it is included only when certain conditions are met.

There are two types of conditions you use in markup—simple conditions and alternative conditions.

Marking Simple Conditions

To mark conditional text, you surround the text using opening and closing square brackets. Immediately after the opening bracket, type the word **if** followed by a colon. After the colon, type a condition name followed by another colon. The document text comes after this last colon.

In this example:

[if:ClientIsMarried:I give to my spouse all interest in property, both real and personal, including buildings, fixtures and improvements to the property.]

The opening bracket, followed by **if**:, indicates the beginning of the condition. The condition name, **ClientIsMarried**, is a true/false variable that has been defined in a True/False Variables table. The closing square bracket indicates the end of the conditional text.

A condition name is an affirmative statement specifying the conditions under which the conditional text is included in the document. This can be a true/false variable, a computation variable that returns a true/false value, or it can be an expression. If the condition name isn't defined elsewhere in the document (for example, as a variable field inserted elsewhere in the document or in a True/False Variables table), HotDocs will use the condition name to create a simple true/false variable. All of the rules that apply to variable field names apply to condition names.

When conditional text consists of a complete paragraph, it is usually desirable to include the paragraph mark as part of the conditional text. To do this, type the closing bracket after the paragraph mark, such that it appears at the beginning of the following paragraph. The following example, which shows the paragraph marks, demonstrates this:

My name is [ClientName;te]. ¶
[if:ClientIsMarried:I am married to [SpouseName;te]. ¶
][if:ClientHasChildren:The names and birth dates of my children are
[ChildrenNamesAndBirthdates;te]. ¶
]Except as specifically noted in this document … ¶

Additionally, conditional text may be "nested" within other conditional text. For example:

[if:ClientIsMarried:I give to my spouse all interest in property, both real and personal, including buildings[if:GiftsIncludeFixtures:, fixtures] and improvements to the property.]

Marking Alternative Text

Conditional text may include one or more alternatives, only one of which will be included in the final document. To specify alternatives, introduce each alternative with a forward slash character (/) followed by **elseif** to identify a conditional alternative, or by **else** to identify a conditionless alternative that will be included in the document if no other alternative is included.

Here is a simple example showing a conditionless alternative:

```
I nominate my [if:ClientIsMale:wife/else:husband], [SpouseName;te] as my
personal representative.
```

Here is a complex example with several conditional alternatives followed by a conditionless alternative:

```
I direct that my body be [if:ClientToBeBuried:buried in the
[ClientBurialCemetery;te]/elseif:ClientToBeCremated:cremated and that my ashes
be [ClientAshDisposition;te]/elseif:ClientBodyToMedicine:donated to
[ClientBodyInstitution;te]/else:disposed of as directed by my personal
representative].
```

A conditionless alternative (else instruction) is not required, but if you include one, it must appear after all other conditional alternatives (else if instructions).

See Simple Markup Example (Contract), Simple Markup Example (Agreement), and Complex Markup Example with Tables (Last Will and Testament) for examples of how to mark up a HotDocs Model.

Previous Topic Next Topic

Previous Topic Next Topic

Mark Repeated Text

As you mark up a document, you may need to identify sections of the document that you need to repeat. For example, if you're inserting a list of children, you can cause any variable field (for example, [ChildName;te]) to repeat.

To do this, you surround the text in the document you want to repeat (including variable fields) with repeat field markers. A repeat field consists of an opening bracket, the repeat instruction (including the name of the dialog you want to repeat) and the text you want repeated. You must also include a closing bracket at the end of the repeatable section of text. In the field itself, you must separate the dialog name from its properties using semicolons. After the last defined property, you include a final colon.

For example:

My children are listed as follows: [repeat:Children;format="a, b, and c":[ChildName;te]].

Fields between the brackets are added to the repeated dialog. If you want to change the order in which they appear, you can either define the order in the Dialogs table, or you can include a **Contents**= property directly in the repeat field. Separate the field names using forward slashes, like this: [repeat:BequestList;contents=BeneficiaryName/BequestProperty: I give [BequestProperty;te] to [BeneficiaryName;te]. ...].

When repeating a complete paragraph, it is usually desirable to include the paragraph mark as part of the repeated text. To do this, type the closing bracket after the paragraph mark, such that it appears at the beginning of the following paragraph. For example:

[repeat:SpecificBequestInformation:I give [BequestProperty;te] to
[BeneficiaryName;te]. If [BeneficiaryName;te] is not living upon my death, the
specific bequest shall lapse and be distributed to my then living children.¶]

You can assign the following additional properties to a repeat field:

Property Name	Description	Value
name	Shows the name of the dialog you are repeating	A valid dialog name
contents	Lists the variable fields you want repeated	Variable1/Variable2/Variable3/etc.
format	Describes how HotDocs formats and punctuates a series of answers in a sentence	("")
		"a, b, and c"
		"a; b; and c"
		etc.
ascend	Sorts the list, based on the specified variable, in alphanumeric (A to Z) order	A variable used in the dialog
descend	Sorts the list, based on the specified variable, in reverse alphanumeric (Z to A) order	A variable used in the dialog
filter	Lists a computation variable that limits (or filters) the list of answers you want merged in the final document	A variable used in the dialog
sortfirst	Sorts the list of items before it applies any filters you've specified	(false)

Many of these properties are described in greater detail here.



true

You can sort answers in your list on two different levels by including two ascend or descend properties in the field. The first ascend/descend property will define the first level of sorting, while the second ascend/descend property defines the second level of sorting. For example, you may want to sort all states in a list in alphabetical order. You can then sort all cities within a state in alphabetical order, too.

HotDocs processes the ascend, descend, and filter properties in the order you provide. For example, if you place the ascend or descend properties before the filter property, HotDocs sorts the list of answers before the it applies the filter.

By default, HotDocs sets a repeated dialog having five or fewer variables to appear in the interview as a spreadsheet dialog with just three rows.

Using a repeat instruction in a table

HotDocs handles repeating the information in one cell of a table just like using a repeat anywhere else in the model. It consists of the repeat instruction, the dialog you want to repeat, and the variable or text you want to repeat, all enclosed in square brackets. For example the following table shows how to repeat the information in one cell of a table:

Child's Name	Child's Birth Date	Items
[ChildName;te]	[ChildBirthDate;da]	<pre>[repeat:Item_List:[Items;te]]</pre>

Repeating Rows in a Table

At times, you need to repeat a row of a table. To do this, you insert an opening field marker before the repeat instruction, but then leave off the closing field marker. This is because the end of the row signifies the end of the repeated field. For example, the following shows how to mark up a repeated table:

Child's Name	Child's Birth Date	Items
<pre>[repeat:Children:[ChildName;te]</pre>	[ChildBirthDate;da]	[Items;te]

Note there is no closing bracket after the **Items** field. This is because the end of the table row signifies the end of the repeat field.

Nesting Repeated Fields

You can nest repeat fields inside each other. For example, say you need to create a list of children, and then for each child, list the property items each child will inherit.

There are two steps to doing this: 1) nesting the repeat fields in the HotDocs Model, and 2) listing the nested, repeated dialog as a variable of the main dialog in the Dialogs table.

So, for example, in the document, your text would look like this:

```
[repeat:ChildrenInfo:To [ChildName;te], my child, I give the following items:
[repeat:ItemListInfo;format="a; b; and c":[Item;te]].
```

]

The Dialogs table would look like this:

Dialogs

Name	Title	Contents	Resource	Additional
ChildrenInfo	Children Information	ChildName ItemListInfo		style=repeated; ascend=ChildName
ItemListInfo	Item Information	Item		style=ssonparent

Repeats can not be nested more than four levels deep.

The dialog **ItemListInfo** appears in the Contents list for **ChildrenInfo**. Both dialogs include a repeat style.

The property, ssonparent, stands for spreadsheet on parent. It means that HotDocs displays the dialog as a spreadsheet and embeds the spreadsheet directly in its parent dialog.

For examples of how to mark up a HotDocs Model, see Simple Markup Example (Contract), Simple Markup Example (Agreement), and Complex Markup Example with Tables (Last Will and Testament).

◀ Previous Topic Next Topic ▶

Previous Topic Next Topic

Other Instruction Fields for HotDocs Models

In addition to using IF and REPEAT fields in a HotDocs Model, you can use the following other instructions:

Field Property Name	Title	Contents
ask	When HotDocs creates an interview for the user, it reads through the	[ask:VariableName]
	HotDocs Model and displays dialogs of questions based on the order it	[if:ClientIsMarried:[ask:ClientSpouseName]]

	encounters markup fields in the document.	[if:MONTHS FROM(Date of Previous Filing DA, TODAY) >= 13:[ASK Current Insurance TE]
	However, if you want your dialogs to appear in a different order—for instance, if you want a certain dialog to appear first, even though variables that prompt the dialog to be asked appear at the end of the document— you can use an ask field to force HotDocs to display the dialog.]
default	If a variable field is unanswered, a default field will set the value of a	[default:VariableName=Value]
	variable to a specific value. (If the variable field is already answered, the default instruction has no effect.)	[default:EmployeeStatus=Temporary]
set	Variable fields normally get their values from the answers users enter	[set:VariableName=Value]
	during an interview, but sometimes you may want to assign an answer to a variable instead of allowing the user to specify the answer.	[set:SpouseAddress=ClientAddress]
	For example, a document might include the address of the client and, in another place, the address of the client's spouse. Once the client's address has been entered by the user, you could use a set instruction to automatically fill in the same address for the spouse, since it will be the same.	
increment/	The increment and decrement	[increment:VariableName]
decrement	increase or decrease a number variable, usually a counter, by the	[decrement:VariableName]
	value of 1.	[increment:TempNum]
		[decrement:TempNum]
insert	You can insert one template into another by using an insert field. For	[insert:"FileName.docx"; keep=header]

	example, you might want to include boilerplate text in multiple documents, or you may want to include a set of related documents in one main HotDocs Model so users can choose the document they want. During document assembly, when HotDocs finds an insert field, it stops assembling the main document so it can assemble the inserted document. When it finishes, it continues assembling the main document	[insert:"Provisions.docx"; keep=both] Both the parent document and any documents referenced in the insert field must be saved to the same location.
	An optional property of an insert field is keep , which can have the value of header , footer , or both .	
assemble	You can use an assemble field to add templates to the list of documents that should be assembled. Unlike the insert field, an assemble field adds the model to the list of assemblies (known as the assembly queue) and then waits until the main document is finished assembling before it starts assembling the new, added document.	[assemble:"FileName.docx"] [assemble:"trustdocument.docx"] Both the parent document and any documents referenced in the assemble field must be saved to the same location.
languages	You can mark up a HotDocs Model in languages other than English. To do this, you must create a language field in the document that allows HotDocs to format dates and numbers in the template so they appear correctly in the assembled document. Optional properties of the language field include decimal , which identifies the character used as the decimal mark, and grouping , which identifies the character used as the thousands separator.	 [language:fra;decimal=',';grouping='.'] Valid language codes are: dea (Austrian German) des (Swiss German) deu (German) eng (English) fra (French) nld (Dutch) esn (Spanish) ita (Italian) ptb (Brazilian Portuguese)

span Frequently, users need to edit document text once a document has been assembled. To allow this, you must mark sections of document text using span fields. Inserting span fields in a HotDocs Model allows users to edit the text of the assembled document while viewing the Document Preview tab of the assembly window. Changes made to the text can be saved in an answer file, which allows users to later reassemble the document and still have access to the changes they made. An optional property of the **span** field is title.

[span:spanName;title=SpanTitle:DocumentText]

[span:trialPeriod: The length of [Employee Name]'s employment will be an initial term of six months, with the possibility of continuation beyond that period depending on Hobble Creek Publishing's needs and upon the employee's performance.]

Each of the above fields can also specify a **comment** as an additional property, for example:

[ask:VariableName;comment=put comment here]

Previous Topic Next Topic



Manage Markup Fields in the Document

As you mark up a HotDocs Model, you may want to distinguish between markers and the actual text of the document. For example, you may want to mark fields using specific colors. There are two ways to do this—by using the HotDocs Markup Tools, or by using the built-in features of Word.

Using the HotDocs Markup Authoring Tools

HotDocs Markup Tools are included as part of the HotDocs ribbon, helping you better manage the markup fields in the document. For example, you can apply color to the different fields, as well as hide field properties that aren't essential for reviewing the document. You can also use the toolbar to draft a test document, so you can see what kind of interview the document produces.

You can use the following buttons in the toolbar to complete your different tasks:

Button Name Description

🥍 Apply Color	Applies color to the different fields in a HotDocs Model. Variable fields are colored using blue, If fields using green, and Repeat fields using maroon. Using color like this can help you distinguish between markup and the actual text of the document. (To apply color to a selection of text within the document, select the text before clicking the button.)
	property to the text. This makes it so you can click the Hide Clutter button and hide these properties. This can make the document more readable, particularly when you need to review it.
Hide Clutter	Hides the properties of a variable (such as formats and other styles) that aren't
Show Clutter	clutter in a selection of text rather than the entire document, select the text before clicking the button.) To view the full markup language again, click Show Clutter .
[™] Select Field	Highlights the block of text merged using an IF field or a REPEAT field. This helps you identify all of the text affected by the instruction.
	If you click Select Field once, it will select just the immediate section of the instruction (for example, the first part of a multi-part condition). If you click the button twice, it will select the entire instruction.
Draft Document	Lets you view the HotDocs interview that is created by the HotDocs Model.
Coptions	Lets you view whether the HotDocs Model uses an embedded component file. (When you convert a HotDocs template to a HotDocs Markup, HotDocs embeds the component file in the document so that it can retain its properties. Clicking this button helps you know if this is the case. If you want to delete the embedded component file, click Remove .) (See Convert HotDocs Templates to HotDocs Markups)
Help	Opens the HotDocs Markup Help file.

Using the Features of Microsoft Word

As you mark up the document, you may find that the number of properties you need to specify for a variable field, conditional field, or repeat field makes the field hard to read and understand, especially for someone unfamiliar with markup. You may also have a difficult time distinguishing between mark-up fields and document text. The following describes some things you can do using tools in Microsoft Word to make this process of managing your markup easier.

Keep Properties from Appearing in Markup Fields

To keep the information appearing in markup fields to a minimum, you have several options:

- Place the field properties in hidden text: Once you have assigned all of the properties to the field, apply the hidden text property to the field property. (To do this, highlight the text you want to hide and choose Font (Format menu). At the Font dialog box, select Hidden in the Effects group.) To view the hidden text, click the **Show/Hide 1** button in the Word toolbar.
- Place the field properties in footnotes: Once you have assigned all of the properties to the field, move the text to a footnote. (See the Microsoft Word help for instructions on creating footnotes.)
- Place the field properties in variable tables at the end of the document: You can create tables at the end of the document where you define the properties of the variables you use in the document. See Define Field Properties for an explanation of how to do this.

Apply Color to Markup

The markup examples used in this help file use blue to identify variable fields, green to identify conditional text, and maroon to identify repeated text. While color is not mandatory, it is much easier to see markup in a HotDocs Model when color has been applied.

To apply color using the built-in features of Word, select (or highlight) the text to be colored, choose **Font** (**Format** menu), and select the font color. Or, after selecting text, use the **Font Color** button on the **Word Formatting** toolbar.

Previous Topic Next Topic

◀ Previous Topic Next Topic ▶

Rules and Tips for Creating a HotDocs Model

You should keep the following in mind when working with HotDocs Models:

- Sometimes you may want to include the value from another answer in the prompt for a specific variable. To do this, make sure you include the opening and closing brackets around the variable name. For example, the following prompt Please enter [EmployeeName]'s gender, would merge the answer for the employee's name in the prompt during the interview, like this: Please enter Jane Porter's gender.
- Markup keywords (like **format**, **text**, **style**, and **enddocument**) are case-insensitive. However, variable names (like **ClientName**) are case sensitive.
- Property definitions are cumulative, meaning that if the same property is defined more than once for a given variable or dialog, the last one found in the HotDocs Model takes precedence.
- If you do not define prompts for your variables, by default, HotDocs will attempt to create default prompts based on the field name. For example, the field name [EmployeeName] would become the prompt Employee name. (HotDocs determines where spaces should be included in the prompt based on capitalization in the field name. Because of this, you should capitalize each individual word in the field name.) See Define Settings for a HotDocs Model for details.

• The same is true for dialog titles. If no title is specified, HotDocs will attempt to create a default title, based on the dialog name. For example, the dialog **[EmployeeInformation]** would use the title **Employee Information**. See Define Settings for a HotDocs Model for details.

Escaping Characters In Your Markup

If you need to use characters in your text that are typically reserved for the Markup specification, you can "escape" them by following these guidelines:

Properties appearing in a variable table column do not need to be escaped. The only exception is when you're listing multiple, different properties in the **Additional** column of the table.

• The opening and closing square bracket ([]), forward slash (/), and back slash (\) characters, if used as literal characters anywhere in the HotDocs Model, must always be marked as literal. You can do this by preceding the character with a back slash character.

For example, this text in a HotDocs Model:

For a complete description, see H:\\Publications\\Complete Description.doc
\[revised 2006\]

appears like this in the final document:

For a complete description, see H:\Publications\Complete Description.doc [revised 2006]

• A forward slash (/), if used as a literal character in any list properties (such as a list of multiple choice options), must be marked as literal. You can do this by enclosing the text string in quotation marks.

For example, you would mark the following different options :

Cars/Trucks

Recreational Vehicles

Towing Trailers

Like this:

"Cars/Trucks"/Recreational Vehicles/Towing Trailers

• Paragraph marks or line breaks, if used in any list properties (such as multiple choice options), must be enclosed in quotation marks.

For example, you would mark the following different multiple choice option prompts:

(Option 1) Cars

Trucks (Option 2) Recreational Vehicles (Option 3) Towing Trailers Like this: "Cars Trucks"/Recreational Vehicles/Towing Trailers Previous Topic Next Topic

Previous Topic Go to Section 3. Changing the Appearance of Questions and Dialogs in the Interview

Draft a Test Document

As you are marking up a HotDocs Model, you can use the **Praft Document** button in the HotDocs Markup Tools to draft a test document. When you click this button, HotDocs starts and displays the interview and assembled document for the HotDocs Model you are working on. Being able to see the interview and document can help you make sure the markup you're using in the model creates a complete and accurate document. It also gives you the chance to review how questions appear in the interview, in case you need to define additional properties, such as formats or prompts.

To draft a test document

- 1. Click the Real **Draft Document** button in the HotDocs Markup Tools. A HotDocs assembly window appears.
- 2. Review the questions in the interview, answering them different ways to produce different versions of your document.
- 3. Review the assembled document for accuracy and completeness.

If you need to make corrections, make them in the HotDocs Model.

Once you have completed a HotDocs Model, you can add it to a HotDocs template library so it can be assembled by other HotDocs users. See Assemble a HotDocs Model for details.

Previous Topic Go to Section 3. Changing the Appearance of Questions and Dialogs in the Interview

3. Changing the Appearance of Questions and Dialogs in the Interview

◀ Previous Topic Next Topic ▶

Group Variables in Dialogs

If you plan to assemble HotDocs Models using HotDocs, you may find that you want variables to be grouped together during the interview. For example, you may want variables asking for information about the client (such as name, gender, birth date, and so forth) to be asked in the same group, rather than asked individually.

To group variables into dialogs, you must create a Dialogs table in the HotDocs Model. This table lists each dialog you want to use. For each dialog, you list the variables you want to include in the dialog as well as other properties you want defined for the dialog. For example, a typical Dialogs table may look like this:

DIALOGS

Name	Title	Contents	Resource	Additional
DependentData	Dependent Information	DependentExplained ChildName ChildBirthDate	Please list children in birth order, from oldest to youngest	style=repeated
ClientData	Client Information	ClientName ClientGender ClientBirthDate		
PropertyData	Property Information	PropertyDescription PropertyLocation PropertyValue		style=repeated

Sometimes you may need to add standalone text to a dialog to help the user better understand the questions that are being asked. You can do this either by typing the text directly in the Contents column (surrounding it in quotation marks) or by creating a dialog text variable.

Just as you assign properties to variables, you can assign properties to a dialog. For example, you can define a title for the dialog as well as choose a style for the dialog if it must be repeated. The following table describes these properties:

Property Name	Description	Value
name (required)	Specifies the name of the dialog.	Any valid variable name. (See Mark Variable Text for rules on naming variables.)
contents (required)	Lists the variables and dialogs that are to be included in the dialog. (The order in which you list the contents will be the order they appear in the dialog.)	Variable1/ Variable2/ Variable 3 Variables can be

		separated either by hard returns or by a forward slash.
title	Specifies the title of the dialog. The title is what the user will see during the interview.	Any text
	If you do not specify a title, HotDocs will attempt to generate a title, based on the dialog name. It does this by inserting a space wherever it finds a capital letter. So, for example, the dialog name ClientInformation would generate the title Client Information .	
resource	Provides additional help text for answering all of the questions in the dialog. This text appears in the resource pane of the HotDocs assembly window.	Any text
style	Specifies whether the dialog is asked once, as a series of dialogs,	(regular)
	dialog.	repeated
		spreadsheet
		ssonparent
group	Lets you group True/False variables so they can appear either as option buttons (which let the user choose one of the options) or	(none)
	check boxes (which let the user choose multiple options).	single
		multiple
none	Provides users with a "none of the above" option when True/False	(no)
	neids in the dialog are grouped.	yes
label	Lets you assign a label to a dialog that is repeated as a series. The label is used to identify the top node of the repeated dialog in the interview outline.	Any text
prompt	Specifies the text that will be used to identify a child dialog on its parent.	Any text
rows	Defines the visible number of rows in a dialog that is repeated as a spreadsheet.	Any number
ask	Causes the dialog to be asked automatically in the interview.	(yes)
		no

irrelevant Defines how a dialog that isn't required in the interview should be (hide) treated. For example, if a dialog contains variables that aren't used in the interview, this setting will determine whether the dialog is hidden (because it isn't necessary) or shown.

You can control the order dialogs are asked in the interview. See Define the Order of Dialogs in the Interview for details.

Previous Topic Next Topic

Previous Topic Next Topic

Define the Order of Dialogs in the Interview

As HotDocs processes the HotDocs Model, it displays the dialogs you have created in the order the variable fields are merged in the document. For example, if the first field in the HotDocs Model is **EmployeeName**, HotDocs will ask the variable **EmployeeName** first. If this variable is associated with a dialog, it will ask the dialog instead. It will then move to the next field in the HotDocs Model and repeat the process. (If the dialog has already been asked, it won't ask it again.)

For many, this default order for asking dialogs is adequate. However, if you want more control, you can define this sequence by creating an Interview table. The order dialogs are listed in this Interview table is the order they will be asked by HotDocs.

When defining an interview, you must account for all of the dialogs. Failure to list all of the dialogs in the Interview table will result in some questions not being asked in the interview.

An Interview table lists the dialogs in the order you want them asked. For example:

ClientData DependentData PropertyData

INTERVIEW

Previous Topic Next Topic

Previous Topic Go to Section 4. Using the HotDocs Model in HotDocs

Define Settings for a HotDocs Model

When you work with HotDocs Models, you can define several settings that control how parts of the interview look and feel. For example, you can define options that let HotDocs generate default titles, prompts, and resources for variables in the HotDocs Model.

Additionally, you can select an option that allows you to use non-standard names when marking fields in the HotDocs Model. To explain, the Markup requires you to use only alphanumeric characters and underscores when naming markup fields in your HotDocs Model. However, sometimes when you're converting existing HotDocs templates to model format, variables in the template may use spaces and other special characters in the variable name. Selecting this option allows you to continue to use the template as a model without having to rename all of your variables.

The following table describes the different settings you can include in a Settings table, along with descriptions of each setting:

Property Name	Description	Value
StandardNames	Allows authors to assemble HotDocs Models or convert templates to HotDocs Model where field names include spaces and other special characters. These characters are typically not allowed in HotDocs Model authoring.	(True) False
	When a template is converted to a HotDocs Model and it uses non-standard field names, this setting is automatically added to the Settings table and set to False . If this property isn't defined in the Settings table, HotDocs will use the property defined in HotDocs Options. (See the Allow Non-Standard Names in HotDocs Model and Rules for Naming Variable Fields for details.)	
DefaultPrompts	Generates default prompts based on field names. Prompts are used to identify the question in a dialog. For example, if a text field name is ClientName , this setting will automatically assign it the prompt of Client name .	(True) False
DefaultTitles	Generates default titles based on field names. Titles are used to describe the field when it appears as an icon in the HotDocs interview outline. For example, if a text field name is ClientName , this setting will automatically assign it the title of Client Name .	(True) False
DefaultResources	Generates resource text for markup fields based on the document text surrounding the field. This provides context for the question in the interview.	(True) False
IrrelDefault	Defines the default value (hidden, grayed, or always showing) for irrelevant variables in the interview. This option is available in HotDocs 2009 and later.	(Gray) hide
		show

The following is an example of how you can define a Settings table:

SETTINGS

Name	Value
StandardNames	True
DefaultPrompts	True
DefaultTitles	True
DefaultResources	False

You can also define these settings directly in the template. To do this, simply create a Settings field, like this: [Settings:StandardNames=true;DefaultPrompts=false].

Previous Topic Go to Section 4. Using the HotDocs Model in HotDocs

4. Using the HotDocs Model in HotDocs

◀ Previous Topic Next Topic ▶

Assemble a HotDocs Model

Once you have created a HotDocs Model, you can add the model to the HotDocs template library and assemble it using HotDocs.

To add the document to the library and assemble it

- 1. Start HotDocs.
- 2. At the template library, click the **Add Item** button. The **Add Item** dialog box appears.
- 3. Click the **Type** drop-down button and choose **HotDocs Model** from the list.
- 4. Click the Rowse button next to the File name field. The Add Item File Name dialog box appears.
- 5. Locate and select the HotDocs Model and click **OK**.
- 6. Enter a title in the **Title** field and click **OK**.
- 7. Once added, select the HotDocs Model in the item list and click 🏁 Assemble.
- 8. Complete the assembly process.

By default, when HotDocs generates an interview from a HotDocs Model and displays the questions, it shows the surrounding text in the resource pane of the assembly window. This can help provide context for the question being answered. (To view the resource pane, click the \heartsuit

Dialog Resource Pane button. (You can keep resources from appearing in your interview by defining a setting in the Settings table. See **Define Settings for a HotDocs Model** for details.)

When you add a HotDocs Model to the template library, HotDocs appends the **HotDocs Model** command-line option (**/mo**) to the file path. This indicates to HotDocs that this is a HotDocs Model and not just a regular Word document.

◀ Previous Topic Next Topic ▶

At a Glance: The Create HotDocs Model from Template Dialog Box

emplate file name:		
	Δ	
<u>1</u> odel file name:		
	B	
Add the new model to this library		

After clicking on **Create HotDocs Model from Template** in the **Template** menu of your HotDocs Library you will see the **Create HotDocs Model from Template** dialog box.

In field \boxed{A} you need to select the template you wish to export. The easiest way to do this is to click on the \boxed{A} **Browse** button to the right of field \boxed{A} and navigate to the template you want to export.

HotDocs will automatically fill field **B** with the same file name as your chosen template. If you would like to use a different file name for the HotDocs Model you can type it into field **B**.

If you check box C then HotDocs will put a reference to the newly created HotDocs Model into the HotDocs Library you have open. If you would not like to add the HotDocs Model to this library then make sure this box is unchecked.

Go here for more information on creating a HotDocs Model from a template.

Previous Topic Next Topic

Create HotDocs Model from Template

You can create a HotDocs Model from an existing HotDocs template. Saving templates as models allows you to share your templates with subject matter experts unfamiliar with HotDocs. As long as those experts understand (and use) the markup rules specified in this help file, they can further automate the template and use it with HotDocs User to assemble documents from it.

When you create a model from a template, HotDocs converts variable and instruction fields to markup fields. Variable and instruction properties are stored in tables at the end of the document. Component file properties are saved in an embedded component file.

Some template development features are not supported in HotDocs Models. This includes pointed component files and clause libraries.

To Create a HotDocs Model from a Template

- 1. At the template library, select the template you want to export.
- 2. Click Create Model from Template (Template menu). The Create HotDocs Model from Template dialog box appears.
- Optionally, change the name of the HotDocs Model in the HotDocs Model file name field. (By default, HotDocs appends the text .model to the file name. This helps distinguish between it and the original template file.)
- Optionally, select Add the new model to this library to add a reference to the model to the library. This allows you to more easily assemble the HotDocs Model using the Assemble command.
- 5. Click **OK**. HotDocs exports the template as a model. You can further edit the model or distribute copies of it to others for review or additional markup.

Previous Topic Next Topic

At a Glance: The Create HotDocs Template from Model Dialog Box

Create HotDocs Template from Model		? 🗙
Model file name:		
	Α	
Template file name:		
	B	
Add the new template to this library		
		OK Cancel

After clicking on **Create HotDocs Template from Model** in the **Template** menu of your HotDocs Library you will see the **Create HotDocs Template from Model** dialog box.

In field \triangle you need to select the HotDocs Model you wish to import. The easiest way to do this is to click on the \triangle **Browse** button to the right of field \triangle and navigate to the document you want to import.

HotDocs will automatically fill field \mathbf{B} with the same file name as your chosen HotDocs Model. If you would like to use a different file name for the template you can type it into field \mathbf{B} .

If you check box C then HotDocs will put a reference to the newly created template into the HotDocs Library you have open. If you would not like to add the template to this library then make sure this box is unchecked.

Go here for more information on creating a HotDocs template from a model.

Previous Topic Next Topic

Create HotDocs Template from Model

Once you have marked up a document, you can create a template from a HotDocs Model. Markup fields in the model are converted to variable and instruction fields. Once created, you can edit the template or assemble it.

To Create a HotDocs Template from a Model

- 1. At the HotDocs library, choose **Create Template from Model** (**Template** menu). The **Create HotDocs Template from Model** dialog box appears.
- 2. Click the Rowse button next to the HotDocs Model **file name** field and locate the HotDocs Model.

- Enter a file path and name for the newly created template in the HotDocs template file name field. (HotDocs automatically suggests the same name as the HotDocs Model, with the text .template appended to it.)
- 4. Optionally, select **Add the new template to this library**. This places a reference to the template in the HotDocs library.
- 5. Click **OK**. The template is created from the marked up HotDocs Model.

Previous Topic Next Topic

Previous Topic Go to Section 5. Including HotDocs Scripting In a HotDocs Model

HotDocs Model Naming Standards

When marking up HotDocs Models, you must adhere to the Markup rules, which requires you to follow certain standards when naming variables and other fields in the document. Failure to follow these standards, particularly #1, may result in errors or other problems when users attempt to assemble the HotDocs Model.

Specifically:

- 1. You must use only letters, digits, and underscores. You cannot use spaces or other special characters.
- 2. You should use names that are at least two words.
- 3. You should make one of the words in the name the "subject" word (Client, Child, etc.).
- 4. You should capitalize the first letter of each word, regardless of part of speech.
- 5. You must never use names that are all capitals.

If you must include spaces or special characters in your variable and other field names, you can select a HotDocs option that allows this. See Allow Non-Standard Names in HotDocs Markup.

Previous Topic Go to Section 5. Including HotDocs Scripting In a HotDocs Model

5. Including HotDocs Scripting In a HotDocs Model

Previous Topic Next Topic

Including HotDocs Scripting In a HotDocs Model

As you mark up your HotDocs Models, you may need create scripts that perform a particular task, such as insert one model into another or perform some action based on an answer the user provides. Similarly,

you may need to add up several dollar amounts, or find the number of years between two given dates. Or, you may need to search a user's answer for a certain string of text.

To accomplish this, you must use the HotDocs scripting language, which consists of instructions, expressions, operators, and values—such as text, numbers, dates, or answers users enter. You can include scripts in your markup to help you accomplish these different tasks.

Specifically, an instruction tells HotDocs to perform some sort of function, while an expression retrieves a special value. Most instructions and expressions also include placeholders, which you must replace with a value. Possible values include text strings, number amounts, other expressions, or variables. An instruction or expression will not work until all of its placeholders are replaced.

In addition to using instructions and expressions, you can use operators to control how a script is processed. Most operators are common mathematical signs, but there are also Boolean operators such as; AND and OR, as well as comparison operators.

When creating scripts, instruction and expression keywords can either be entered in the Script column of a Computation Variables table, or as an in-line property of a computation field. Keyword names are not case sensitive, but it is recommended you enter them in lowercase. Multi-word keywords are entered as one word, capitalizing the second (and any remaining) words in the expression or instruction. For example, to determine the absolute value of a number, you would enter the keyword as absoluteValue.

Template developers automating templates in HotDocs are required to use all uppercase letters when using scripting keywords. To accommodate this difference between template development and creating HotDocs Models, you can specify a setting in the HotDocs Model that allows for uppercase letters in expression and instruction keywords. See Define Settings for a HotDocs Model.

Previous Topic Next Topic

◀ Previous Topic Next Topic ▶

Full List of Instruction Models

Instruction	Description
ascend var; descend var	The ascend instruction sorts lists of answers (gathered using a repeat instruction) in alphanumeric order, from 1 to 9, and from A to Z. The descend instruction sorts lists of answers from 9 to 1, and from Z to A.
ask dialog	The ask dialog instruction allows you to control the order in which dialogs appear in an interview.
ask var	Sometimes a variable needs to be asked by itself. You can use the ask var instruction so that during the interview, HotDocs displays the variable in its own default dialog.

default var to value	This instruction suggests a value for a variable if the variable is unanswered.
filter computation_var	The filter instruction filters out certain entries from a repeated list, based on conditions you specify.
format "list_format"	The format "list_format" instruction allows you to create a sentence-style list within a computation.
if expression; elself; else; endlf	You can make sections of a HotDocs Model conditional by using if instructions. You can also make instructions or expressions in a script conditional. A conditional section will be included only if a condition you specify is true. The elself instruction allows two or more conditions to be included in an if instruction. The else instruction establishes a final condition for an if instruction, specifying that if all preceding conditions are false, the following information should be included. It must be the last item of the if instruction.
increment num_var; decrement num_var	The increment and decrement instructions cause HotDocs to increase or decrease a number variable, usually a counter, by the value of 1.
insert "filename"	This instruction inserts another HotDocs Model into the document currently being assembled. When HotDocs encounters an insert instruction, it immediately processes the instruction and inserts the model into the current document. If there are variables to be answered, HotDocs presents them before finishing the interview of the main document.
language code	This instruction tells HotDocs to format numbers and dates in a particular language.
repeat dialog; endRepeat	A repeat instruction gathers lists of answers and merges them into a document.
set var to value	This instruction lets you specify a given value for a variable's answer automatically, rather than allow the user to specify an answer. With the set instruction, you can transfer names and other values from one variable to another.
while expression; endWhile	The while expression instruction allows you to repeatedly process (or loop through) an answer or set of answers until a certain condition is met, such as a certain answer is found or a limit is reached.

Previous Topic Next Topic

Full List of Expression Models

Expression

Description

absoluteValue(num)	Using the absolute value expression, you can find the absolute value of a given number. You can calculate a negative number, but have it appear as a positive number.
age(date)	The age(date) expression produces an age, in years, by calculating the number of years between the current date (as determined by your computer's system clock) and a date you provide in the computation script.
answered (dialog)	HotDocs can determine whether a dialog has been answered using the answered expression. Even if only one variable in the dialog is answered, the expression returns a value of true.
answered (var)	You can use the answered expression to determine whether a HotDocs variable has been assigned a value. If so, the expression receives the value of true.
count(dialog)	You can find out how many sets of answers a user provides for a repeated dialog. A repeated dialog is any dialog used in a repeat instruction. This expression produces a number, based on each answered dialog.
count(mult_choice_var)	This expression counts how many options a user chooses when answering a Multiple Choice variable. The result it produces is a number.
counter	HotDocs uses counter as a way to compare two incrementing number values. For example, perhaps you want to list the last child named in a repeated dialog. You could create the following computation script:
date - num days	You can subtract any number of days from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
date - num months	You can subtract a certain number of months from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
date - num years	You can subtract a certain number of years from a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
date + num days	You can add any number of days to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
date + num months	You can add any number of days to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
date + num years	You can add a certain number of years to a Date variable. The result of this computation is a new date value, which can be merged into the assembled document.
dateOf(num, num, num)	This expression finds a date value based on day, month, and year values.
dayOf(date)	This expression returns the day portion (1 to 31) of a given date.

dayOfWeek(date)	This expression determines on which day of the week a specific date falls and converts that value to an integer.
daysFrom(date, date)	This expression allows you to find the number of days between two dates.
first(text, num)	Using this expression, you can return any number of characters starting with the first character in an answer value.
format(value, "example")	Sometimes you may need to add a date, number, or true/false value to a text value. You can do this by formatting the date, number, or true/false value as text.
integer(text)	Sometimes you may have a text value that contains number characters, as in the case of a time of day value. The integer expression allows you to convert those number characters into numeric values so you can perform calculations or compare them with other values.
last(text, num)	The last expression finds and returns a certain number of characters from the end of a text string.
length(text)	The length expression counts the number of characters—including spaces and punctuation—in a text value, such as a Text variable.
max(num, num)	The max expression compares two number values and returns the greater of the two.
mid(text, num, num)	Like the first and last expressions, this expression extracts a specified number of characters from within a text string.
min(num, num)	The MIN expression compares two number values and returns the lesser of the two.
monthOf(date)	This expression returns the month portion of a given date.
monthsFrom(date, date)	The months from expression calculates the number of months between two given dates.
mult_choice = text; mult_choice != text	The mult_choice = text expression returns true when the user chooses a Multiple Choice option that is equal to (=) a given text value. If it is not equal (!=), the expression returns false. The mult_choice != text expression functions in the opposite way—testing instead to see if an answer is not equal to (!=) a given text value.
not true_false	You can use the not true_false expression to find out if a True/False variable is false.
other(mult_choice_var)	This expression determines whether the user has chosen the Other option of a Multiple Choice variable and, if so, returns the text entered in the Other field. It can also be used to test whether the user has selected the None of the Above option.
position(text, text)	The position expression finds the position of a certain character or character string in a given text value. It is useful if you need to find a character you know will be in an answer but are not sure where it will appear. It returns a number value, which represents the first character.

power(num, num)	The power expression generates a numeric value, based on a given exponent.
remainder(num, num)	The remainder expression returns the remainder of a division. If the denominator is a zero, HotDocs generates a divide by zero error.
replace(text, text, text, num)	This expression lets you search a string of text for a given character string and replace the results with new text.
result	As you write computations, you often need HotDocs to acknowledge what the result would be at that point in the script. You can update this answer by using the result expression.
round(num, num)	You can round a number value to a specified number of places.
selection(mult_choice_var, num)	This expression lets you retrieve individual options (answers) selected in a Multiple Choice variable. It returns a text value that corresponds to the defined answer (as designated by the num placeholder).
space(text, text)	This expression tests whether the variable is answered. If it is, it merges the answer, followed by a space character. If the variable is unanswered, it merges nothing ("").
strip(text, text, true_false, true_false)	This expression removes a specified character or characters from the beginning or end of a text answer. By default, HotDocs removes the characters from both the beginning and the end of the text. If you want to specify just one or the other, you must use the true_false parameters.
sum(computation_var)	Using the sum expression, you can add computation values that have been repeated.
sum(num_var)	Using the sum expression, you can add repeated number values.
text contains text	The text contains text expression determines whether the first text value contains the same text as the second value. If it does, it returns the value of true.
today	This expression returns the current date, according to your computer's system clock.
truncate(num, num)	You can truncate a decimal number a specified number of places after a decimal point.
unanswered	This expression removes an assigned value from a variable. It is used most often with the set instruction:
union(mult_choice, mult_choice)	This expression creates a single list of all unique options (answers) that have been selected across two Multiple Choice variables.
value(var, expression)	This expression returns a default value for the variable type if the variable is unanswered. If the variable is answered, the value is the answer the user specifies.
yearOf(date)	You can use this expression model to find the year portion of a given date.

yearsFrom(date, This expression calculates the number of years between two given dates. date)

zero(num_var)

This expression returns the value of zero only if a Number variable is unanswered. If the Number variable is answered, the value is the answer the user specifies.

Use Operators When Scripting

An operator is a symbol or word that causes an operation such as addition or a comparison to be performed in a computation or expression. Most operators can be used when working with both number and text values.

There are three types of operators:

- **Comparison operators**: These compare two values of the same type (text, number, date, multiple choice, or true/false). They return values of true or false depending on whether the comparison is true or not.
- Arithmetic operators: These calculate new values. Operands used in the script must be the same type. You can use the Add (+) operator to string together (concatenate) two text values.
- **Logical operators**: These return a true/false value based on a logical comparison of their operands, both of which must be true or false.

The following tables explain how each operator works:

Comparison Operator	Description
=	The two items in the comparison are of equal value. For example:
	BirthDate = 17 Dec 1989
	Employee Name = "Louisa Gehrig"
!=	The two items in the comparison are not of equal value. For example:
	if ExhibitA != true
	if PlaintiffGender != "Male"
<	The first item in the comparison has a lesser value than the second item. For example:

	AccountBalance < 9000
	counter < 10
>	The first item in the comparison has a greater value than the second item. For example:
	DependentAge > 18
<=	The first item in the comparison is less than or equal to the second item. For example:
	ClientAge <= 65
	counter <= 2
>=	The first item in the comparison is greater than or equal to the second item. For example:
	TaxedIncome > = 75000
Arithmetic	Description
Operator	
Operator +	Add the different components of the script together. For example:
Operator +	Add the different components of the script together. For example: Value1 + Value2
Operator +	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState
Operator +	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState Subtract the different components of the script from each other. For example:
Operator + -	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState Subtract the different components of the script from each other. For example: MonthlyIncome - AmountOfOwedChildSupport
<pre>Operator + - *</pre>	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState Subtract the different components of the script from each other. For example: MonthlyIncome - AmountOfOwedChildSupport Multiply the different components of the script. For example:
<pre>Operator +</pre>	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState Subtract the different components of the script from each other. For example: MonthlyIncome - AmountOfOwedChildSupport Multiply the different components of the script. For example: PurchasePrice * 0.625
Operator + - *	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState Subtract the different components of the script from each other. For example: MonthlyIncome - AmountOfOwedChildSupport Multiply the different components of the script. For example: PurchasePrice * 0.625 Divide the different components of the script. For example:
<pre>Operator + /</pre>	Add the different components of the script together. For example: Value1 + Value2 ClientStreet + ", " + ClientCity + ", " + ClientState Subtract the different components of the script from each other. For example: MonthlyIncome - AmountOfOwedChildSupport Multiply the different components of the script. For example: PurchasePrice * 0.625 Divide the different components of the script. For example: YearlySalary / 12

AND	The statement to the left and the statement to the right must both be true. For example:
	if ClientIsMarried and ClientHasChildren
OR	The statement to the left or the statement to the right must be true. For example:
	if ClientIsSingle or ClientIsWidowed
ΝΟΤ	The two items in the comparison must not be equal to each other.

The final operator, the parentheses (), instructs HotDocs to perform the operation inside the parentheses first.

Example HotDocs Models

Simple Markup Example (Contract)

This example contains simple text, number, and date markup fields. Where an answer needs a special format, it has been marked as such. The example also contains a simple conditional statement. (To see a slightly more complex markup example, see Simple Markup Example (Agreement).)

If you're creating a text field, you do not need to include a variable property (te) in the field as HotDocs will automatically assume text as the default. With all other field types, however, you must define a field type property.

PUBLISHING CONTRACT

This Publishing Contract, by and between [EmployeeName], whose address is [EmployeeAddress], and Hobble Creek Publishing, is entered into on [AgreementDate;da].

Hobble Creek Publishing is in the business of publishing books and desires to publish [EmployeeName]'s novel entitled [NovelTitle]. [EmployeeName] desires to have [NovelTitle] published by Hobble Creek Publishing.

[EmployeeName] hereby grants to Hobble Creek Publishing the right to print, publish, distribute, and sell [NovelTitle] in the United States and Canada in various editions, including hardbound, trade paperbound, mass market paperbound, and special edition.

Hobble Creek Publishing agrees to publish [NumberOfCopies;nu] hardback copies of [NovelTitle] during its initial printing. Numbers for additional publications will be determined at a later date, based on the profitability of the initial printing. [EmployeeName] and Hobble Creek Publishing agree that Hobble Creek Publishing shall be under no obligation to publish [NovelTitle] provided that this Contract shall be terminated. All rights granted herein to Hobble Creek Publishing shall revert to [EmployeeName] if [NovelTitle;te] is not published by [PublishingDeadline;da].

[if:NoObligationToPublish:In addition, if Hobble Creek Publishing does not publish [NovelTitle] by [PublishingDeadline;da], Hobble Creek Publishing agrees to pay [EmployeeName] \$[KillFee;nu].

]

James Tyson

Hobble Creek Publishing

[EmployeeName]

Simple Markup Example (Agreement)

This markup example is slightly more complex than Simple Markup Example (Contract). It contains all of the variable types (text, number, date, true/false, multiple choice, and computation). All variables are fully marked using their field type.

Because this example doesn't incorporate variable tables, you'll note additional field properties for some variables, such as multiple choice options and merge text as well as a simple script for a computed field.

EMPLOYMENT AGREEMENT

This Employment Agreement, by and between Hobble Creek Publishing and [EmployeeName;te], is entered into this [AgreementDate;da;format="dth Day of Mn, YYYY"].

As of [HireDate;da], Hobble Creek Publishing employs [EmployeeName;te], and [EmployeeName;te] accepts employment, as a full-time [JobTitle;te]. Job duties shall include [JobDuties;te]. [EmployeeName;te] shall be paid \$[MonthlySalary;nu] per month, which is equivalent to \$[Yearly Salary;co;script=MonthlySalary * 12] per year. Salaries are paid monthly, on the last business day of the month.

[EmployeeName;te] shall be entitled to a paid annual vacation of [NumberOfVacationDays;nu;format=alpha] ([NumberOfVacationDays;nu]) days each year during the continuation of this agreement. Vacation time must be taken in the year earned. In addition to vacation time, [EmployeeName;te] may take the following paid holidays: New Year's Day, Martin Luther King, Jr., Day,

President's Day, Memorial Day, Independence Day, Labor Day, Thanksgiving (plus one additional), and Christmas (plus one additional).

[if:PaidSeminarDays:In addition, [EmployeeName;te] shall be allowed [NumberOfSeminarDays;nu] days each year to attend professional meetings or seminars, provided that [EmployeeGender;mc;options=male/female;merge=he/she] plans attendance at such meetings or seminars for minimum interference with the business of Hobble Creek Publishing.

][if:TrialPeriod:The length of [EmployeeName;te]'s employment will be an initial term of six months, with the possibility of continuation beyond that period depending on Hobble Creek Publishing's needs and upon [EmployeeGender;mc;options=male/female;merge=his/her] performance.

][EmployeeName;te]'s employment with Hobble Creek Publishing is "at will." The terms of employment are subject to change at Hobble Creek Publishing's discretion with advance written notice.

[CompanyRepresentative;te]

Hobble Creek Publishing

[EmployeeName;te]

Complex Markup Example with Tables (Last Will and Testament)

The following example shows a marked up HotDocs Model that incorporates variable tables.

Colors have been used to distinguish variable fields, If fields, and Repeat fields from the rest of the document text.

To see an example of a HotDocs Model that does not use variable tables, click here.

LAST WILL AND TESTAMENT

0F

[ClientName;format=upper]

I, [ClientName;format=upper], of [ClientCityOrCounty], Washington, being of sound and disposing mind, memory, and understanding, do hereby make, publish and declare this to be my Last Will and Testament (this "Will"), hereby revoking all prior wills, codicils, or other testamentary dispositions made by me.

[if:ClientIsMarriedOrHasChildren:

ITEM ONE: IDENTIFICATION

[if:ClientIsMarried:As used in this Will, the word "spouse" refers to [SpouseName;format=upper].][if:ClientHasChildren:I have [NumberOfChildren;format=alpha][ClientHasOneChild;merge=child/children] living on the date of execution of this will, [repeat:ChildList;format="a, b, and c":[ChildName;format=upper]]. As used herein, the terms "child" and "children" shall refer to any of my children named herein as such and any children subsequently born to or adopted by me. As used herein, the term "my descendants" shall include all children. A person who has a relationship by or through legal adoption shall take under this will as if the person had the relationship by or through birth.]]

ITEM TWO: FUNERAL PROVISION

I direct my Personal Representative to pay my funeral expenses, including the cost of a suitable marker for my grave, the cost of cremation by a reputable funeral director, and/or the expenses of any memorial service as my Personal Representative may deem appropriate, to be paid from the principal of my residuary estate, free of any limitation or restriction imposed by law with respect to the amount thereof and without the necessity of an order of court.

ITEM THREE: PAYMENT OF TAXES

I direct that all taxes due by me, all federal and state inheritance and estate taxes due and payable by reason of my death, and all expenses of administration of my estate, shall be payable out of the principal of the rest and residue of my estate, and no one shall be required or called upon by my Personal Representative to contribute to the payment of any such taxes.

ITEM FOUR: TANGIBLE PERSONAL PROPERTY

(A) I give and bequeath all automobiles, equipment and machinery, furniture, chinaware, silverware, household furnishings, books, pictures and other similar objects, all clothing, jewelry and all other tangible personal property which I may own at the time of my death in accordance with the Personal Property Memorandum. To the extent this property is not disposed of by such writing, this property shall pass as part of my Residuary Estate.

(B) I direct that any and all expenses related to the maintenance and storage of such tangible personal property after my death, and the transportation and delivery of such property to the beneficiary or beneficiaries, shall be borne by my estate.

ITEM FIVE: DISPOSITION OF RESIDUARY ESTATE

[if:ClientIsMarried: (A) Disposition to Spouse. After the payment of all expenses of administration and other charges payable from my estate, I hereby give, devise and bequeath all the rest, residue and remainder of my estate,
real, personal or mixed, wheresoever situated and howsoever acquired (my "Residuary Estate") unto my spouse, if my spouse survives me.

(B) Disposition to Children and Descendants. If my spouse predeceases me, then my Residuary Estate shall be divided into a sufficient number of equal shares, if more than one (1), so that there shall be set apart therefrom one (1) such share for each of my children as shall survive me, and one (1) such share for the descendants (as a group) of each child of mine who predeceases me, but of whom one (1) or more descendants shall survive me. Any share of my Residuary Estate to which the descendants (as a group) of a deceased child of mine shall be entitled as hereinabove provided shall be further divided into equal shares, so that there shall be set apart therefrom one (1) such share for each child of such deceased child of mine. Subject to ITEM SEVEN of this Will, each share of my Residuary Estate shall be distributed to my respective beneficiaries free of trust.

(C) Alternate Disposition. If none of my spouse, children, or other descendants survive me, then my Residuary Estate shall be distributed, free of trust, to [repeat:BeneficiaryList;format="a, b and c":[BeneficiaryName;format=upper]].

/else: (A) After the payment of all expenses of administration and other charges payable from my estate, I hereby give, devise and bequeath all the rest, residue and remainder of my estate, real, personal or mixed, wheresoever situated and howsoever acquired (my "Residuary Estate")

[if:ClientHasChildren:to my descendants who survive me. If none of my descendants survive me, I give the remainder of my real and personal estate (hereinafter referred to as my "Residuary Estate")

][if:BeneficiaryIsIndividual:[if:MoreThanOneBeneficiary:in equal shares]to
[repeat:BeneficiaryList;format="a, b and c":[BeneficiaryName;format=upper]].
But if

[if:OneBeneficiary:[BeneficiaryGender;merge=he/she]/elseif:TwoBeneficiaries:eit her /else:any such individual] predeceases me leaving descendants who survive me, [if:OneBeneficiary:my Residuary Estate/else:that individual's share] shall pass to such descendants[if:MoreThanOneBeneficiary:. If

[if:TwoBeneficiaries:either/else:any] such individual predeceases me leaving no descendants who survive me, then that individual's share shall pass [if:TwoBeneficiaries:to the other individual who survives me or, if the other individual predeceases me, then to the descendants who survive me, collectively, of that other individual/else:in equal shares to each such individual who survives me and to the descendants who survive me, collectively, of each such individual who predeceases me leaving descendants who survive me]]/else: to [BeneficiaryCharityName;format=upper], or its successor in interest].

(B) Whenever property is to be distributed to the descendants of a person (the "ancestor"), such property shall be divided into equal shares, one share for each then living descendant in the first generation below the ancestor in which at least one descendant is living, and one share for each deceased descendant in such generation who has a descendant then living. Each share created for a living descendant shall be distributed to such descendant. Each share created for a deceased descendant shall be divided and distributed according to the directions in the two preceding sentences until no property remains undistributed.

(C) A person who has a relationship by or through legal adoption shall take under this will as if the person had the relationship by or through birth.

ITEM SIX: FAILURE OF BENEFICIARIES

Any property of my estate maturing for ultimate and absolute distribution in respect to which there is no one then living and qualified to take under the foregoing provisions hereof shall be distributed to such person or persons as would have been entitled to receive my estate, and in the same proportions as they would have taken, had I died immediately following the time as of which there is no taker, intestate, unmarried, domiciled in the State of Washington, and the absolute owner of such share, portion of a share or other property then to be disposed of, as the case may be.

ITEM SEVEN: TRANSFERS TO MINORS

If any beneficiary entitled to a share of my estate under the foregoing provisions of this Will shall not have attained the age of twenty-one (21) years at the time for outright distribution of such share, then my Personal Representative shall distribute the share which such beneficiary would have been entitled to receive to such person (including any Personal Representative of mine) as my Personal Representative shall determine, to be held by such person as custodian for such beneficiary under the Washington Uniform Transfers to Minors Act until such beneficiary attains the age of twenty-one (21) years.

ITEM EIGHT: APPOINTMENT OF PERSONAL REPRESENTATIVE

(A) I hereby nominate and appoint [PersonalRepresentativeName;format=upper] to serve as Personal Representative of my estate under this, my Last Will and Testament.[if:AlternatePersonalRepresentativeNominated: If my above-named Personal Representative predeceases me, or survives me and is unable or unwilling for any reason to serve as my Personal Representative, then I hereby nominate and appoint [AlternatePersonalRepresentativeName;format=upper] to serve as my Personal Representative in his/her place and stead.]

(B) I direct that my Personal Representative herein named be excused from the necessity of giving bond.

(C) I confer upon my Personal Representative all powers necessary, proper or convenient for the preservation, management and direction of my estate, and I authorize my Personal Representative to buy, sell, assign, transfer, convey, exchange, divide, invest, reinvest, hypothecate, pledge, mortgage, borrow, lend (with or without security), lease, release, deed, grant options, compromise, arbitrate, consent to or oppose reorganizations, consolidations, mergers or partitions, select depositories for the funds of my estate and from time to time to deposit therein the funds of the estate, employ and pay counsel, and otherwise to deal with the whole or any portion of my estate, real and personal, as my said Personal Representatives may deem to be proper and advantageous to my estate, and to that end, to make contracts, deeds, conveyances, leases, releases, transfers and other instruments in writing, and to receive payment and to do all other acts and things incident thereto, all of which powers shall be exercised without the necessity of prior application to or subsequent ratification by any court.

(D) I further confer upon my Personal Representative all powers necessary, proper or convenient, without filing reports with any court, to continue, incorporate, enter into, or operate any business, whether as a stockholder, general or limited partner, sole or joint owner, or otherwise; to invest whatever assets may be needed in the business; to employ agents to operate the business; to serve in any capacity with the business; to receive reasonable compensation for such services, in addition to compensation for services as a fiduciary; and to reorganize, liquidate, merge, consolidate, or transfer the business or any part of it.

(E) It is my intention, and I hereby declare, that the mention of the above powers is not intended to be a limitation upon the exercise of other powers by my Personal Representative, but that my Personal Representative shall have all powers which my Personal Representative may deem to be necessary, proper or convenient for the advantageous administration of my estate and to carry out the purposes of this Will.

(F) My Personal Representative, while acting in good faith, shall not be liable or held responsible for any loss.

ITEM NINE: MISCELLANEOUS

(A) Governing Law. This Will shall be interpreted pursuant to the laws of the State of Washington, without regard to its conflict of laws principles.

(B) Headings. The item and paragraph headings are for convenience only, and shall have no bearing on the construction or interpretation of this Will.

(C) Severability. If any court of competent jurisdiction or any regulatory agency shall at any time invalidate any of the separate provisions of this Will, such invalidation shall not be construed as invalidating the whole of this Will, but only the separate provision or provisions in controversy. All of the remaining provisions shall be undisturbed as to their legal force and effect.

(D) Context. Gender, singular, or plural shall be interpreted as the context requires.

[if:GuardianNominated:

ITEM TEN: GUARDIAN FOR MINOR CHILDREN

In the event it becomes necessary for a guardian of person and property to be appointed for any child of mine, then I constitute and appoint [GuardianName;format=upper] to serve as guardian of the person and property of any child of mine. [if:AlternateGuardianNominated:Should [GuardianName;format=upper] fail or cease to serve, I name [AlternateGuardianName;format=upper] to be the guardian of the person and property of any child of mine.]I request that any guardian designated herein or pursuant to the provisions hereof be allowed to qualify as such without being required to give security on any required fiduciary bond. IN TESTIMONY WHEREOF, I have hereunto subscribed my name and affixed my seal
this [WillExecutionDate;format="dth day of Mn, y";unans="____ day of
______"].

_____(SEAL)

[ClientName;format=upper]

The foregoing instrument was SIGNED, SEALED, PUBLISHED and DECLARED by [ClientName;format=upper], as and for [ClientGender;merge=his/her] Last Will and Testament, in the presence of the undersigned, who, at [ClientGender;merge=his/her] request, in [ClientGender;merge=his/her] presence and in the presence of each other, hereunto subscribe our names as attesting witnesses.

Signature of First Witness

[WillWitness1Name]

Address:

]

[WillWitness1Address]

Signature of Second Witness

[WillWitness2Name]

Address:

[WillWitness2Address][enddocument]

TEXT VARIABLES

Name	Prompt	Resource	Additional
ClientName	Client full name		
ClientCityOrCounty	City or county of residence (e.g., Falls City, Emory County, Sanpete County)		

SpouseName	Spouse full name	irrel=hide
ChildName	Name of child	
PersonalRepresentativeName	Name of personal representative	
AlternatePersonalRepresentativeName	Name of alternate personal representative	
BeneficiaryName	Name of beneficiary	
BeneficiaryCharityName	Name of beneficiary	irrel=hide
GuardianName	Name of guardian	irrel=hide
AlternateGuardianName	Name of guardian	irrel=hide
WillWitness1Name	Name of first witness	warn=no
WillWitness1Address	Address	warn=no;height=4
WillWitness2Name	Name of second witness	warn=no
WillWitness2Address	Address	warn=no;height=4

DATE VARIABLES

Name	Prompt	Resource	Additional
WillExecutionDate	Date will is executed		warn=no

TRUE/FALSE VARIABLES

Name	Prompt	Resou rce	Additi onal
ClientIsMarried	Is the client married?		
ClientHasChildren	Does the client have children?		

GuardianNominated	Does the client want to nominate a guardian for minor children?
AlternateGuardianNominated	Does the client want to nominate an alternate guardian?
AlternatePersonalRepresentativ eNominated	Does the client want to nominate an alternate personal representa tive?

MULTIPLE CHOICE VARIABLES

Name	P r o m p t	R e s o u r c e	Options	Prompts	Ad di ti on al
Client Gender	G e n d e r o f C l i e n t		ClientIsMale/ClientI sFemale	Male/Fe male	

Benefi ciaryT ype	BeneficiaryIsIndivid ual/BeneficiaryIsCha rity	One or more individ uals/Ch arity	ir re l= hi d
Benefi ciaryG ender	BeneficiaryIsMale/Be neficiaryIsFemale	Male/Fe male	ir re l= hi d

COMPUTATION VARIABLES

Name	Script
ClientIsMarriedOrHasChildren	ClientIsMarried or ClientHasChildren
ClientIsMarriedAndHasChildren	ClientIsMarried and ClientHasChildren
ClientHasOneChild	<pre>count(ChildList) = 1</pre>
NumberOfChildren	<pre>count(ChildList)</pre>
BeneficiaryIsIndividual	BeneficiaryType = "BeneficiaryIsIndividual"
OneBeneficiary	count(BeneficiaryList) = 1
TwoBeneficiaries	count(BeneficiaryList) = 2
MoreThanOneBeneficiary	count(BeneficiaryList) = 1

DIALOGS

Name	Contents	Re so ur ce	Additio nal
ClientInformation	ClientName		
	ClientGender		
	ClientCityOrCou	inty	
	ClientIsMarried	I	

	SpouseName	
	ClientHasChildren	
ChildList	ChildName	style=s preadsh eet
PersonalRepresent ativeInformation	PersonalRepresentativeNa me	
	AlternatePersonalRe presentativeNominat ed	
	AlternatePersonalRe presentativeName	
GuardianInformati on	GuardianNominated	
	GuardianName	
	AlternateGuardianNo minated	
	AlternateGuardianNa me	
BeneficiaryInform ation	"Identify here the beneficiaries who are to receive the residue of the estate"	
	BeneficiaryType	
	BeneficiaryCharityN ame	
	BeneficiaryList	
	п п	
	BeneficiaryGender	
BeneficiaryList	BeneficiaryName	style=s sonpare nt;

irrel=h ide

INTERVIEW

ClientInformation ChildList BeneficiaryInformation PersonalRepresentativeInformation GuardianInformation ExecutionInformation

Quick Reference Guides

Quick Reference—Field Types

The following table describes the different types of fields you can mark in a HotDocs Model.

Field Type	Parts of the Field	Examples
Variable Field	Variable name	[ClientName;te;format=upper][ClientAge;nu;format=alpha][ClientBirthDate;da;format=dd Mn, YYYY]
	Variable type	
	Variable properties (optional)	
Conditional Text Field	If / Else If / Else keyword	[if:PaidSeminarDays:Document Text][if:ClientIsMarried:Document
	Variable type	text/elseif:ClientIsDivorced: Document text/else:Document text]
	Variable properties (optional)	
Repeated Text Field	Repeat instruction	[repeat:ChildrenInformation; format="a, b, and c":ascend=ChildAge: filter=Minors:Document text]
	Repeat field properties (optional)	
	Document text you want repeated	
	Repeat field properties (optional) Document text you want repeated	

Comment Field	Comment instruction	[comment:Calculates the monthly average]
	Comment text	
Ask Field	Ask instruction	[ask:EmployeeStartDate]
	Variable name	
Default Field	Default instruction	[default:HireDate="2 Nov 2008"]
	Variable name	
Set Field	Set instruction	[set:MaritalStatus=Single]
	Variable name	
Increment Field / Decrement	Increment or Decrement instruction	[increment:TempNum][decrement:TempNum]
Field	Number variable name	
Insert Field	Insert instruction	[insert:"healthcaredirective.docx"; keep=header]
	Model file name	
	Keep property with header or footer value (optional)	
Assemble Field	Assemble instruction	[assemble: "healthcaredirective.docx"]
	Model file name	
Language	Language Instruction	[language:fra; decimal=','; grouping='.']
riela	Language Code	
	• eng (for English)	
	 dea (for German) des (for Swiss German) 	

	 dea (for Austrian German) fra (for French) nld (for Dutch) esn (for Spanish) ita (for Italian) ptb (for Brazilian Portuguese) 	
	Decimal (optional)	
	Grouping (optional)	
Span Field	Span instruction	[span:ExclusionaryClause; title=Exclusions: Document Text]

To apply colors to the fields in your HotDocs Model, click the **P** Apply Color button on the HotDocs Markup Tools.

You can assign a comment property to any of the above-named fields.

Quick Reference—Field Properties

The following tables describe properties you can assign to the different field types in your HotDocs Models. For a full description of each property, see Define Field Properties. Default values are in parentheses.

Text Variable Properties

Property Name	Value	Where It Can Be Used
name	text	Variable field
title	text	Variable field
		Variable table
prompt	text	Variable field

		Variable table
resource	text	Variable field
		Variable table
format	(none)	Variable field
	upper	Variable table
	lower	
	title	
	sentence	
nonbreak	(no)	Variable field
	yes	Variable table
height	(0)	Variable field
	Number between 1 and 12	Variable table
	By default, HotDocs displays one-line answer fields	
max	(0)	Variable field
	Number between 1 and 15,000	Variable table
	By default, HotDocs lets you enter any number of characters between 1 and 15,000.	
pattern	(none)	Variable field
	(999) 999-9999	Variable table
	999-99-9999	
	99:99	
	99:99am	

enter	(break)	Variable field
	paragraph	Variable table
ask	(yes)	Variable field
	no	Variable table
warn	(yes)	Variable field
	no	Variable table
save	(yes)	Variable field
	no	Variable table
irrelevant	(gray)	Variable field
	hide	Variable table
	show	
unanswered	Any text you enter	Variable field
		Variable table
font	(font used in document)	Variable field
	Font Name	
comment	text	Variable field
		Variable table

Number Variables

Property Name	Value	Where It Can Be Used
name	text	Variable field
title	text	Variable field

		Variable table
prompt	text	Variable field
		Variable table
resource	text	Variable field
		Variable table
format	alpha	Variable field
	ordinal	Variable table
	09	
	9 1/8	
	9,999.00	
	9.9	
	9999	
	9th	
	IX	
	Nine Dollars and Twelve Cents	
nonbreak	(no)	Variable field
	yes	Variable table
minimum	(0)	Variable field
	any number	Variable table
maximum	(0)	Variable field
	any number	Variable table

decimal	(0)	Variable field
	any number between 1 and 7	Variable table
currency	(\$)	Variable field
	£	Variable table
	€	
	DM	
	Any currency (3 characters or less)	
ask	(yes)	Variable field
	no	Variable table
warn	(yes)	Variable field
	no	Variable table
save	(yes)	Variable field
	no	Variable table
irrelevant	(gray)	Variable field
	hide	Variable table
	show	
unanswered	Any text you enter	Variable field
		Variable table
font	(font used in document)	Variable field
	Font Name	
comment	text	Variable field

Variable table

Date Variables

Property Name	Value	Where It Can Be Used
name	text	Variable field
title	text	Variable field
		Variable table
prompt	text	Variable field
		Variable table
resource	text	Variable field
		Variable table
format	d	Variable field
	dd	Variable table
	dth	
	dy	
	m	
	mm	
	mn	
	mnt	
	у	
	уу	
	уууу	
	yr	

	wd	
	wdy	
nonbreak	(no)	Variable field
	yes	Variable table
ask	(yes)	Variable field
	no	Variable table
warn	(yes)	Variable field
	no	Variable table
save	(yes)	Variable field
	no	Variable table
irrelevant	(gray)	Variable field
	hide	Variable table
	show	
unanswered	Any text you enter	Variable field
		Variable table
font	(font used in document)	Variable field
	Font Name	
comment	text	Variable field
		Variable table

True/False Variables

HotDocs Models

Property Name	Value	Where It Can Be Used
name	text	Variable field
		Variable table
title	text	Variable field
prompt	text	Variable field
		Variable table
resource	text	Variable field
		Variable table
format	/x	Variable field
	true/false	Variable table
	x/	
	yes/no	
	truetext/falsetext	
style	(row)	Variable field
Style	(1011)	
	column	variable table
nonbreak	(no)	Variable field
	yes	Variable table
ask	(ves)	Variable field
	0	Variable table
	10	
warn	(yes)	Variable field

	no	Variable table
save	(yes)	Variable field
	no	Variable table
irrelevant	(gray)	Variable field
	hide	Variable table
	show	
unanswered	Any text you enter	Variable field
		Variable table
font	(font used in document)	Variable field
	Font Name	
comment	text	Variable field
		Variable table

Multiple Choice Variables

Property Name	Value	Where It Can Be Used
name	text	Variable field
title	text	Variable field
		Variable table
prompt	text	Variable field
		Variable table
resource	text	Variable field
		Variable table

format (multiple-select variables)	a, and b	Variable field
	a, b	Variable table
	a, b and c	
	a, b or c	
	a, b, and c	
	a; b; and c	
format (single-select variables)	upper	Variable field
	lower	Variable table
	title	
	sentence	
options	text	Variable field
		Variable table
optionprompts	text	Variable field
		Variable table
optionresources	text	Variable field
		Variable table
merge	text	Variable field
		Variable table

select	(single)	Variable field
	multiple	Variable table
other	(no)	Variable field
	yes	Variable table
none	(no)	Variable field
	yes	Variable table
style	(dropdown)	Variable field
	(grid)	Variable table
	column	
	list	
nonbreak	(no)	Variable field
	yes	Variable table
ask	(yes)	Variable field
	no	Variable table
warn	(yes)	Variable field
	no	Variable table
save	(yes)	Variable field
	no	Variable table
irrelevant	(gray)	Variable field
	hide	Variable table
	show	

HotDocs Models

unanswered	Any text you enter	Variable field
		Variable table
font	(font used in document)	Variable field
	Font Name	
comment	text	Variable field
		Variable table
comment	text	Variable field Variable table

Computation Variables

Property Name	Value	Where It Can Be Used
name	text	Variable field
script	A valid HotDocs script	Variable field
		Variable table
merge	text	Variable field
		Variable table
		This property can be used when the Computation field generates a true/false value.
nonbreak	(no)	Variable field
	yes	Variable table
unanswered	Any text you enter	Variable field
		Variable table
font	(font used in document)	Variable field
	Font Name	
comment	text	Variable field

Variable table

Dialogs

Property Name	Value	Where It Can Be Used
name	text	Dialog table
title	text	Dialog table
resource	text	Dialog table
contents	names of variables, dialogs, and/or dialog text variables	Dialog table
group	(none)	Dialog table
	single	
	multiple	
style	(regular)	Dialog table
	repeated	
	spreadsheet	
	ssonparent	
label	text	Dialog table
prompt	text	Dialog table
rows	(3)	Dialog table
	number	
ask	(yes)	Dialog table
	no	
irrelevant	(hide)	Dialog table
	show	

none	(no)	Dialog table
	yes	

Repeat Fields

Property Name	Value	Where It Can Be Used
name	text	Repeat field
		Dialog table
format	a, and b	Repeat field
	a, b	Dialog table
	a, b and c	
	a, b or c	
	a, b, and c	
	a; b; and c	
title	text	Repeat field
		Dialog table
resource	text	Repeat field
		Dialog table
contents	names of variables, dialogs, and/or dialog text	Repeat field
	variables	Dialog table

group	(none)	Repeat field
	single	Dialog table
	multiple	
style	(regular)	Dialog table
Style		Blaidy table
	repeated	
	spreadsheet	
	ssonparent	
label	text	Repeat field
		Dialog table
prompt	text	Repeat field
		Dialog table
rows	(3)	Repeat field
	number	Dialog table
ask	(yes)	Repeat field
	no	Dialog table
irrelevant	(hide)	Repeat field
	show	Dialog table
none	(no)	Repeat field
	yes	Dialog table
ascend	VariableName	Repeat field
ascenu		
		Dialog table

descend	VariableName	Repeat field
		Dialog table
filter	ComputationVariableName	Repeat field
		Dialog table
comment	text	Repeat field
		Dialog table

Insert Fields

Property Name	Value	Where It Can Be Used
keep	header	Insert field
	footer	
	both	
comment	text	Insert field

Span Fields

Property Name	Value	Where It Can Be Used
title	text	Span field
comment	text	Span field

Language Fields

Property Name	Value	Where It Can Be Used
decimal	decimal separator character	Language field
grouping	thousands separator character	Language field

Using Command Line Options

Introduction: Command-Line Options

Command-line options control the way HotDocs assembles text and form documents. You can place most options on any command line that causes HotDocs to run. They are case-sensitive and must be typed in lowercase letters. If the option requires you to include a full file path, you must enclose the file path with quotation marks (" ").

You can use command-line options in different ways, including specifying the options at the command line, at the **Properties** dialog box of a given template, within an ASSEMBLE instruction, and for the program file's shortcut.

Most command-line options control certain aspects of document assembly. For example, you can specify an option that always prints an assembled document once it has been sent to the word processor.

When you start HotDocs from the command line, it will continue to run, even after the processing of the command-line request is completed. To close HotDocs after it processes the command-line request, use the **Exit HotDocs** option.

When using multiple options on a single command line, separate each option with a space character. If the command line includes space characters, you must enclose the path in quotation marks (" ").

See Also

Full List of Command-line Options

Use Command-Line Options When Starting HotDocs

To use command-line options when starting HotDocs

- 1. Choose **Run** from the **Start** menu. The **Run** dialog box appears.
- 2. Type **"C:\Program Files\hotdocs.exe"** (including the quotation marks) followed by a space and the options you want in the **Open** field. For example:

"C:\Program Files\HotDocs\Hotdocs.exe" /tf=demoempl.docx

3. Click **OK**. If HotDocs is not already running, it loads and then performs the command-line instructions. If HotDocs is already running, it performs the command-line instructions.

Use Command-Line Options when Using a Shortcut to Start HotDocs

To use command-line options when using a shortcut to start HotDocs

- 1. Locate the HotDocs program file shortcut. (A shortcut is an icon on the desktop or **Start** menu that a user can click to quickly access a program.)
- 2. Right-click the icon and select **Properties** from the shortcut menu. The program item's **Properties** dialog box appears.
- 3. Select the **Shortcut** tab.
- 4. In the **Target** field, enter a space after the executable (.EXE) file and type the options you want.

Use Command-Line Options within ASSEMBLE Instructions

To use command-line options within ASSEMBLE instructions

- 1. Open the library you need at the HotDocs library window.
- Select the template with the ASSEMBLE instruction you wish to add a command-line option to, and click **Edit**.
- 3. Double click on the ASSEMBLE instruction.
- 4. The **Other Field** dialog box will open.
- 5. In the **Template to assemble** field, following the file name, type a space and the command-line options you want to use. For example:

Collection Letter.docx /sa

Use Command-Line Options at File Properties

To add command-line options to a library item

- 1. Open a library at the HotDocs library window.
- 2. Select the library item and click the **Properties** button to open the **Item Properties** dialog box.
- 3. In the **File name** field, following the file name, type a space and the command-line options you want to use. For example:

/af="C:\Documents and Settings\Username\My
Documents\HotDocs\Answers\jalvey.anx"

Full List of Command-Line Options

Below is a list of all the available command-line options grouped by use. For more information on any option follow the link to its full description.

Application Control Switches

These switches are used to feed instructions to the HotDocs executable from the command line.

Option		Description
Mutually Exclusive*	/tf=""	The Template File option causes HotDocs to assemble a document using the specified template or clause library.
	/ed=""	The Edit Template option causes HotDocs to edit a template using the specified template or clause library file.
	/ha=""	The HotDocs Auto-Assemble File option causes a document to be assembled using the specified auto-assemble (.HDA) file.
	/hi=""	The HotDocs Auto-Install File option causes the template set in the specified auto-install (.HDI) file to be installed. During this process, HotDocs prompts the user for any required information it needs to install the template library to the correct location.
/lf=""		The Library File option allows you to start HotDocs and open a specific library. If HotDocs is already running, it opens the library specified by the path and file name.

	/11	The Lock Library option locks the current library and prevents the user from editing the library or its contents. Specifically, when HotDocs is launched and the library appears, users can select templates and assemble documents from them. They can also view the answer library and change user preferences at the HotDocs Options dialog box. All other options are unavailable.
/ex		The Exit HotDocs option closes HotDocs when both of the following conditions are met: 1) there are no documents waiting to be assembled, and 2) all other programs are finished using HotDocs.
	/db	The Don't Brag option stops HotDocs from displaying the splash screen when it opens.
Mutually Exclusive*	/hl	The Hide Library option causes HotDocs to assemble a document without first displaying the HotDocs template library. The user will not see the template library window at all during assembly. It is most commonly used by integrators who are using HotDocs with a third-party program and want to start an assembly without first displaying the template library window.
	/sl	The Show Library option forces HotDocs to display the template library if it is currently not showing. This is useful if you are integrating HotDocs with another program and you have hidden the library using the Hide Library option.

Template Type Switches

When a template is referred to by HotDocs, the template type is inferred from the file name extension. However, in some cases additional information is required so HotDocs knows how to process the requested template. These **mutually exclusive** switches (usable in combination with the /tf="" switch, on library items, or in INSERT or ASSEMBLE instructions) help with that.

Option		Description
	/mo	The HotDocs Model command-line option indicates that the file referenced in the library is a HotDocs Model. When you select the document in the library and click Assemble , HotDocs will create an interview for the model.
Mutually Exclusive*	/cl="name"	The Clause Name option is used by HotDocs to identify which clause component is associated with an item in a clause library. It is also used by HotDocs to process INSERT instructions during the assembly process. Generally speaking, developers should never have to modify this option unless they are converting clauses from one file format to another. Likewise, end users may see the Clause Name option while working with clauses at a clause library or during assembly, but should not modify it.

Answer Initialization Switches

These switches are used to initialize & define the answer set that will be used for an interview or assembly. They can be used in combination with the /tf="" switch and on library items.

Option		Description
/df=""		The Default Answer File option specifies a default answer file that is used to "seed" any answer file created during assembly. When a new answer file is created, it is automatically loaded with answers from the default answer file.
Mutually Exclusive*	/af=""	The Answer File option is useful if you want to use a specific answer file when you assemble a document. The option does two things: 1) when a template is selected for assembly, it immediately opens the specified answer file without displaying the Answer File dialog box, and 2) it sets the value for path and file name as the current answer file name to be used when answers are saved. If the specified answer file doesn't exist, it will be created when the user saves the answers
	/na[=""]	The New Answer File option specifies a new, untitled answer file to be used when assembling a given document. This option causes HotDocs to suppress the Answer File dialog box, which normally appears before assembly. Specifying a path and file name is optional. If a file name is specified, it will be used for the new answer file. If an answer file with that same name already exists, HotDocs overwrites the existing file with the new one. If no file name is specified, HotDocs displays a Save Answer File dialog box at the end of assembly.
/ov=""		The Overlay Answer File option causes HotDocs to take answers from a specific answer file and overlay them in the current answer file. For example, if you have specific information about a client that can be used in assembling multiple documents, you can save just that information in an overlay answer file and then use the Overlay Answer File option to force HotDocs to use those answers when assembling a document. All answers entered during assembly (including overlaid answers) are saved to the current answer file—not the overlay answer file—thus maintaining the integrity of the overlay answer file. An overlay answer file is loaded after the regular answer file so that the answers contained therein can overlay existing answers.

Interview Behavior Switches

These switches are used to modify the default behavior of the assembly window. They can be used in combination with the /tf="" switch and on library items.

Option		Description
	/nw	The No Assembly Window option causes HotDocs to assemble a document without displaying the assembly window.
Mutually Exclusive*	/ni	The No Interview option removes the Interview tab from the assembly window, and,by default, displays the assembled document in the Document Preview or Form Document tab (depending on whether you are assembling a text or form document). To present a correctly assembled document, you should specify an answer file using the Answer File option. Otherwise, the document will be assembled without any answers.

/fia	When a user starts assembling a template that has the Finish Interview Action command-line option applied, HotDocs will complete the action defined in HotDocs Options—either display the assembled document at the Document tab of the assembly window or send the document to the word processor or HotDocs Filler. (See Control What Happens When You Finish an Interview.)
/la	The Lock Answer File option prevents users from opening, closing, and saving answer files during document assembly. If it is the only option used, however, users can choose an answer file before assembly and save any answers they have entered after assembly.
/sig	The Start Interview Group option is used to control which questions are asked when assembling a group of related documents, specifically, it keeps questions that are already answered in one interview from being asked in subsequent interviews. It must be used with the Keep Interview Group option, which must be assigned to each subsequent template within the group.
/kig	The Keep Interview Group option is used to control which questions are asked when assembling a group of related documents, specifically, it keeps questions that are already answered in one interview from being asked in subsequent interviews. It must be used with the Start Interview Group option, which must be assigned to each subsequent template within the group.
/is=u	The Interview Scope option allows you to ask only those dialogs that contain questions not answered by an existing answer file. This may be useful, for example, if you have some answers you are retrieving from a database that you don't want the user to change. Using this option will ask only those questions that don't have answers. Cannot be used with the Start Interview Group option or the Keep Interview Group option.
/sw	The Suppress Unanswered Warning option keeps HotDocs from displaying the warning dialog box that appears when the user attempts to either print, save, or send the assembled document to the word processor and the assembled document still contains unanswered questions.

Answer Disposition Switches

These **mutually exclusive** switches can be used to dictate what happens to answers that were modified while the assembly window was open or during assembly. They can be used in combination with the /tf="" switch and on library items.

Option		Description
Mutually Exclusive*	/sa	The Save Answers option forces an answer file to be saved at the end of an assembly. If using an existing answer file, any answers entered during the interview will be saved automatically. If using a new, untitled answer file, HotDocs will force the user to specify an answer file name.
	/sap	The Save Answers Prompt option, which is used in connection with an ASSEMBLE instruction, prompts the user to save an answer file after completing an interview. Regardless of whether the user uses an existing answer file during assembly, when

		the user finishes that assembly, HotDocs prompts to save the answers in a different file.
	/ss	The Suggest Save option, which is used in connection with an ASSEMBLE instruction, causes HotDocs to ask users after assembly of a document has finished if they want to save answers entered during the interview in an answer file. Specifically, if the user has assembled a document and made changes to an existing answer file, HotDocs prompts to save the answers to that file. If saving a new, untitled file, HotDocs allows the user to specify the new answer file name.
	/ssn	The Suggest Save New option, which is used in connection with an ASSEMBLE instruction, causes HotDocs to ask if answers should be saved in a new answer file after assembly of a document has finished. Regardless of whether the user is using an existing answer file during assembly, when the user finishes that assembly, HotDocs gives the user the option of saving the answers in a new answer file.
	/da	The Discard Answers option prevents the user from saving answers after the document has been assembled. This option is useful when you know you will never want to save the answers you use with a particular template (for example, a fax cover sheet), and you don't want HotDocs to ask about saving the answers when you close the assembly window. However, the user can save the answer file during the interview.

Document Disposition Switches

These switches can be used to dictate what happens to a document after its assembly is complete. They can be used in combination with the /tf="" switch and on library items.

Option	Description
/of=""	The Output File option causes HotDocs to assemble the document and save it using the file name specified. If you are using the Answer Summary or Question Summary options, the Output File option specifies the name for either of those generated documents. This is useful if you know you want to save an assembled document every time assembly of that document finishes.
/as	The Answer Summary option is used with the Output File option to specify the path and file name for saving an answer summary. It is useful if you want a certain template to always generate an answer summary document.
/qs	The Question Summary option is used with the Output File option to specify the path and file name for saving a question summary. It is useful if you want a certain template to always generate a question summary document.
/sto	The Send to Plugin option sends the assembled document to a specified output plugin.
/stw	The Send to Word Processor option sends the assembled document to the word processor once the user closes the assembly window. This is useful if you know you always want to view the assembled document using the word processing program.

Automator/Filler Command Line Switches

These switches can be used to dictate how HotDocs will print form templates from HotDocs Automator/Filler.

C	Option	Description
	/pr	The Print option causes HotDocs to print a copy of the assembled text or form document once the user closes the assembly window. This is useful if you know you will always need to print a copy of a specific assembled document.
/pw		The Print Without Dialogs option causes HotDocs to bypass the Print dialog box and print the form using the current printer. The form is printed when the user clicks the Print Document button at the assembly window.
/ps=""		The Paper Size option selects the specified paper size when the user prints a copy of the form template or document. The effect is the same as manually setting the page size from the Print dialog box. This option works with form templates and documents only.
/pt=""		The Paper Tray option causes a specified printer paper tray or manual feed option to be used when printing a form document from HotDocs Filler. Paper tray values that can be used include manual , upper , lower , and so forth. For a complete list of acceptable values, either at the assembly window or at the HotDocs Filler window, click Document Properties > Printing (File menu) and click the Paper Source drop-down button.
/pc=n		The Print Copies option specifies the number of copies that should be printed when the user prints the form document (type in the number of copies needed instead of the letter n). This number should appear in the Number of Copies field at the Print dialog box.
/pd		The Print Duplex option sets the duplex printing option for a given form document. It prints the document Double-Sided , Side-to-Side , as if that option were selected at the Printing Properties dialog box (which you can access by clicking Document Properties > Printing (File menu).) When the user prints the form document, it is printed using this option.
Mutually Exclusive*	/pa	The Print Answers Only option selects the Answers Only (Use Preprinted Form) option at the Print dialog box. Then, when the user prints the assembled form document, it prints only the form's answers and not the underlying static text. This allows you to use preprinted forms.
	/ро	The Print Form Only option selects the Form Only (Blank Form) option at the Print dialog box. Then, when the user prints the form document, it prints a blank copy of the form without answers.
	/pb	The Print Both option selects the Form with Answers option at the Print dialog box. Then, when the user prints the form document, the current form and its answers are printed.

*Mutually exclusive switches will not show an error if used in the same command-line but will cause unpredictable behavior. We recommend you do not use these switches together.

Answer File

/af="path and file name"

The **Answer File** option is useful if you want to use a specific answer file when you assemble a document. The option does two things: 1) when a template is selected for assembly, it immediately opens the specified answer file without displaying the **Answer File** dialog box, and 2) it sets the value for path and file name as the current answer file name to be used when answers are saved. If the specified answer file doesn't exist, it will be created when the user saves the answers.

When using an existing answer file, you can retrieve an answer file from a location on a Web server by specifying a URL for the path and file name (for example, */af=http://www.yoursite.com/answers.anx*). (You cannot, however, save an answer file back to the server.)

You cannot assign the Answer File (/af) option to a HotDocs Auto-Assemble file.

If using this option at the command line, include the **Template File** (/tf) option.

Answer Summary

/as

The **Answer Summary** option is used with the **Output File** option to specify the path and file name for saving an answer summary. It is useful if you want a certain template to always generate an answer summary document.

The **Answer Summary** option is normally used with the **No Assembly Window** and **Answer File** options, which cause HotDocs to automatically create and save the answer summary document without displaying the assembly window.

Answer summaries are saved in HTML format.

If using this option at the command line, include the **Template File** (/tf) and **Output File** (/of) options. If using this option at the library properties, include the **Output File** (/of) option.

Clause Name

/cl=clausename

The **Clause Name** option is used by HotDocs to identify which clause component is associated with an item in a clause library. It is also used by HotDocs to process INSERT instructions during the assembly process. Generally speaking, developers should never have to modify this option unless they are converting clauses from one file format to another. Likewise, end users may see the **Clause Name** option while working with clauses at a clause library or during assembly, but should not modify it.

Default Answer File

/df="path and file name"

The **Default Answer File** option specifies a default answer file that is used to "seed" any answer file created during assembly. When a new answer file is created, it is automatically loaded with answers from the default answer file.

When specified, it does not need to have the same file name as the template's component file, nor does it need to be saved in the same folder as the component file. However, the default answer file name should be different from the current answer file name. Also, when using an existing default answer file, you can retrieve it from a location on a Web server by specifying a URL for the path and file name (for example, //df=http://www.yoursite.com/defaultanswers.anx).

If using this option at the command line, include the **Template File** (/tf) option.

Discard Answers

/da

The **Discard Answers** option prevents the user from saving answers after the document has been assembled. This option is useful when you know you will never want to save the answers you use with a particular template (for example, a fax cover sheet), and you don't want HotDocs to ask about saving the answers when you close the assembly window. However, the user can save the answer file during the interview.
To disable all answer file usage (saving, selecting new, and so forth) during a given assembly, use the **Lock Answer File** (/la) option.

If using this option at the command line, include the **Template File** (/tf) option.

Don't Brag

/db

The **Don't Brag** option stops HotDocs from displaying the splash screen when it opens.

Edit Template

/ed="path and file name"

The **Edit Template** option causes HotDocs to edit a template using the specified template or clause library file.

Exit HotDocs

/ex

The **Exit HotDocs** option closes HotDocs when both of the following conditions are met: 1) there are no documents waiting to be assembled, and 2) all other programs are finished using HotDocs.

Finish Interview Action

/fia

When a user starts assembling a template that has the **Finish Interview Action** command-line option applied, HotDocs will complete the action defined in HotDocs Options—either display the assembled document at the **Document** tab of the assembly window or send the document to the word processor or HotDocs Filler. (See Control What Happens When You Finish an Interview.)

If using this option at the command line, include the **Template File** (/tf) and **Answer File** (/af) options. If using this option at the library properties, include the **Answer File** (/af) option.

Hide Library

/hl

The **Hide Library** option causes HotDocs to assemble a document without first displaying the HotDocs template library. The user will not see the template library window at all during assembly. It is most commonly used by integrators who are using HotDocs with a third-party program and want to start an assembly without first displaying the template library window. (See also Show Library.)

If using this option at the command line, include the **Template File** (/tf) option.

HotDocs Auto-Assemble File

/ha="path and file name"

The **HotDocs Auto-Assemble File** option causes a document to be assembled using the specified autoassemble (.HDA) file.

You can also specify a URL for the path and file name (for example, */ha=http://www.yoursite.com/hdafile.hda*).

HotDocs Auto-Install File

/hi="path and file name"

The **HotDocs Auto-Install File** option causes the template set in the specified auto-install (.HDI) file to be installed. During this process, HotDocs prompts the user for any required information it needs to install the template library to the correct location.

You can also specify a URL for the path and file name (for example,

/hi=http://www.yoursite.com/hdifile.hdi). When this command is passed, HotDocs downloads the file and prompts the user for the information needed to install the template library.

HotDocs Model

/mo

The **HotDocs Model** command-line option indicates that the file referenced in the library is a HotDocs Model. When you select the document in the library and click **Assemble**, HotDocs will create an interview for the model.

See Introduction: Create HotDocs Models for more information.

Interview Scope

/is=u

The **Interview Scope** option allows you to ask only those dialogs that contain questions not answered by an existing answer file. This may be useful, for example, if you have some answers you are retrieving from a database that you don't want the user to change. Using this option will ask only those questions that don't have answers.

If you do not want certain variables to appear in an interview, do not include them in an explicit ASK instruction. Otherwise, HotDocs will present the variables to the user.

Cannot be used with the **Start Interview Group** (/sig) option or the **Keep Interview Group** (/kig) option.

Keep Interview Group

/kig

The **Keep Interview Group** option is used to control which questions are asked when assembling a group of related documents, specifically, it keeps questions that are already answered in one interview from being asked in subsequent interviews. It must be used with the **Start Interview Group** option, which must be assigned to each subsequent template within the group.

For example, you have three related templates that will be added to the assembly queue (*Template A*, *Template B*, and *Template C*). Each of these templates uses *Variable A*. To keep *Variable A* from being asked in all three interviews, you would assign the **Start Interview Group** option to *Template A*. Then you would assign the **Keep Interview Group** option to *Templates B* and *C*. Once the user answers *Variable A*, it will not be asked in any subsequent interviews.

If a template is added to the assembly queue that doesn't use either of these options, it and any subsequent templates will not be included in the interview group.

Library File

/lf="path and file name"

The **Library File** option allows you to start HotDocs and open a specific library. If HotDocs is already running, it opens the library specified by the path and file name.

Lock Answer File

/la

The **Lock Answer File** option prevents users from opening, closing, and saving answer files during document assembly. If it is the only option used, however, users can choose an answer file before assembly and save any answers they have entered after assembly.

To keep users from choosing an answer file before assembly, specify an answer file using the **Answer File** option. To keep them from manually saving their answers, use either the **Discard Answers** option or the **Save Answers** option.

If using this option at the command line, include the **Template File** (/tf) option.

Does not work in conjunction with the No Assembly Window (/nw) option.

Lock Library

/11

The **Lock Library** option locks the current library and prevents the user from editing the library or its contents. Specifically, when HotDocs is launched and the library appears, users can select templates and assemble documents from them. They can also view the answer library and change user preferences at the **HotDocs Options** dialog box. All other options are unavailable.

New Answer File

/na[="path and file name"]

The **New Answer File** option specifies a new, untitled answer file to be used when assembling a given document. This option causes HotDocs to suppress the **Answer File** dialog box, which normally appears before assembly. Specifying a path and file name is optional. If a file name is specified, it will be used for the new answer file. If an answer file with that same name already exists, HotDocs overwrites the existing file with the new one. If no file name is specified, HotDocs displays a **Save Answer File** dialog box at the end of assembly.

When specifying an answer file name, you must include the file name extension .ANX.

If using this option at the command line, include the **Template File** (/tf) option.

No Assembly Window

/nw

The **No Assembly Window** option causes HotDocs to assemble a document without displaying the assembly window.

If using this option at the command line, include the **Template File** (/tf) and **Answer File** (/af) options. If using this option at the library properties, include the **Answer File** (/af) option.

You should also use the **Output File** (**/of**) or **Send to Word Processor** (**/stw**) option if you want HotDocs to produce a document at the end of the assembly.

No Interview

/ni

The **No Interview** option removes the **Interview** tab from the assembly window, and, by default, displays the assembled document in the **Document Preview** or **Form Document** tab (depending on whether you are assembling a text or form document). To present a correctly assembled document, you should specify an answer file using the **Answer File** option. Otherwise, the document will be assembled without any answers.

While viewing an assembled document that was generated using this command-line option, you cannot edit answers while viewing the **Document** tab.

If using this option at the command line, include the **Template File** (/tf) and **Answer File** (/af) options. If using this option at the library properties, include the **Answer File** (/af) option.

Output File

/of="path and file name"

The **Output File** option causes HotDocs to assemble the document and save it using the file name you specify. If you are using the **Answer Summary** or **Question Summary** options, the **Output File** option specifies the name for either of those generated documents. This is useful if you know you want to save an assembled document every time that HotDocs finishes assembling that document.

Changing the file extension on the file specified in the Output File option instructs HotDocs to convert the output file to the new file type. For example, adding the command line option /of="C:\templatename.docx" to an RTF template results in a DOCX format output file.

Since HotDocs cannot convert to all format types, we recommend that you manually test HotDocs conversion capablities for a desired format before adding the command-line option. Be aware as well that converting output files to alternate formats can sometimes cause variations in formatting and user experience.

If you use this option with the command line, be sure to include the **Template File (/tf)** option.

Limitations When Using /of with /nw

Normally, the file extension on the file specified in the /of command line switch instructs HotDocs to convert the assembled document to a file of that type after assembly. However, when converting to PDF, this only works when the assembly window is visible. When you hide the assembly window (by using the /nw switch), conversion to a PDF document is not supported. HotDocs does, however, save the assembled document to a file with a .PDF extension as instructed by the /of command line switch. This can be confusing if you specify "tf=template.docx /of=document.pdf" in the command line after using a /nw switch. In this case, HotDocs assembles a DOCX document from the DOCX template, but saves the DOCX document iin the file system with the filename specified by the /of switch (document.pdf). If you then double-click that file in Windows Explorer, the application mapped to the PDF file extension (usually Adobe Reader) tries to open the file but fails because the file contains a DOCX document, not a PDF document (as the file extension claims).

In cases where the assembly window is hidden by specifying the /nw switch, only native word processor file types are supported for the /of commandline switch, as follows:

Template type	Allowed output file
DOCX	DOCX
RTF	RTF, DOCX
DOT	RTF, DOCX

Work Around

A possible work around is to use Microsoft Word's PDF conversion feature to accomplish the same result, using either Word's graphical user or programming interface.

Overlay Answer File

/ov="path and file name"

The **Overlay Answer File** option causes HotDocs to take answers from a specific answer file and overlay them in the current answer file. For example, if you have specific information about a client that can be used in assembling multiple documents, you can save just that information in an overlay answer file and then use the **Overlay Answer File** option to force HotDocs to use those answers when assembling a document. All answers entered during assembly (including overlaid answers) are saved to the current answer file—not the overlay answer file—thus maintaining the integrity of the overlay answer file. An overlay answer file is loaded after the regular answer file so that the answers contained therein can overlay existing answers.

If you do not include a full path on the command-line, HotDocs will first look for the answer file in the same folder as the template. If it's not located there, HotDocs will look in the *Answers* folder.

You can retrieve an overlay answer file from a location on a Web server by specifying a URL for the path and file name (for example, */ov=http://www.yoursite.com/overlayanswers.anx*).

If you are saving the overlay answer file to the same folder as the template, do not use the same name as the template. Otherwise, HotDocs will think the answer file is a default answer file. (See Create a Default Answer File.)

If using this option at the command line, include the **Template File** (/tf) option.

Paper Size

/ps=paper size

The **Paper Size** option selects the specified paper size when the user prints a copy of the form template or document. The effect is the same as manually setting the page size from the **Print** dialog box. This option works with form templates and documents only.

Paper size values that can be used include **letter**, **legal**, and so forth. For a complete list of acceptable values, either at the assembly window or at the HotDocs Filler window, click **Document Properties** > **Printing** (**File** menu) and click the **Paper Size** drop-down button. Values that include a space character must be placed inside quotation marks. You can shorten the values as long as the shortened form matches only one paper size. Paper size values are not case-sensitive.

If a paper size is specified at the **Printing Properties** dialog box and the **Paper Size** (/ps) command-line option is also used, the command-line option takes precedence.

If using this option at the command line, include the **Template File** (/tf) option

Paper Tray

/pt=paper tray

This option is used with form documents only.

The **Paper Tray** option causes a specified printer paper tray or manual feed option to be used when printing a form document from HotDocs Filler. Paper tray values that can be used include **manual**, **upper**, **lower**, and so forth. For a complete list of acceptable values, either at the assembly window or at the HotDocs Filler window, click **Document Properties > Printing** (**File** menu) and click the **Paper Source** drop-down button.

If a paper source is specified at the **Printing Properties** dialog box and the **Paper Tray** (/pt) command-line option is also used, the command-line option takes precedence.

If using this option at the command line, include the **Template File** (/tf) option.

Print

/pr

This option is used with form documents only

The **Print** option causes HotDocs to print a copy of the assembled text or form document once the user closes the assembly window. This is useful if you know you will always need to print a copy of a specific assembled document.

If using this option at the command line, include the **Template File** (/tf) option.

Print Answers Only

/pa

This option is used with form documents only

The **Print Answers Only** option selects the **Answers Only (Use Preprinted Form)** option at the **Print** dialog box. Then, when the user prints the assembled form document, it prints only the form's answers and not the underlying static text. This allows you to use preprinted forms.

If using this option at the command line, include the **Template File** (/tf) option.

Print Both

/pb

This option is used with form documents only

The **Print Both** option selects the **Form with Answers** option at the **Print** dialog box. Then, when the user prints the form document, the current form and its answers are printed.

If using this option at the command line, include the **Template File** (/tf) option.

Print Copies

/pc=numberofcopies

This option is used with form documents only

The **Print Copies** option specifies the number of copies that should be printed when the user prints the form document. This number should appear in the **Number of Copies** field at the **Print** dialog box

If using this option at the command line, include the **Template File** (/tf) option.

Print Duplex

/pd

This option is used with form documents only

The **Print Duplex** option sets the duplex printing option for a given form document. It prints the document **Double-Sided**, **Side-to-Side**, as if that option were selected at the **Printing Properties** dialog box (which you can access by clicking **Document Properties > Printing** (**File** menu).) When the user prints the form document, it is printed using this option.

If using this option at the command line, include the **Template File** (/tf) option.

Print Form Only

/po

This option is used with form documents only

The **Print Form Only** option selects the **Form Only (Blank Form)** option at the **Print** dialog box. Then, when the user prints the form document, it prints a blank copy of the form without answers.

If using this option at the command line, include the **Template File** (/tf) option.

Print Without Dialogs

/pw

This option is used with form documents only

The **Print Without Dialog** option causes HotDocs to bypass the **Print** dialog box and print the form using the current printer. The form is printed when the user clicks the **Print Document** button at the assembly window.

If using this option at the command line, include the **Template File** (/tf) option.

Question Summary

/qs

The **Question Summary** option is used with the **Output File** option to specify the path and file name for saving a question summary. It is useful if you want a certain template to always generate a question summary document.

The **Question Summary** option is normally used with the **No Assembly Window** option, which causes HotDocs to automatically create and save the question summary document without displaying the assembly window.

Question summaries are saved in HTML format.

If using this option at the command line, include the **Template File** (/**tf**) option.

Save Answers

/sa

The **Save Answers** option forces an answer file to be saved at the end of an assembly. If using an existing answer file, any answers entered during the interview will be saved automatically. If using a new, untitled answer file, HotDocs will force the user to specify an answer file name.

To always force the user to save a new answer file after entering answers in an interview—even if using an existing answer file—use the **Save Answers Prompt** (/sap) option.

To give users the option of saving an answer file, rather than forcing them to save, use either the **Suggest Save (/ss)** or the **Suggest Save New (/ssn)** option.

Save Answers Prompt

/sap

The **Save Answers Prompt** option, which is used in connection with an ASSEMBLE instruction, prompts the user to save an answer file after completing an interview. Regardless of whether the user uses an existing answer file during assembly, when the user finishes that assembly, HotDocs prompts to save the answers in a different file.

To always save an answer file without prompting the user for an answer file name (unless the user is using a new answer file) use the **Save Answers** (/sa) option.

To give users the option of saving an answer file, rather than forcing them to save, use either the **Suggest Save** (/ss) or the **Suggest Save** New (/ssn) option.

Send to Plugin

/sto="Name of Plugin"

The **Send to Plugin** option sends the assembled document to a specified output plugin. The class name of the plugin is the plugin name. For example, to use a google upload plugin the command line switch would be:

/sto="HDGoogleDriveOutputPlugin"

If using this option at the command line, include the **Template File** (/tf) option

Send to Word Processor

/stw

The **Send to Word Processor** option sends the assembled document to the word processor once the user closes the assembly window. This is useful if you know you always want to view the assembled document using the word processing program.

If using this option at the command line, include the **Template File** (/**tf**) option.

Show Library

/sl

The **Show Library** option forces HotDocs to display the template library if it is currently not showing. This is useful if you are integrating HotDocs with another program and you have hidden the library using the **Hide Library** option.

Start Interview Group

/sig

The **Start Interview Group** option is used to control which questions are asked when assembling a group of related documents, specifically, it keeps questions that are already answered in one interview from being asked in subsequent interviews. It must be used with the **Keep Interview Group** option, which must be assigned to each subsequent template within the group.

For example, you have three related templates that will be added to the assembly queue (*Template A*, *Template B*, and *Template C*). Each of these templates uses *Variable A*. To keep *Variable A* from being asked in all three interviews, you would assign the **Start Interview Group** option to *Template A*. Then you would assign the **Keep Interview Group** option to *Templates B* and *C*. Once the user answers *Variable A*, it will not be asked in any subsequent interviews.

If a template is added to the assembly queue that doesn't use either of these options, it and any subsequent templates will not be included in the interview group.

Suggest Save

/ss

The **Suggest Save** option, which is used in connection with an ASSEMBLE instruction, causes HotDocs to ask users after assembly of a document has finished if they want to save answers entered during the interview in an answer file. Specifically, if the user has assembled a document and made changes to an existing answer file, HotDocs prompts to save the answers to that file. If saving a new, untitled file, HotDocs allows the user to specify the new answer file name.

If your user is using an existing answer file but you want to give the user the option of saving the answers in a new, different answer file, use the **Suggest Save New** (/ssn) option.

If you want to force users to save their answers after an assembly is finished, rather than give them the option, use either the **Save Answers (/sa)** or the **Save Answers Prompt (/sap)** options.

Suggest Save New

/ssn

The **Suggest Save New** option, which is used in connection with an ASSEMBLE instruction, causes HotDocs to ask if answers should be saved in a new answer file after assembly of a document has finished. Regardless of whether the user is using an existing answer file during assembly, when the user finishes that assembly, HotDocs gives the user the option of saving the answers in a new answer file.

If your users are using an existing answer file and you want them to save answers they have entered in that file instead of a new one, use the **Suggest Save** (/ss) option.

If you want to force users to save their answers after an assembly is finished, rather than give them the option, use either the **Save Answers (/sa)** or the **Save Answers Prompt (/sap)** options.

Suppress Unanswered Warning

/sw

The **Suppress Unanswered Warning** option keeps HotDocs from displaying the warning dialog box that appears when the user attempts to either print, save, or send the assembled document to the word processor and the assembled document still contains unanswered questions.

Template File

/tf="path and file name"

The **Template File** option causes HotDocs to assemble a document using the specified template or clause library.

If you want an interview template (component file) started from the **Template File** command line, the component file must have an INTERVIEW or STARTUP computation in it or assembly will fail.

Troubleshooting

Template Development

Converting to HotDocs 11.1 Produces Component Name is Invalid Error

Problem

I'm trying to convert a pre-existing template to HotDocs 11 using the Component Manager. I get a message saying that it will be converted to 11 format, so I click **OK**.

I get a message that states "ERROR: The component name "my component name" is invalid. A component name cannot contain..."

Solution

You need to revert to 11.0.x or earlier to address the issue. The issue is caused by using an invalid variable name. This can include using a reserved key word such as a predefined language token, or by using all UPPERCASE letters in your variable names.

Once you have reverted to 11.0.x or earlier and follow the tips on naming your variables. You should then be able to convert to 11.1 without problems.

Can't Dynamically Add Merge Text to a Multiple Choice Variable

Problem

I'm creating my Multiple Choice variables "on the fly," using the CLEAR and ADD instructions. I know I can ADD options and prompts, but can I ADD Merge Text to the variable?

Solution

Unlike options and prompts, the merge text for a Multiple Choice variable comes from another component—a Merge Text component. This means you cannot create and add merge text to a Multiple Choice variable "on the fly."

Changing the Order Questions are Asked in an Interview

Problem

I have completed automation on my template but when I test assemble it, I find that I want the questions asked in a different order than how HotDocs is asking them. Is there some way to override the order dialogs are asked?

Solution

By default, HotDocs generates this interview by asking variables as it reads them in the template. When a variable is linked to a dialog, the dialog is asked. For most templates, this default interview is sufficient. However, there may be times when you want to control the order questions are asked.

There are two ways to accomplish this: 1) Use an ASK instruction to ask a single dialog at a specific place in the interview, and 2) use an INTERVIEW computation to define the order of all dialogs in template.

For details, see Control When Your Dialogs Appear and Define a Custom Interview.

Checking the Spelling of Library Entries

Problem

How can I spell check the items in my template library?

Solution

There are no spell-checking capabilities at the template library. You can, however, copy the text of a library to a word processor and spell check it there. You must manually update entries in the library with any corrections.

To spell check your library

- 1. At the template library, choose **File > Export Library to > Plain Text File**. The **Save As** dialog box appears.
- 2. Specify a file name and location for the text file and click **Save**. The file is created.
- 3. Open the plain-text file using Notepad.
- 4. Select all of the text and copy it. (Press **Ctrl+A** to select the text, and then press **Ctrl+C** to copy it.)
- 5. Open a new word processor document and paste the text. (Press Ctrl+V.)
- Use the word processor's spell checking tool. If there are spelling corrections that need to be made, return to the library and make the change. (To edit template titles, select the file and click the **Properties** button.)

Child Dialog Always Appearing in Interview Outline Even When it Shouldn't

Problem

I have a child dialog that is always appearing in the interview, even when there are times it shouldn't. How can I make the child dialog optional?

Solution

When you add a child dialog to a parent dialog, you can group the child dialog so users can more easily specify if they want to answer it. When you group child dialogs, HotDocs places a check box or option button in front of the child dialog icon. When this check box or option button is selected, HotDocs places the child dialog in the interview outline so that its contents can be answered. Otherwise, HotDocs will treat the child dialog as if it doesn't exist—even if there are required variables in it.

If child dialogs are not grouped, each child dialog will appear in the interview outline, regardless of whether the dialog is optional to answer. If the user does not answer every question in every dialog, HotDocs will report in the End of Interview dialog that there are unanswered questions.

For instructions on grouping child dialogs, see Group Child Dialogs in a Parent Dialog.

Choosing Which Word Processor Will Be Used By Default

Problem

I have multiple word processors installed on my computer. I want to use a specific word processor for creating templates but HotDocs keeps defaulting to the other word processor. I also want to send assembled documents to one word processor, but again, HotDocs defaults to the other. How can I fix this?

Solution

You can specify which word processor you want to use when creating templates. This same word processor will be used for viewing assembled text documents. For details, see Change Your Default Word Processor.

Comparing the Contents of Two or More Component Files

Problem

I need to compare the contents of two or more component files, but I can't seem to open two different copies of Component Manager to do so. Is there some way to do this?

Solution

You can use Template Manager to simultaneously compare the contents of two or more component files. For complete instructions, see Understand Parts of Template Manager Window.

Converting to HotDocs 11.1 Produces Component Name is Invalid Error

Problem

I'm trying to convert a pre-existing template to HotDocs 11 using the Component Manager. I get a message saying that it will be converted to 11 format, so I click **OK**.

I get a message that states "ERROR: The component name "my component name" is invalid. A component name cannot contain..."

Solution

You need to revert to 11.0.x or earlier to address the issue. The issue is caused by using an invalid variable name. This can include using a reserved key word such as a predefined language token, or by using all UPPERCASE letters in your variable names.

Once you have reverted to 11.0.x or earlier and follow the tips on naming your variables. You should then be able to convert to 11.1 without problems.

Converting WordPerfect Templates to Word

Problem

I recently switched from using WordPerfect to Microsoft Word. Can I convert my templates to Word format, and, if so, how?

Solution

You can convert your templates to Word format on a template-by-template basis, or you can convert multiple templates at once using Template Manager. (See Convert Multiple Templates to Work with HotDocs 11and Convert Templates and Clauses to Microsoft DOCX for details.)

Once converted, you should review each template carefully to make sure it still looks and works correctly.

'The Component File Cannot Be Used With This Version of HotDocs' Message Appearing

Problem

When attempting to edit a template, I get the error message, *The component file cannot be used with this version of HotDocs*. What does this message mean?

Solution

This message appears because you are trying to edit a newer version of a template with an older version of HotDocs. For example, you may be trying to edit a HotDocs 2009-11 version template with HotDocs 2005. Generally speaking, templates edited in the latest version of HotDocs are not backwards compatible with older versions.

If you must retain backwards compatibility with your templates, you can select a component file property that makes the template compatible with a specific version of HotDocs. If doing this, you must be careful not to use any features available in newer versions of the software, or else the template will not work with the older version. For example, dot codes are a new feature of HotDocs 2006. If you make a template containing dot codes available to HotDocs 2005 users, the template will generate errors because HotDocs 2005 has no support for dot codes.

See Compatibility of HotDocs 11 Files with Earlier Versions of HotDocs for details.

Dot Code Formatting Doesn't Seem to Work in True/False and Multiple Choice Options

Problem

Formatting dot codes (such bold, italics, and underline) assigned to Multiple Choice and True/False variable prompts don't work.

Solution

If a Multiple Choice or True/False variable prompt contains a formatting dot code and the variable is set to appear either as option buttons or check boxes, the prompt text will not be formatted. This is a known limitation of dot codes.

Editing Answer Source Records Causes Problems

Problem

I have an answer source whose records become jumbled anytime I edit records in it. Specifically, if I add a new record, some answers from other records appear in the new record. Those answers then disappear from their actual records. How can I keep this from happening?

Solution

Answer sources are associated with specific dialogs in a template. When users view the dialog during the interview, they can click a special button, which displays a list of answers that were entered during previous interview sessions. They can either select an existing set of answers or add a new set to the list.

After creating an answer source for one dialog, you can use it with other dialogs, including dialogs in other templates. However, when using an answer source with multiple dialogs, each variable must be represented in both the answer source and in each dialog. If a variable that is referenced in the answer source isn't included in all of the dialogs that link to it (or vice-versa), answers in the answer source will get "mixed together" whenever you add, edit, or delete records.

In some situations, though, it isn't always practical or relevant to show the user every variable—either in the dialog or in the answer source. To accommodate this, you should use the HIDE, CONCEAL, and OMIT instructions in a dialog script to manipulate these variables both in the dialogs and the answer source. Specifically, HIDE keeps a variable from appearing on the dialog, while CONCEAL keeps it from appearing in the answer source. OMIT keeps the variable from being associated with the answer source at all. Often, you must use a combination of these instructions to achieve your desired result.

Enabling Multiple Templates for Use with HotDocs Server At Once

Problem

I am automating a set of templates that I plan to use with HotDocs Server. All of these templates need to be enabled for use with Server. How do I specify this setting for all templates at once?

Solution

If you need to enable this setting for existing templates, you can use Template Manager.

To enable multiple templates for use with HotDocs Server at once

- 1. At your template library, select the templates you want to enable.
- 2. Click the **Template Manager** button. The **Template Manager** window appears.
- 3. Select the templates in the list and click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 4. Click the HotDocs Server tab and select Enable template for use with HotDocs Server.
- 5. Select any other Server options and click **OK**. The selected templates are now enabled for use with Server.

You can also set this option for all new templates you create. To do this, see Enable All New Templates for Use with HotDocs Server.

Entered Answers Getting Overwritten

Problem

In HotDocs 5, I could ask a question in the interview, allow the user to answer the question, and then use a SET instruction to assign a different answer to the question later in the interview. In current HotDocs, however, the answer used in the SET instruction is always overwriting any answers entered by the user. What can I do to fix this?

Solution

Simply stated, don't automate your templates this way. Because scripting in dialogs is processed each time you answer a question in the interview, you cannot ask a variable in an interview and then later SET that variable's value to something else. In all cases, each time the interview is updated, the SET instruction value will overwrite any answers the user enters.

See Introduction: HotDocs Interviews, Automatically Assign Answers to a Variable, and Check Interviews for Improper Scripting for details.

Entering Multiple Choice Option Prompts and Merge Text

Problem

I want to enter a long prompt and merge text for a Multiple Choice variable, but there isn't much room to enter it at the **Properties** tab of the **Multiple Choice Variable Editor**. Is there some way to enter the text and be able to see it all at once?

Solution

If you need to enter lengthy prompt text or merge text, you can do so at the **Options** tab of the **Multiple Choice Variable Editor**.

To add prompt text

- 1. Edit the Multiple Choice variable. (See Customize a Multiple Choice Variable.)
- 2. Click the **Options** tab. The view changes to show different ways you can customize the multiple choice options.
- 3. Click the **Option** drop-down button and choose the option you want to customize.
- 4. Enter the prompt text in the **Prompt** field.
- 5. Repeat steps 3 and 4 for any additional options you want to customize further.
- 6. Optionally, enter longer merge text in the **Default merge text** field.

Fixing Warnings Listed at Warnings Tab

Problem

The HotDocs Warnings tab shows several warnings listed in it. How do I fix them?

Solution

Many of these warnings pertain to variables being asked incorrectly in the interview.

To correct the problem, select the warning in the list and click the **Go to Warning** button. HotDocs takes you to the script or location in the template where the error occurs so you can correct it.

Warning	Description
Variables that are set to a value and are marked "Save in answer file"	When HotDocs generates an interview outline, it processes any SET instructions and assigns answers to the variables that are set. This causes HotDocs to ask to save the answer file, even if the user doesn't enter any answers during the interview.
Variables that are set to a value and asked in the same interview	You should not use a SET instruction to assign an answer to a variable and then ask the variable later in the interview. As HotDocs processes answers and rebuilds the interview, the value assigned by the SET instruction will always overwrite any answers the user provides. An example of this would be the following script:
	SET TF Variable TO TRUE
	(later in the template)
	ASK Dialog
	(where <i>Dialog</i> uses the variable, <i>TF Variable</i> .)
	In this situation, even if the user marks <i>TF Variable</i> as <i>false</i> , HotDocs will always change it back to <i>true</i> because of the SET instruction.
	If you want to suggest an answer for a user, use the DEFAULT instruction instead of the SET instruction. If you want a variable to always have a specific answer, then SET the answer and do not ask that variable anywhere in the template.

The following table lists each warning, followed by a description of what it means:

Variables that are asked more than once in the same interview	You should not ask the same variable more than once in an interview. If you do, HotDocs will always assign the last answer you gave to the variable, no matter where it is used in the interview.		
Variables that are referred to in a dialog script and are set to a value later	You should not refer to a variable in a dialog script and then set that variable to a value later in the template. When HotDocs reprocesses the interview, the set value may change the appearance of the dialog—including how it presents variables to the user and how it processes the answers.		
	An example of this would be the following script:		
	SET Client TO "Husband"		
	ASK Husband Information		
	SET Client TO "Wife"		
	ASK Wife Information		
	In and of itself, this script is OK. However, <i>Husband Information</i> and <i>Wife Information</i> are dialogs that contain the following dialog element text:		
	The following information applies to the «Husband or Wife».		
	<i>Husband or Wife</i> is a computation that merges the literal text "husband" or "wife" into dialog element text, depending on the answer to <i>Client</i> .		
	Because HotDocs continually updates the interview, it will always use the last value resulted for this computation, which means that the dialog element text will always merge the word "wife," even if the user is viewing the <i>Husband Information</i> dialog. This could be confusing to the user.		
Variables that are not asked when referred to but are asked later	You can refer to a variable in a template without asking the variable (by referring to the variable in an ASK NONE block or by clearing the Ask automatically option for the variable). However, if you do this, you should not ask the variable or set it to a value later in the template. Doing this will replace the answer that was used for the variable earlier in the template, thus causing an inconsistency.		

Foreign Characters Are Appearing as Boxes or Question Marks in HotDocs

Problem

I am using foreign (or Unicode) characters in my template, but they are appearing as boxes or question marks.

Solution

Even though HotDocs supports Unicode characters, these characters may not appear correctly if you do not have fonts installed that can display them.

For example, say you automate your template using the Tahoma font on Windows Vista. If the template is used on an earlier operating system, some of the characters you have used may not appear correctly because those characters may not be supported in the version of Tahoma available on that operating system.

Or, perhaps you use a very specialized font for displaying symbolic characters. If your users do not have the same font installed on their computers, your text will appear as unrecognized characters—usually boxes or question marks.

Additionally, you must also be careful when assigning font styles (such as bold or italic) to international characters, as characters may not display correctly when these styles are applied.

Getting Missing END IF or Missing END REPEAT Errors

Problem

When I test assemble my template, I get a *Missing END IF* (or a *Missing END REPEAT*) error. I have a lot of different instructions in my template—how can I determine where the END instruction is missing?

Solution

The HotDocs Outliner can help you quickly locate the opening IF instruction that isn't paired with a closing instruction.

To use the HotDocs Outliner

- 1. At the template, click the drop-down menu below the **HotDocs** button, then select the **I HotDocs Outliner** button. HotDocs displays an error message saying an END instruction is missing. (This is correct.)
- 2. Click on this warning and click **Go to Error in Template**. HotDocs highlights the opening instruction.
- 3. Determine where the closing instruction should appear and merge it in the template. (You can manually create the END field, or you can copy an existing field and replace the text between the chevrons with the appropriate END instruction.)

HotDocs Outliner is supported in only Microsoft Word 2000 and later.

Two other tools can help you match opening instructions with their closing instructions. These are the **\equiv\uparrowMatch Fields** tool and the **\checkmark\landLabel Fields** tool. See Match Opening Instructions with Closing Instructions and Use Labels to Identify Instructions for details.

The only time an END instruction should not be used is when inserting a REPEAT instruction in a table. See Use a Word Processor Table to Display a List for details. In all other cases, every IF and REPEAT instruction must have a corresponding END instruction.

Hiding the End of Interview Dialog During the Interview

Problem

When assembling the document, I want the users to go from the last dialog in the interview directly to the assembled document. I don't want the *End of Interview* dialog to be shown. How can I keep this from appearing?

Solution

You can specify a component file property that keeps the *End of Interview* dialog from appearing during the interview.

To specify this property

- 1. Open Component Manager.
- 2. Click the **Component File Properties** button. The **Component File Properties** dialog box appears.
- 3. Click the Interview tab. The view changes to show interview properties.
- 4. Select Hide End of Interview dialog.
- 5. Click **OK**.

Now, whenever users assemble this template, clicking **Next** at the last dialog will take them to the assembled document.

'HotDocs Cannot Play the Macro HD6EditTemplate.Main' Message Appearing

Problem

When attempting to edit a template, I get the error message, *HotDocs cannot play the macro HDEditTemplate.Main because the macro does not exist in the original template or in a template currently loaded into Word.* What does this message mean?

Several file names and locations were changed for HotDocs 10. For a list of these changes, please see New and Enhanced Features for HotDocs 10

Solution

The first thing to check is to make sure that HotDocs is putting its global macro templates in the right location, and also to make sure that Word is finding them. To do this, complete the following steps:

- 1. In Windows Explorer, browse to C:\Program Files\HotDocs\Source\Word\Macros, and make a note of the file dates on the files in that folder.
- In HotDocs, go to Tools > Options > Word Processors > Word 2000. Make a note of the Startup Folder location and browse to that folder in Windows Explorer. (You can click the Browse button next to the Startup folder field and it will open that folder in Windows Explorer.)
- 3. Compare the files in the folder opened through HotDocs with the files you found in step #1 above and make sure that **hd6edit.dot** and **hd6icon.dot** match exactly. (If they don't, you need to copy the files from the "source" folder to the "startup" folder.)
- 4. In Word, go to **Tools > Templates and Add-ins** and make sure that **hd6edit.dot** and **hd6icon.dot** are both listed and checked, which indicates that they are loaded.

Assuming these steps check out OK, another place to look is in Word's macro security settings. Although the HotDocs macros are digitally signed and should not be blocked even if you have your security set to the highest level, you could try temporarily lowering the security level to see if it makes a difference:

 In Word, go to Tools > Macro > Security. At the Security Level tab, lower the security level to Medium or Low and then see if the macros work under one of those settings. If it doesn't make a difference, be sure to switch it back to the previous setting.

Finally, you should check to make sure that the version of **hd_api.dll** in your Windows system folder is the correct version:

- 1. In HotDocs, go to **Help > About HotDocs > Version Info**. Make a note of the file version for **hotdocs.exe**.
- 2. In Windows Explorer, browse to C:\Windows\System32 and find the file named hd_api.dll.
- 3. Right-click on hd_api.dll and choose Properties from the shortcut menu.
- 4. At the **Version** tab, check to see if the file version is the same as that of **hotdocs.exe**.
- 5. If the versions do *not* match, delete the DLL file from the Windows folder and repair the HotDocs installation.

HotDocs is Creating Database File with Library File

Problem

I've noticed that there's a Microsoft Access database file saved in the same folder as my template library. This file also has the same name as my library. What is this file, and can I edit it?

Solution

When you run Template Manager, it creates a database file in the same folder as the library. This database is necessary in managing the different templates and components you are viewing in Template Manager.

You should not edit this file. If you were to inadvertently change or delete information in the database, it may corrupt your templates or make them unusable.

Incorrect formatting of multi-level list that includes HotDocs Instructions

Problem

When I assemble the document, the answers have inserted correctly but the format of the multi-level list is wrong. It looks like some of the returns are in the wrong place.

Solution

The most likely cause of this problem is an incorrect use of smart returns. You can set you default options for smart returns at the HotDocs Options dialog box.

Keeping Users from Editing Templates

Problem

I need to make templates I've automated available for assembly to several users within our office, but I don't want them to edit the templates. How can I restrict my users like this?

Solution

There are some options available for limiting use of your templates. Nearly all of the following options involve storing copies of the files to a different location (such as a network drive), separate from the folder that contains your working files:

• Publish the template files as auto-assemble files and make the auto-assemble files available to your users. See Publish Templates as Auto-Assemble Files (HDA).

- Export the library to a different location and specify that the exported files be used for assembly only. See Create a New Library By Exporting Part of an Existing Library.
- Obtain a publisher's license from HotDocs Corporation and publish the files for use with HotDocs Player. See Publish and Register Templates for Use With HotDocs Player. (Contact your HotDocs sales representative for information on acquiring a publishing license.)
- Mark the folder where the files are stored as read-only. Mark the template files within the folder as read-only as well. (See the Windows Help file for instructions on marking folders and files as read-only.)
- Publish the templates (as standard template files) but lock the component files. (Make sure you don't inadvertently replace your working component files with the locked component files. There is no way to recover a locked component file.) See Protect Published Files.

Limiting Visible Rows but Not Limiting Number of Answers

Problem

I want to show only a few rows of a spreadsheet dialog in my interview, but I don't want to limit the number of answers a user can enter. How can I do this?

Solution

You can control the number of rows that appear in a spreadsheet dialog. This setting has no effect on the number of rows a user can enter.

To limit the number of visible rows

- 1. Edit the spreadsheet dialog. (See Edit a Custom Dialog.)
- 2. In the **Rows to display** field, enter the number of the rows you want to be visible.
- 3. Click **OK**.

To limit the number of answers a user can enter, use the LIMIT NUM instruction in a dialog script.

Matching Pairs of Instructions in a Computation Script

Problem

I have a complicated script that incorporates several IF and REPEAT instructions. I need to be able to see these pairings better. I know there are several tools for doing this when editing instructions in the text of the template, but I'm unsure of what's available to me when I'm in a computation script.

Solution

There are a few things you can do to match pairs of instructions while working in a computation script:

• Click the **Auto Format** button in the script editor toolbar. This indents scripting between instructions. Nested instructions are likewise indented, like in the following example:

🛠 New Computation - Computation Editor [Invalid] - Claims test.cmp					
Properties Locals Advanced Resource Used In	Notes				
Variable <u>n</u> ame: Claims Address Block	Default <u>f</u> ormat: Non- <u>b</u> reaking				
Script: 🖹 🗢 ా 🖋 🗈 🛍 🖻	▲ ▲ 軸 나를 註 註 症 症 / []				
RESULT + "[address]" IF Claims Country = "United States" RESULT + SPACE(Claims Street, "/lb") IF Claims PO Box Needed RESULT + SPACE(Claims PO Box, ", END IF RESULT + SPACE(Claims City, ",") RESULT + SPACE(Claims State) + Clair ELSE IF Claims Country = "Canada" RESULT + SPACE(Claims Street, "/lb") IF Claims PO Box Needed RESULT + SPACE(Claims Street, "/lb") IF Claims PO Box Needed RESULT + SPACE(Claims PO Box, ", END IF RESULT + SPACE(Claims City, ",") RESULT + SPACE(Claims Province, ",") RESULT + SPACE(Claims Zip, ",") + Clair ELSE DECULT + CDACE(Claims Zip, ",") + Clair	/lb") ms Zip /lb") ims Country				
Components: Available Components	Instruction models: Image: ADD TEXT TO MULT_CHOIC ASCEND VAR ASCEND VAR Sev DATABACE C= Expression models: >= Enter a Date + Enter a Number - Enter some Text * /				
Test Update	QK Cancel Sa <u>v</u> e				

- Place your cursor in the opening or closing instruction and press Ctrl+M. HotDocs moves your cursor to the matching instruction. (You can also choose Match IF/REPEAT from the shortcut menu.)
- Place your cursor in the opening or closing instruction and press **Ctrl+Shift+M**. HotDocs highlights the entire section of the instruction with which you are working. (You can also choose **Select IF/REPEAT** from the shortcut menu.)

For details, see Use the Script Editor.

For details on working with instructions in a text template, see Match Opening Instructions with Closing Instructions.

Merging check boxes in a Text Template

Problem

I know how to automate check boxes in my form templates, but I'm unsure of how to create check boxes in a text template.

Solution

Some fonts on your system include empty box characters, as well as box characters with an X. You simply need to conditionally insert these characters in your template. For example, if you have the *Wingdings 2* font installed on your computer, you can complete the following steps to merge these characters.

To create check boxes in a text template

- 1. Automate your template, including specifying the conditional instructions that will merge the correct check box. (Use placeholder characters to represent the check boxes.)
- 2. Open the **Windows Character Map** tool. (For instructions on finding and opening this tool, refer to your Windows Help.)
- 3. Click the **Font** drop-down button and choose **Wingdings 2**. The character list changes to show the different characters available to you.
- 4. Locate a box with an X in it and click **Select**. This adds the character to the **Characters to copy** field.
- 5. Locate a box without an X in it and click **Select**. This adds this character to the box as well.
- 6. Click **Copy**.
- 7. In the template, place your cursor where the check boxes should go and choose the Paste command (for example, press Ctrl+V). The characters are inserted in your template. (You may need to highlight the character and choose Wingdings 2 from the application's font menu to reapply the correct font.)

A simpler way to merge check boxes may be to place either an X character or empty space character between opening and closing brackets. For example, [X] vs. [].

Merging Just a Portion of a Date Instead of the Whole Date

Problem

I need to merge just the year part of a date in an answer field—not the whole date. How can I limit the answer in this way?

Solution

You can use an example format to merge just a portion of the date. The user will still enter a full date, but only a portion of it will be merged in the document.

To merge the year only

- 1. Create a Date variable. (See Customize a Date Variable.)
- At the Format field (either at the Variable Field dialog box or at the Date Variable Editor), enter 1990.

Using example formats, you can merge a single Date variable in three separate answer fields. For example, say you are automating a form template and the date needs to appear on three separate blank lines. You can merge the same Date variable in all three blank spaces, but use different portions of a date format to control what gets merged in each blank space.

For example, in the following, you'd break the dates into three separate parts:

On the	day of	in the year	, the
undersigned personally appeared			

In the first blank space, you'd merge the Date variable with the format *third*. In the second blank, you would use *June* as the format. In the last blank, you'd use *1990* as the format.

Optional Variables Showing Up as Unanswered in Interview and Document

Problem

My template provides an option for users to enter a second address line. However, when users don't have a second address, HotDocs is merging an unanswered variable marker in the document. It's also reporting

in the *End of Interview* dialog that some questions are unanswered. How can I keep these markers and warnings from appearing?

Solution

To keep HotDocs from merging an unanswered variable placeholder, you must insert the variable conditionally. Specifically, you must insert it only if the variable is answered. For example, the following expression in the template will merge the second address only if the user provides it:

```
«Client Name»
«Client Address 1»
«IF ANSWERED(Client Address 2)»«Client Address 2»«END IF»
«Client City», «Client State» «Client ZIP Code»
```

To keep HotDocs from warning the user that the question in the dialog is unanswered, you specify a variable property. At the **Variable Editor**, click the **Advanced** tab and clear **Warn when unanswered**. (See Control How HotDocs Processes a Variable.)

Punctuating a Repeated List of Answers

Problem

I have created a REPEAT instruction in my template, but when I test it, the answers all run together, for example, like *Answer OneAnswer TwoAnswer Three*. How can I separate these answers with the correct punctuation and conjunction?

Solution

You can assign a format to a repeated list of answers. This format separates answers with either commas or semi-colons, and it also inserts the conjunction *AND* or *OR*, depending on your needs. To assign the format, at the **REPEAT Field** dialog box, click the **Format** drop-down button and choose the punctuation style you need.

For instructions on assigning a format in a computation script, see Punctuate a List Using a Computation Variable.

Receiving 'Invalid Variable Name' Errors When Creating Components

Problem

When I create a new variable and try to save it, HotDocs displays an Invalid Variable Name error message. What am I doing wrong that causes this error to appear?

Solution

When naming variables, there are a few rules you must follow:

- A variable name can have up to 50 characters, including letters, numbers, and some symbols.
- The first character in a variable name must be a letter.
- Each variable name must be unique. Even if the variables are different types, their names cannot be identical.
- DO NOT use all uppercase letters in your variable names. Because HotDocs instruction and expression keywords use uppercase letters, you may inadvertently use a word that may someday become a keyword, which will prevent HotDocs from reading your variable name correctly. (See Introduction: Instruction and Expression Models.)
- Do not use any of the following characters when naming your components:

. (period)

\$ (dollar sign)

" (quotation mark)

- : (colon)
- [] (brackets)
- , (comma)
- () (parenthesis)

% (percent)

- These characters can be used only if there is a character other than a space immediately before or after it:
 - + (plus)

- (hyphen)

* (asterisk)

/ (forward slash)

> < (greater than and less than signs)

>= <= (greater than or equal to and less than or equal to signs)

= (equals)

!= (does not equal)

Reducing Template File Size

Problem

I completed automation on my template and now the template's file size is several megabytes. What can I do to reduce the file size?

Solution

Most likely the increased file size is because you've used graphics in your template. When you insert a graphic in an RTF template, Word creates an extra copy of the graphic and merges it in the file. Other causes may be invisible merge ID marks and unused property codes.

You can keep Word from storing all of this information in the template. See Using the Hidden Data Remover dialog box for details. You can also remove existing hidden data from your templates. For details, see Remove Hidden Data from Word Templates.

Renaming Components Correctly

Problem

I renamed a component in my component file, but now my template won't assemble. I keep getting 'unrecognized variable' errors.

Solution

When you rename a component, it is changed everywhere it is used in the component file. For example, renaming a variable will update all references to it in dialogs, scripts, and prompts. However, you must update references to the component in the template text itself for the template to continue to work. To do this, you may need to remove the old variable field and insert the new variable. See Rename Components in a Single Template for details.

You can use Template Manager to rename components. If you do, all references to the component—both in the component file and in the template—will be updated. See Rename Components Across Multiple Component Files for details.
Repeating a Single Variable in a Dialog

Problem

I have a dialog that contains several related variables. I want to repeat one of the variables in the dialog but not all of them. How do I do this?

Solution

To repeat a single variable in a dialog that contains other variables, you must effectively create two dialogs and then nest the repeated variable's dialog in the non-repeated dialog. Specifically:

- 1. Create a dialog for all of the non-repeated variables. (See Gather Questions into a Custom Dialog.)
- Create a dialog for the repeated variable and assign **Spreadsheet on Parent** as its repeat **Style**. (See Choose a Presentation Style for the Repeated Dialog.)
- 3. Edit the non-repeated dialog and drag the repeated dialog to its **Contents** list.

See Insert Dialogs Into Dialogs and Create a List Within a List for more details.

Repeating the Contents of a Single Cell in a Table

Problem

I have a table. I need to repeat the contents of a single cell. I only selected that cell when I added the repeat instruction, but when I assemble the document, HotDocs creates an entire new row for each repetition. How can I create the list in a single cell and not repeat the entire row?

Solution

To repeat the contents of one cell in a table, rather than the whole row, you need to select only the text within that cell, rather than the cell itself.

You can tell when you have just the text within the cell selected when the grey highlight looks like the lefthand image below:

Select the text within the cell

Items	
«Items»	

Items	
«Items»	

Select the whole cell

To repeat the contents of a single cell, you can also create a REPEAT instruction in a computation script and then insert that Computation variable in the cell.

SET Instructions Using Dialog Elements No Longer Working

Problem

When I try to assemble a template that includes a Dialog Element with contents determined by a SET instruction, I see an error message.

Solution

You cannot use SET to determine the content of a Dialog Element. While this has never been a supported HotDocs feature it was possible to run these templates without error in earlier versions.

The suggested way to use a SET command in conjunction with a Dialog Element is by adding a text variable to the Dialog Element and then using the SET instruction, in a computation variable, to determine the contents of that text variable. For information on customizing Dialog Elements with variables see Use Variables and Scripts in Prompts, Dialog Element Text, and Resources

Test Assembling Template with ASSEMBLE Instructions

Problem

When I try to test assemble my template, the ASSEMBLE instruction in the template doesn't work.

Solution

You cannot test an ASSEMBLE instruction from within the template—you must first close the template and then assemble it from the library. See ASSEMBLE "FILENAME" for details.

Text Not Displaying Properly in Documents Assembled from Word 2007 Templates

Problem

When I assemble a document, some of the text is too far to the left; sometimes; the text is not even visible because it is too far to the left in the document.

Solution

The issue here is two fold: the existence of a smart tag in your template, and a bug in Microsoft's Open XML Format SDK. When the Open XML SDK encounters a smart tag is in the template, the SDK converts the "left" attribute of the <w:ind> to a "start" attribute. Word 2010 and above can handle the "start" attribute; however Word 2007 (which follows an older version of the DOCX specification) cannot handle the "start" attribute.

In Word 2007, since no "left" attribute is found, a default value of 0 is used. If your template also uses a large hanging indent, the assembled document's text can be so far to the left as to be completely off the page.

Removing Smart Tags

You can solve the issue described above by making sure you remove all smart tags in your template.

To remove smart tags:

1. In Word 2007, open the template containing smart tags.

Because later versions of Word (2010 and 2013) do not support smart tags, there is no way to remove smart tags in these later versions.

- 2. At the top left corner of Word, click the **Windows button**.
- 3. In the bottom right of the drop-down menu, click **Word Options**.
- 4. On the left, select Add-Ins.
- 5. On the Manage drop-down list, select **Smart Tags**; then click **Go**.
- 6. Be sure to deselect **Label text with smart tags**, or after you remove smart tags, Word automatically adds smart tags again.
- 7. Click **Remove Smart Tags** (if the document contains smart tags, the button should be enabled). All smart tags are removed.

True/False Variables Default to False on Spreadsheet Dialogs

Problem

When I include a True/False variable on a repeated dialog and set the style to Spreadsheet, the variable always defaults to False.

Solution

This is in fact expected behavior as True/False variables on Spreadsheet dialogs are primarily used as a type of check list. In the interview, when the user wants to answer false, they leave the question blank and when they wish to answer true they check the cell.

Trying to Save an Invalid Computation Script

Problem

When I try to save my Computation variable, I instead get an error message and HotDocs won't let me exit the script. I need to work on other projects, but I don't want to cancel and lose all of my work on this script.

Solution

HotDocs will not let you save an invalid computation script. However, you can 'disable' the script and save the variable. Disabling the script keeps HotDocs from trying to validate it when you save it.

There are two ways to disable a script:

Highlight all of the text in the script and click the ECOMMENT Block button. This turns all of the text in the script into a comment, which HotDocs won't process. When you are ready to work on the script again, highlight the text and click the ECOMMENT Block button. This enables the script once more.

Comments are descriptions or remarks about the computation script. Many developers enter comments to explain how they are using instructions and expressions in the script to achieve the result they are seeking.

• At the beginning of the script, enter the instruction **QUIT**. This instructs HotDocs to not process any of the script after the instruction.

Understanding the Difference Between Multiple Choice Options and Merge Text

Problem

What is the difference between a Multiple Choice option and merge text?

Solution

There are actually three parts to a Multiple Choice variable:

- **Options:** Multiple Choice options are the actual possible answers to the variable. When an option is selected, it is the value that is saved in the answer file.
- **Prompts:** Prompts are the text used to describe the options. For example, if your option text isn't descriptive enough, you can enter a prompt for the option and the prompt will be used in the interview. (If you don't enter a prompt, HotDocs will use the Option text.)

• **Merge Text:** Merge text is the text that gets merged in the document, based on the options the user selects. (If you don't enter **Merge Text**, HotDocs will merge the **Option** text in the document.)

See Customize a Multiple Choice Variable for details.

If you are using merge text in a computation variable, make sure you remember to use quotations around the multiple choice variable name otherwise you will be using the option text. For example using «MCVariable» will use the option text in the assembled document but using "«MCVariable»" will use the merge text.

Using DEFAULT Instruction Causes Extra Repetition

Problem

I have a DEFAULT instruction in my repeated dialog script, but it's causing an unanswered repetition to appear in my interview outline. This causes unanswered questions to appear in the document, as well. What do I do?

Solution

Do not use the DEFAULT instruction in the script of a repeated dialog unless the instruction is used in conjunction with a conditional expression or a LIMIT instruction. If you use it by itself in a repeated dialog script, it will always add an unanswered dialog to the interview, which will produce an incorrectly assembled document.

See DEFAULT VAR TO VALUE for details.

Using Headers and Footers of Inserted Templates

Problem

I have several inserted templates, each with their own defined headers and footers. However, when they are inserted in the parent template, the headers and footers are lost. How can I keep the headers and footers from being overwritten?

Solution

When working with inserted templates, you can select an option that uses the headers and footers of the inserted template when the template is merged in the main document. If the text is merged "inline" with the rest of the main document, the main document's headers override the inserted template's headers. If

the inserted template is merged as a separate section of the main document, each section will use the headers defined for it. See Define Headers and Footers in Inserted Word Templates for details.

Using INSERT Instructions in a Table

Problem

I've placed an INSERT instruction in a word processor table, but when I assemble the document, the inserted text appears outside the table.

Solution

You cannot place INSERT instructions in the cell of a word processor table.

Using One Variable in Two Different Dialogs

Problem

I have a template where I need to ask one of two dialogs, based on how users answer other questions in the interview. I have one variable that needs to appear in both dialogs. How can I link the variable to both dialogs without breaking my template?

Solution

As long as only one dialog gets asked during the interview, you can associate the variable with both dialogs. To do this, however, you will need to select some dialog options. You will also need to explicitly (but conditionally) ask both dialogs.

To associate the variable with both dialogs

- 1. Create the first dialog and add the required variables to it, including the variable you need to share between the two dialogs.
- 2. At the **Dialog Editor**, click the **Options** tab.
- 3. Clear Link variables to this dialog and click OK.
- 4. Create the second dialog and add the required variables to it, including the variable you've already added to the first dialog.
- 5. Either in a Computation variable or directly in the template, insert two ASK instructions—one for each dialog.
- 6. Condition the ASK instructions accordingly.

See Use the Same Variable or Clause in Two or More Dialogs for more information.

Variables Not Working in Word Text Boxes

Problem

I have a text field with a variable inside of it in my Word template. However, when I assemble the document, the variable isn't processed.

Solution

You cannot insert variables inside Word text boxes. Alternatives to using text boxes may include creating a single table cell to place your variable, or implementing a frame (Word XP). Refer to Microsoft Word's help file for specifics on creating these types of boxes.

Watermark Disappearing

Problem

When I assemble my text template using HotDocs Server, my watermark disappears.

Solution

Text watermarks do not work with HotDocs Server (or any browser based services, like HotDocs Cloud Services). To compensate, you need to turn your text watermark into an image and use it in Microsoft Word as a Picture watermark.

Add a Picture Watermark to a Template

- 1. Using image processing software, save the text or image you need as an image file on your computer.
- 2. At the HotDocs Library, select the template you want.
- 3. Click **Edit**.
- 4. On the Page Layout tab, click Watermark.
- 5. Click **Custom watermark**.
- 6. Select **Picture watermark**.
- 7. Click Select Picture...
- 8. Select the image file you saved earlier; then click **Insert**.
- 9. Click **OK**.

Document Assembly

Can't Attach Answer File or Document to Outlook 2007 E-mail Message

Problem

When I try to attach an answer file or an assembled document file to an e-mail message, HotDocs displays the following error message: "Error Sending Message. One or more unspecified errors occurred." I am using Outlook 2007.

Solution

This is an Outlook 2007 problem. It sometimes occurs when the Outlook 2007 forms cache becomes corrupt. To correct it (at least temporarily), follow these steps:

- 1. In Outlook, choose **Tools > Forms > Choose Form**. The **Choose Form** dialog box appears.
- 2. Click the **Look In** drop-down button and choose **Standard Forms Library**.
- 3. Select one of the forms and open it.
- 4. Close the form. (You don't need to make any changes.)

You may need to restart HotDocs for this to work.

Changing Unanswered Text Markers

Problem

When I leave questions in the interview unanswered, HotDocs merges asterisks in the assembled document where the answer is supposed to go. I'd rather have it insert blank lines or something else less obtrusive. Can I change this?

Solution

You can choose the unanswered marker HotDocs uses when you leave questions unanswered in the interview. For details, see Format Unanswered Variables in a Document.

The template provider may have selected a specific unanswered marker for some questions in the interview. If this is the case, these markers will override your selection.

Discerning Between Template Preview and Actual Template

Problem

When I preview the template at the **Preview** tab of the library or at the **Document Preview** tab of the assembly window, the preview looks different from the actual template or assembled document when I view it in the word processor. Why this difference in appearance?

Solution

Because of technical restrictions, HotDocs cannot display the actual word processor document in the library or at the assembly window. Instead, it uses a third-party application to render the text of the template or document.

This application depicts a Microsoft Word template or document fairly accurately because it is able to convert the text to RTF (if it's not already in RTF format). Some of the formatting may be different, but, for the most part, it is a good representation. However, because WordPerfect text can't be converted to RTF at the time the document is displayed, the application is only able to display the text in plain format—there is no formatting.

In either case, however, the actual text in the template or document is correct. And, once you edit the template in the correct word processor or send the assembled document to the word processor, the document will appear exactly as it's supposed to.

Downloading HDIs and HDAs as ZIP Files

Problem

When downloading HotDocs auto-install (.HDI) or auto-assemble files (.HDA) from a server, files are saved to disk in ZIP format. Additionally, attempting to open or run the file results in HotDocs prompting the user for a ZIP file password.

Solution

The first possible (and less likely) problem is that HDI file associations are not set up properly on the workstation. To test whether this is true, create an HDI file, close HotDocs, and then double-click the .HDI file in Windows Explorer. If HotDocs starts and then launches the installation, HDI files are properly associated. If HotDocs does not launch, the following are two solutions that should correct the problem.

- Solution 1: See Receiving Error Messages When Launching HDAs, HDIs.
- Solution 2: Repair your installation of HotDocs. To do this, go to Start > Settings > Control
 Panel > Add or Remove Programs. Select your version of HotDocs in the list of programs and
 click Change/Remove. When prompted to either repair or uninstall HotDocs, choose Repair and
 complete the installation.

The second possible (and more likely) problem occurs because Windows XP examines files being downloaded. If they appear to be in the ZIP format (which both HDI and HDA files are), then regardless of the download extension, Windows concludes they are ZIP files and applies a .ZIP file name extension. Even though you have a proper MIME type set up on your computer, Windows XP makes the change during the download.

To set the correct MIME types on the server

- 1. Bring up the IIS management tool (i.e. go to **Start > Programs > Administrative Tools >** Internet Services Manager).
- 2. Choose the Web site whose MIME types you need to change.
- 3. Right-click and choose **Properties** from the shortcut menu.
- 4. Click the **HTTP Headers** tab.
- 5. Click the **File Types** button.
- 6. Click the New Type button. (You will do this for both HDIs and HDAs.)
- 7. Enter the following information for each file type:

Type of File	MIME Type
Auto-install (HDI)	application/x-hotdocs
Auto-assemble (HDA)	application/x-hotdocs-auto

Editing Answers in Assembled Document

Problem

When viewing the assembled document, I can't edit the answers that are merged in the document. The help file says I should be able to.

Solution

You may not be able to edit answers at the **Document Preview** tab for one of three reasons:

- You are viewing a WordPerfect document. (Answer editing is not supported in WordPerfect templates and documents.)
- The template developer has not allowed it.
- The **Highlight Fields** option is turned off. To highlight fields again, click the **Highlight Fields** button in the assembly window toolbar.

Entering a Two-Digit Year

Problem

I want to enter a two-digit year whenever I answer a date question, but HotDocs always displays the following error message: "You must enter a four-digit year".

Solution

As a safety precaution, HotDocs requires you to enter four digits when entering the year portion of a date. If you enter a two-digit date, such as 7/9/99, HotDocs responds with this message: "You must enter a four-digit year."

You can override this default option by specifying a century rollover year at the **HotDocs Options** dialog box. The number you enter controls how HotDocs interprets two-digit years. Two-digit years less than or equal to the number you enter are understood as *2000*-century years. Two-digit years greater than the number you enter are understood as *1900*-century years.

For example, if you specify a rollover value of 34, dates entered as 5/14/34 will appear as 14 May 2034. A date entered as 5/14/35 will appear as 14 May 1935.

See Control How HotDocs Handles Two-Digit and Four-Digit Years for information on setting this option.

HotDocs Skips Questions When Navigating from Child Dialog

Problem

After I answer questions in a child (or nested) dialog, I click **Next** and HotDocs takes me to the next dialog in the interview—not back to the main-level dialog I was in originally. How can I keep this from happening?

Solution

HotDocs moves to the next dialog in the interview outline because of a default setting. Its behavior is most useful when child dialogs appear as the last item in a dialog, and, after answering questions in the child dialog, you want to move to the next dialog in the interview. However, there are times when you need to return to the parent dialog and continue answering questions in the current dialog.

If, instead, you want to return to the main-level dialog that uses the child, you can clear this default setting. At the assembly window, clear **Next Dialog Follows Outline** (**Navigate** menu).

Images Appear Incorrectly After Converting an RTF to PDF Using HotDocs Server

Problem

When RTF (or DOCX) documents are converted to the PDF format using HotDocs Server, images using a CYMK colour format are colored incorrectly in the resulting PDF document.

Solution

CMYK images are not supported for use in HotDocs templates. RGB images should be used instead.

- 1. Re-save the relevant source image in a RGB color format, using appropriate image editing software.
- 2. Re-insert the image into the HotDocs template.
- 3. Save the template and test a RTF(or DOCX) to PDF conversion in HotDocs Server. The image should now appear correctly.

Increasing Font Size

Problem

When answering questions in the interview, the text used in the dialogs is too small or too hard to read. Can I change it?

Solution

You can change the font, size, and color of the text used in a dialog. You can also change the properties of text used in the interview outline. For details, see Customize the Look of the Dialog Pane.

Installing Support for New Word Processor

Problem

I've installed a new word processor, but when I try to use HotDocs with it, HotDocs won't recognize it.

Solution

After you have installed the new word processor, you must run the word processor at least once. This creates entries in the Windows System Registry, which HotDocs must be able to access before it can integrate with it.

Once the registry is updated with the required information, you must select the new word processor at HotDocs Options and choose to install support for it. For details, see Install Support for New Word Processors.

It is possible your new word processor isn't compatible with HotDocs. If you are unsure, please see the list of HotDocs System Requirements (see System Requirements.)

Moving Between Questions in Interview Slow

Problem

Whenever I answer a question in an interview and try to move to the next question, it seems like it takes HotDocs a long time to respond. Moving between dialogs in the interview also seems slow. What can I do to make this process faster?

Solution

In an effort to keep the assembly process up to date, HotDocs updates the dialog and the interview outline each time you enter or edit an answer in the interview. Depending on the complexity of the interview or the document, this updating may take longer than desired.

You can minimize the amount of time HotDocs takes to update the interview or document with your changes by forcing it to update only when it needs to. You do this by turning **2** Instant Update off. When instant update is off, HotDocs updates dialogs and interviews only when it needs to. For details, see Update Your Interview Outline and Document.

Template Developers: You can create a custom interview component that can greatly reduce the amount of processing required to keep the interview up to date. This can improve response time for your users. For details, see **Define a Custom Interview**.

Moving to the Next Dialog After Entering a Date or Number

Problem

When I answer a date or number question in an interview and then click **Next** or **Previous**, HotDocs doesn't move me to the next dialog. I have to click **Next** or **Previous** again to move to that dialog.

Solution

When you type an answer in an answer field, HotDocs sometimes reads the answer and reformats it to appear a certain way. For example, if you type *February 6, 2006* in a date field, HotDocs will reformat the answer to appear as *6 Feb 2006*. Sometimes this reformatting can reveal an incorrectly interpreted answer,

especially if the answer you typed is ambiguous. For example, the date 05/06/2006 may either be interpreted as June 5, 2006 or as May 6, 2006.

Depending on where the reformatted answer appears in the dialog, you may not even be aware that it has been reformatted. For example, when you enter an answer in one of the first answer fields in a dialog and then move to the next field, you will most likely see that your answer has been reformatted. However, if you enter an answer in the last answer field and click **Next** to advance to the next dialog, you may not see that the answer was reformatted.

In this situation, HotDocs keeps you in the dialog so you can review the change it has made to the format. For information on changing this default behavior, see Warn When HotDocs Reformats Date and Number Answers.

Organizing Files in the Library vs. on the Disk

Problem

I don't understand the difference between how files are organized in my library and how they are organized on disk.

Solution

A HotDocs library is simply a tool for viewing and working with the files on the disk. Specifically:

- Files in the library are shortcuts that link to the actual files on disk (much like a desktop shortcut to an application you use frequently links to a program in your *Program Files* folder).
- Files in the library can be organized into "virtual" folders. These folders do not need to match the organization of folders on the desktop.
- Files in the library can be viewed using either the template title or the template file name. Files on disk are viewed only by file name.
- If you manually move a file on disk, you must update the reference to the file in the library or HotDocs will not know how to find the file. To move files while keeping the reference to the file updated in the library, use the **Move** or **Copy** commands in the library.

Overriding an Answer in a Form

Problem

What happens when I override an answer in a form document?

Solution

Sometimes you need to enter an answer in a form document, but you are unable to enter that answer because of formatting restrictions on that field. (Remember, dates and numbers must be formatted a certain way for HotDocs to consider them valid.) For example, say you enter *3,000,000* as an answer to a number variable. The field, however, may not be large enough to show the whole answer without sending it to the addendum. You can override the field's format and enter *3 mill.* as the answer.

Or perhaps you have a date that is appearing in one format (*19 May 2006*), but you want it to appear in a different format (*06/19/06*). You can override the field and enter the answer you need.

Overriding the format of the field allows you to enter whatever text you need in the answer field.

See Override an Answer in a Form for details on overriding form fields.

Overriding a field is like placing a sticky note over the field and writing on it—you can see what is written on the note, but HotDocs continues to use the original answer. Any other fields that use the answer, either in a calculation or a condition, will continue to do so. The original answer will likewise be saved in the answer file.

Printing Form Document is Off-Center

Problem

When I print my PDF-based form, the document appears off-center on the printed page.

Solution

This problem appears when you use Adobe Acrobat to print the document and you're using a non-PostScript printer driver.

To correct the problem, always use a PostScript printer driver.

See Print an Assembled Form Document for more details.

Printing Form Document Takes Incredibly Long Time

Problem

When printing a PDF-based form document, it takes an incredibly long time for the document to actually print. What can I do to speed up this process?

Solution

This slowness happens when you print the document through HotDocs. To print more quickly, use Adobe Acrobat or Reader to print the document. See Print an Assembled Form Document for details on selecting this preference.

Receiving Error Messages When Launching HDAs, HDIs

Problem

When attempting to run either a HotDocs Auto-Install file (HDI) or HotDocs Auto-Assemble file (HDA), HotDocs displays an error message and is unable to launch the file. The error message frequently says it cannot find *hd_dispatch.exe*, the application used to launch these files.

Solution

During installation, HotDocs creates certain entries in the Windows System Registry. One of these entries designates where the file *hd_dispatch.exe* should be found so that HDAs and HDIs can be properly run. By default, this file is usually created and registered in *C:\Program Files\HotDocs*. However, if you performed a custom installation and changed the program folder for HotDocs to some location other than *C:\Program Files\HotDocs*, this registry key was not updated with the new program files path. Because of this, HotDocs cannot run the HDA or HDI because it doesn't know where to look for *hd_dispatch.exe*.

To update the registry with the correct information

This workaround requires you to work in the Windows System Registry. Failure to follow these instructions carefully and entirely may result in you changing a setting that may adversely affect several other programs you use. If you are unsure of how to work with the registry, please ask your system administrator for assistance.

- 1. Click **Start > Run**. The **Run** dialog box appears.
- 2. In the **Open** field, type **regedit**. The **Registry Editor** appears.
- 3. Open **HKEY_CLASSES_ROOT** > **HotDocsHdaFile**. (To update the information for an HDI file, navigate to **HKEY_CLASSES_ROOT** > **HotDocsHdiFile**.)
- 4. Expand the folder so you can see its subfolders.
- 5. Click the **DefaultIcon** folder.
- 6. In the right pane, double-click (Default). The Edit String dialog box appears.
- 7. Change the folder path in the **Value data** field so that it matches the folder where HotDocs is installed. Do not remove the **,1**.
- 8. Click **OK**.
- 9. In the same subfolder (in the right pane), expand **Shell > open > command**.
- 10. Repeat steps 6 through 8, updating the folder path so that it matches the folder where HotDocs is installed. Again, do not change the information after the file path.

11. Remember to update the file locations for both HDAs and HDIs. See step 3 in the list above.

Restoring System Defaults to HotDocs

Problem

I've altered several default settings for HotDocs, but now I want to restore the default settings I had when I installed HotDocs. How can I do this?

Solution

HotDocs uses the Windows System Registry to store your personal preferences for working with HotDocs. If you need to restore the installation defaults, you must clear the *Current User* section of the HotDocs portion of the System Registry.

You must be extremely careful when working in the System Registry. Failure to follow the instructions below exactly could result in your making changes that negatively affect all of the programs on your computer. You may want to ask your system administrator for help if you are unsure of what you are doing.

To understand HotDocs installation and the changes that are made to the registry, please see Understand HotDocs Installation.

- 1. At the **Start** menu, select **Run**. The **Run** dialog box appears.
- 2. In the **Open** field, type **regedit** and click **OK**. The **Registry Editor** appears.
- 3. Navigate to HKEY_CURRENT_USER > Software > HotDocs > HotDocs.
- 4. Select the **HotDocs** subfolder and press **Delete**. Click **Yes** to confirm the deletion.
- Check for the registry key HKEY_CURRENT_USER > Software > LexisNexis > HotDocs 6. If it exists, proceed to Step 6. Otherwise, proceed to Step 7.
- 6. Select the HotDocs 6 subfolder and press Delete. Click Yes to confirm the deletion.
- 7. Close the **Registry Editor**.

Saving Answers During Interview

Problem

HotDocs won't let me save answers during an interview. The buttons and menu options for doing so are disabled.

Solution

If the buttons and menu options for saving an answer file are disabled, it is because the template provider has either prohibited you from saving the answer file, or because the answer file is being saved automatically. If you have specific questions, please contact the template provider.

Specifying Printing Options When Printing Text Documents

Problem

When I print my assembled text document from the assembly window, HotDocs just automatically prints the document without prompting me for any of my preferences. (I don't get a **Print** or **Print Options** dialog box.) How do I first set my preferences before printing?

Solution

To set your preferences before you print the document, you must first send the document to the word processor. Once in the word processor, choose the **Print** command and then specify your settings.

Text Not Displaying Properly in Documents Assembled from Word 2007 Templates

Problem

When I assemble a document, some of the text is too far to the left; sometimes; the text is not even visible because it is too far to the left in the document.

Solution

The issue here is two fold: the existence of a smart tag in your template, and a bug in Microsoft's Open XML Format SDK. When the Open XML SDK encounters a smart tag is in the template, the SDK converts the "left" attribute of the <w:ind> to a "start" attribute. Word 2010 and above can handle the "start" attribute; however Word 2007 (which follows an older version of the DOCX specification) cannot handle the "start" attribute.

In Word 2007, since no "left" attribute is found, a default value of 0 is used. If your template also uses a large hanging indent, the assembled document's text can be so far to the left as to be completely off the page.

Removing Smart Tags

You can solve the issue described above by making sure you remove all smart tags in your template.

To remove smart tags:

1. In Word 2007, open the template containing smart tags.

Because later versions of Word (2010 and 2013) do not support smart tags, there is no way to remove smart tags in these later versions.

- 2. At the top left corner of Word, click the **Windows button**.
- 3. In the bottom right of the drop-down menu, click Word Options.
- 4. On the left, select **Add-Ins.**
- 5. On the Manage drop-down list, select **Smart Tags**; then click **Go**.
- 6. Be sure to deselect **Label text with smart tags**, or after you remove smart tags, Word automatically adds smart tags again.
- 7. Click **Remove Smart Tags** (if the document contains smart tags, the button should be enabled). All smart tags are removed.

Understanding What Happens When Deleting Templates from the Library

Problem

What happens when you delete templates from a library?

Solution

When you select a template or folder in a HotDocs library and click the **Remove Items** button, HotDocs asks if you want to remove just the reference to the file from the library or if you want to delete the actual files from disk, too.

If you remove just the reference from the library, that specific reference to the file is removed from the library but the actual file still exists on disk. Additionally, if the item is referenced any other place in the library or in a different library, those references still exist.

If you choose to delete the item from disk, the reference is removed from the library and the file is sent to the Recycle Bin (if working with files on your local drive) or permanently deleted (if you're working with files on a network drive).

When you remove a folder from the library, you can choose to delete just the folder (in which case, items in the folder are moved to the next highest folder), or you can choose to delete the folder along with all of the items referenced in the folder.

For details, see Remove Items from a Library.

Updating Changes Made in Word Processor Document in the Answer File, Template

Problem

I've edited text in my assembled document but neither the template nor the answer file shows these changes. How do I update these files with my changes?

Solution

Once you send an assembled document to the word processor, it becomes "disconnected" from HotDocs. If you make changes to the document, those changes will not be represented in either the template or the answer file.

To update the template with your changes, you must edit the template and make the changes there. If you do not have editing privileges, you should notify the template publisher of your recommended changes.

If you need to update answers, you must reassemble the document using the saved answer file and make your changes during the interview. Make sure you save the answer file when you are finished.

Updating Cross-References in an Assembled Document

Problem

I've assembled a document and sent it to Microsoft Word, but now the Table of Contents is outdated. How do I update these cross-references?

Solution

To update cross-references in an assembled Word document

- 1. Highlight all of the text in the document (press Ctrl+A or choose Select All (Edit menu)).
- 2. Press the F9 key.
- 3. Select your update options and click **OK**. Cross-references in the document are updated.

Template Developers: You can select a component file property that automatically updates references in the document once the document is sent to the word processor. See **Change Component File Properties** for details.

Viewing File Names Instead of Template Titles at Library

Problem

I'm looking for a specific template in my library. I know what its file name is, but am not sure how to find it based on its title. Is there some way to display templates by their file names in the item list?

Solution

To view items in the library by file name, choose **File Names** (**View** menu). To show template titles again, choose **Template Titles**.

You can search a list of library items for a specific template by using the **Find** command. Simply select **Find** and then enter the text for which you are searching in the field. HotDocs will search both template titles and file names for the text you enter. Any files that contain that specific text will be displayed in the list.

HotDocs Automator

Double-Clicking Doesn't Create Field as Expected

Problem

I double-clicked to create a form field, but HotDocs didn't create the field as I expected it would.

Solution

Several factors may contribute to these difficulties:

- **Insufficient surrounding features:** To detect a field, HotDocs searches for surrounding features, such as lines, text, or graphics. If there are insufficient surrounding features, HotDocs may have difficulty detecting a field. In such cases, HotDocs creates a field of the default size.
- Label text: When label text is inside the field area and there is enough space between the text and the bottom of the field area, the **Detect** command will extend the field up to the bottom of the label text. If you want the field to occupy the area to the left or right of the label, or if you want to include the label inside the field, you must create the field manually.
- **Field not completely visible:** If part of the intended field is not visible (for example, it's scrolled out of the window), HotDocs attempts to scroll to detect the field. It is recommended that you use a zoom level such as **Fit Page to Width** so HotDocs can find the entire field on the screen.

If HotDocs is unable to automatically create or resize a field to the size and position you want, you must create the field manually. See Create a Form Field for details

Fixing Page Rotation Problems in PDF Templates

Problem

Pages that are rotated in a PDF document aren't rotated when I convert the PDF to template format.

Solution

Starting with the release of HotDocs 2006, page rotation is honored when creating a new template. However, you may need to update existing forms where page rotation is not honored.

To fix page rotation problems in existing PDF templates

- 1. Make a back-up copy of the form.
- 2. Edit the original form.
- 3. Delete all of the fields on the original form.
- 4. Save and close the form.
- 5. Reopen the original form. The rotated pages should appear correctly.
- 6. Open the back-up copy of the form and copy the fields for the non-rotated pages.
- 7. Create new fields for the rotated page.
- 8. Optionally, copy any printing or addendum properties from the back-up form.

Getting Errors When Automating Table in Form

Problem

When I try to group my fields in a table, I get errors that say things like *Row 2 has 1 columns when it should have 3 columns*. What am I doing wrong?

Solution

HotDocs expects fields in a table to appear in standard rows and columns that read from top to bottom and from left to right. When fields in a table are laid out differently, you must first define the rows and columns by specifying the field order before you try to group the fields as a table. For complete instructions on doing this, please see Merge a List of Answers in a Standard Table and Merge a List of Answers in a Nonstandard Table.

Keeping Control Field Text From Appearing on Form

Problem

I have a control field on my form template which runs a computation. When I assemble the document, the text that results from the computation script appears in the field—and I don't want it to. (For example, I want the result to go to the addendum.) How can I keep this text from appearing directly on the form?

Solution

You can keep text from being merged in a control field. To do this, you will increase the field border so that text that doesn't fit in the field will be sent to the addendum.

To keep text from appearing

- 1. Select the field and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 2. Click the **Layout** tab. The view changes to show options for field placement and appearance.
- 3. In the **Borders** group, enter **1** in the **Left border** field.
- 4. The underlying form field should turn a dark shade of green. If it does not, increase the field border until it does

See Change the Borders and Margins of a Field for complete instructions.

Modifying a Single Field in a Group

Problem

I want to change the properties of a single field in a group, but I can't seem to select the field. How do I get to the field so I can edit it?

Solution

You must first ungroup the fields and then make the changes to the field. When you are finished, regroup the fields. See Group Form Fields So Answers Can Flow From One Field to Another and Ungroup Form Fields.

Properties for field and table groupings are stored with the first field in the group. If you delete the first field in the group, you will lose these grouping properties.

Moving Form Fields Using Keyboard Takes Too Long

Problem

When working with fields in a form, I use keys on the keyboard to move them because I like the precision it affords me. However, this process, at times, takes too long. Is there some faster way to move fields using the keyboard?

Solution

To use the keyboard to move fields on the form, you can use the arrow keys. This moves the field pixel by pixel, which is useful when you want precise placement. To move the fields more quickly, press **Shift** as you press the arrow keys. See Move a Field on a Form for details.

Needing Characters in Answer to Appear in Individual Boxes

Problem

I have a form that requires a date be split into separate boxes—each number appearing in its own box. However, when I try to do this, all of the characters end up in the first few boxes of the form or the field overflows. How can I accomplish this?

Solution

To do this, you must group the fields and then assign a maximum number of lines and characters to the fields in the group. You may also need to assign an example format to the field. For complete instructions, see Attach a Variable to Single-Character Boxes.

No Interview When Assembling Form Document

Problem

I finished automating my form template and when I test assemble it, there isn't an interview. Instead, HotDocs just shows the **Form Document** tab and I have to type my answers directly in the fields. How do I get an interview for the template?

Solution

When users assemble form documents, they can provide the information in one of two ways: answering questions in an interview or typing answers directly on the document (which is called direct-fill assembly).

All forms, when first automated, are set to allow only direct-fill assembly. However, you can allow interview-based assembly by specifying an interview. The interview can be a default interview (one in which HotDocs determines the order dialogs are asked in the template) or it can be a custom interview (one in which you specify the order dialogs and variables are asked). Either way, you can designate which type of interview you want to use. For details, see Define the Interview for a Form Template.

If you want all new templates you create to use a default interview, you can specify a HotDocs option. See Automatically Generate Default Interviews for Form Templates for details.

Setting Form Properties Not Taking Effect in Existing Templates

Problem

At HotDocs Options, I've assigned several properties for how I want my form templates to look and work, but none of these settings are working in the templates I've created. How come?

Solution

Many of the properties you specify in the **Form Documents** section of HotDocs Options only affect *new* templates or fields you create. To update existing templates and fields you've already created with the settings, you must manually specify the properties while working in the template or working with the fields.

Tabbing Through Fields Sequentially in Form

Problem

When I tab through questions at the **Form Document** tab during a test assembly, I can't tab through the fields sequentially. Instead, HotDocs moves somewhat randomly between the fields. Is there some way I can correct this tab order?

Solution

When determining the order to ask questions in the interview, HotDocs begins with the top-left field of the form and processes all of the fields until it reaches the bottom-right field. This processing affects two things—the order questions are asked in the default interview and the order in which the user is able to tab through fields on the form.

Because of this default field ordering, sometimes you may find that the tab order during direct-fill assembly isn't working as you expect. To correct the problem, you must manually assign a tab order to the fields on the form. You do this at the **Additional** tab of the **Field Properties** dialog box. Specifically, you assign rows and/or column numbers to fields on the form. For complete details on assigning a tab order, see Change the Tab Order of Fields.

Want check box to Use a Tick in the form check box

Problem

I want to use an actual check mark in a form check box-not the default X character. How do I do this?

Solution

If you have symbolic fonts installed on your computer, you can 1) assign the font to the field, and 2) assign the special character to the True/False variable format.

The following steps provide two different ways of creating a tick mark using fonts.

To merge a check mark in a check box using the Symbol font

- 1. On the form template, create a check box field. (See Create a check box Field.)
- 2. Attach a True/False variable to the field. (See Customize a True/False Variable.)
- 3. Enter the following text in the **Default format** field: **Ö**/. (The capital O with an umlaut can be entered by pressing **Alt+0214**. This character represents the check mark character in the Symbol font. You may need to enter the character in a separate text editor and then copy and paste it into the **Format** field.)
- 4. Select the field and click the *Field Properties* button. The **Field Properties** dialog box appears.
- 5. Change the font to **Symbol**. (See Change the Font Used for a Field.)

When the template is assembled, HotDocs uses the check mark \checkmark instead of the **X**.

To merge a check mark in a check box using the Wingdings2 font

- 1. Select the grouped check box fields and click the *Field Properties* button.
- 2. Edit the variable field properties by clicking on the ****Variable** button.
- 3. Edit the output text based on variable type.
 - For a True/False variable in a single field: use a format of P/ (i.e 'P/<space>'); this will merge the correct character only if TRUE
 - For a True/False variable in two grouped fields: use a format of **P / P** (i.e 'P<space>/<space>P'); this will merge the correct characters across the two fields
 - For a Multiple Choice variable in two or more grouped fields: use the character P in each appropriate merge text field
- 4. Click **OK** to get back to the **Field Properties** dialog box.
- 5. Click the Font... button, set the font to Wingdings 2 and click OK.
- 6. Close the **Field Properties** dialog box.

Your check boxes will now show the \checkmark character instead of an **X**.

Legal Notices

Software License Agreement:

IMPORTANT NOTICE: PLEASE READ CAREFULLY BEFORE INSTALLING THE SOFTWARE. This license agreement ('License') is a legal agreement between you (an individual or a single entity) and the licensor ('Licensor') of this HotDocs software product, which includes computer software, the associated media (if any), printed materials (if any) and electronic documentation (together 'Software'). By opening the packaging for this Software (in the case of Software supplied on physical media) or by clicking on a button which says 'I agree' or 'I accept' (or any comparable action) during installation of the Software, you will be agreeing to contract with the Licensor on the terms of this License. Written approval of the terms in this License is not a prerequisite to the validity and enforceability of this License and no solicitation of any such written approval by or on behalf of Licensor is to be construed as an inference to the contrary.

IF YOU DO NOT AGREE TO THE TERMS OF THIS LICENSE, WE ARE UNWILLING TO LICENSE THE SOFTWARE TO YOU AND YOU MUST DISCONTINUE INSTALLATION OF THE SOFTWARE NOW BY CLICKING ON THE "I DO NOT ACCEPT" BUTTON BELOW. IN THIS CASE YOU MUST RETURN THE MEDIA ON WHICH THE SOFTWARE IS STORED AND ALL ACCOMPANYING DOCUMENTATION TO THE SUPPLIER, TOGETHER WITH PROOF OF PAYMENT WITHIN 14 DAYS OF PURCHASE FOR A REFUND.

If you purchased this Software from an authorised reseller, please note that the reseller has the right to enter into this license on our behalf.

If you reside in the European Union: the Licensor is HotDocs Limited, a limited company registered in Scotland, UK, with company number SC166375. This License, its subject matter and its formation (including non-contractual disputes or claims) is governed by and construed in accordance with English law and any disputes are subject to the exclusive jurisdiction of the English courts.

If you reside outside the European Union: the Licensor is HotDocs Corporation, a corporation incorporated in Delaware, USA, with its principal place of business at 387 South 520 West, Suite 210, Lindon, Utah 84042, USA. This License is governed by and construed under the laws of the State of Utah and the United States without regard to conflicts of laws provisions thereof. Unless otherwise elected by the Licensor in writing for the particular instance (which Licensor may do at its option), the sole jurisdiction and venue for actions related to the subject matter hereof shall be the Utah state and U.S. federal courts having within their jurisdiction the location of the Licensor's principal place of business. Both parties consent to the jurisdiction of such courts and agree that process may be served in the manner provided herein for giving of notices or otherwise as allowed by Utah or federal law.

If you have a written agreement signed by the Licensor which relates to the supply of the Software and which expressly states that its terms prevail over the terms of any 'click-wrap' or 'shrink-wrap' agreement packaged with the Software, then the former agreement will prevail over any inconsistent terms of this License. Otherwise, this License represents the entire agreement between us in relation to the licensing of the Software and supersedes any prior agreement, understanding or arrangement between us, whether oral or in writing.

THIS SOFTWARE REQUIRES A COMPUTER WITH SUFFICIENT MEMORY AND DISK SPACE, AND A COMPATIBLE OPERATING SYSTEM. SOME FUNCTIONALITY MAY REQUIRE A SPECIFIC INTERNET BROWSER OR OTHER SOFTWARE NOT INCLUDED WITH THIS SOFTWARE.

1) HotDocs License Keys. In this section, a 'license key' means a unique alphanumeric identifier supplied by the Licensor and required to be entered by the user during installation of the software. A 'machine ID' means an alphanumeric identifier which is unique to the relevant computer hardware (but which contains no other data) which merely distinguishes one hardware device from another. The Licensor reserves the right (but not the obligation) to use license keys, as further described in this provision, to record installations of some or all Software and, at the Licensor's option, restrict the execution of Software to the duration of the relevant license. Software will (i) attempt to connect over the Internet to a license key server operated by the Licensor; (ii) upload to the licensing server the license key and a machine ID for the machine on which the Software is installed; (iii) store a token on the local machine to record a successful attempt to connect to the licensing server as aforesaid; and, at the Licensor's option, (iv) operate only for a short time-limited period in the manner of evaluation software until a validated and subsisting license key is successfully verified. The licensing server with the Licensor's records of customers and sales for the purposes of comparing the installations of Software with the Licensor's record of licenses lawfully granted. The license key is licensed on the same terms as the Software.

2) If Software requires a license key, and until a license key is validated as described in section 1 of this License, the Software will be deemed to installed for evaluation purpose only. The Licensor hereby grants to you a non-exclusive, non-transferable license to use the Software on a single computer or workstation for evaluation purposes only for a maximum period of 30 days.

3) In consideration of you agreeing to abide by the terms of this License and the payment of the license fee, the Licensor hereby grants to you a non-exclusive, non-transferable license to use the Software on a single computer or workstation. You may install the Software on a second computer for use by the same person. The Software must not be shared or used concurrently on other computers. If any updates, maintenance releases or other modified versions of the Software are made available to you they will be deemed to be Software supplied under the terms of this License unless otherwise stated at the time of supply.

4) The Software is licensed for authoring and assembling templates under the direct control of the licensed user. This License does not permit use of the Software as part of an automated document assembly solution (for example, where document assemblies are instigated in bulk or through a database-driven solution).

5) You may make one copy of the Software for back-up purposes only. You may not transfer your rights to the back-up copy unless you also transfer your rights to the Software.

6) Except as expressly set out in this License or as permitted by any local law, you must not (and must not allow a third party to):

6.1. copy the Software except where such copying is incidental to normal use of the Software or where it is necessary for the purpose of back-up or operational security;

6.2. rent, lease, sub-license, loan, translate, merge, adapt, vary or modify the Software;

6.3. make alterations to, or modifications of, the whole or any part of the Software;

6.4. disassemble, de-compile, reverse engineer or create derivative works based on the whole or any part of the Software.

7) You acknowledge that all intellectual property rights in the Software throughout the world belong to HotDocs Limited, that rights in the Software are licensed (not sold) to you, and that you have no rights in, or to, the Software other than the right to use it in accordance with the terms of this License.

8) The Licensor warrants that the media (if any) on which the Software is stored and distributed are (at the time of supply), and will be for the period of 90 days after that time (Warranty Period), free from defects in design, material and workmanship under normal use. If a defect in the media occurs during the Warranty Period, the Licensor will replace it free of charge if you return it to the Licensor with proof of purchase and (so far as you are able) a documented example of such defect or error.

9) You acknowledge that the Software has not been developed to meet your individual requirements. It is your responsibility to ensure that the facilities and functions of the Software meet your requirements.

10) You acknowledge that the Software may not be free of bugs or errors and you agree that the existence of any minor errors does not constitute a breach of this License.

11) If, within the Warranty Period, you notify the Licensor in writing of any defect in the Software in consequence of which it fails to perform substantially in accordance with the electronic documentation, and such defect does not result from you having amended the Software or used it in contravention of the terms of this License, the Licensor shall, at its sole option, (i) repair or replace the Software or (ii) accept a return of the Software (with proof of purchase) and provide a refund of the purchase price, provided that you make available all information that may be necessary to assist the Licensor in resolving the defect, including sufficient information to enable the Licensor to recreate the defect.

12) Nothing in this License limits or excludes the liability of either party for death or personal injury resulting from negligence, fraud, fraudulent misrepresentation, or for liability which cannot lawfully be excluded or limited.

13) Subject to section 12:

13.1. the Licensor will not be liable for any loss of profit, loss of business, loss of goodwill, loss of savings, loss of data, loss of anticipated savings, whether direct or indirect in each case, or for any indirect loss or consequential loss whatsoever and howsoever caused (even if caused by the Licensor's negligence and/or breach of contract and even if the Licensor was advised that such loss would probably result);

13.2. the Licensor's maximum aggregate liability under or in connection with this License, or any collateral contract, whether in contract, tort (including negligence) or otherwise, is limited to a sum equal to the actual price you paid for the Software.

14) This License does not provide for maintenance or support for the Software, which must be purchased separately if required. Licensor has no obligation to provide maintenance, support or updates, modifications or new releases under this License.

15) This License sets out the full extent of the Licensor's obligations and liabilities in respect of the Software. There are no conditions, warranties, representations or other terms, express or implied, that are binding on the Licensor except as specifically stated in this License. Any condition, warranty, representation or other term concerning the Software which might otherwise be implied into, or incorporated in, this License, or any collateral contract, whether by statute, common law or otherwise, is hereby excluded to the fullest extent permitted by law.

16) You must not transfer, assign, charge or otherwise dispose of this License, or any of your rights or obligations arising under it, without our prior written consent.

17) Upon request from time to time, you agree to certify to the Licensor the number of copies of the Software installed on your computers.

Third-Party Software Acknowledgements

Colosseum Builders, Inc.

This product uses the image library software created by Colosseum Builders, Inc., to load, decode, draw, and print .BMP, .JPG, and .PNG files in form templates and documents.

Copyright

Copyright © 1996-2018 AbacusNext.

Contact HotDocs Sales and Support

HotDocs Technical Support

Support for customers with technical support agreements is available by calling the numbers below. To expedite your call, please be at the computer on which the program is running.

For the United States, Canada, and Mexico:

Method of Contact	Information
Telephone	(800) 828-8328 (U.S.)
	(801) 615-2200 (International)
	U.S. technical support is available from 7:00am to 6:00pm (MST), Monday through Friday.
E-mail	support@hotdocs.com
Web Site	http://www.hotdocs.com/support/

For Europe, Africa, Asia, Australia, and South America:

Method of Contact	Information
Telephone	0843 208 0704 (U.K.) +44 131 226 3999 (International)
	U.K. technical support is available from 9:00am to 5:00pm (GMT), Monday through Friday.

E-mail	tech@hotdocs.co.uk
Web Site	http://www.hotdocs.co.uk

You may also find answers or solutions to questions you have in the HotDocs Wiki. To view the Wiki (as well as access other support options), please visit http://wiki.hotdocs.com.

Click here for information on providing the HotDocs Publications team with documentation feedback.

HotDocs Sales Support

Experienced HotDocs consultants are available to help you with a variety of services, including integrating HotDocs with other products, building a template library, or providing training. Please contact your sales representative to learn more.

For the United States, Canada, and Mexico:

Method of Contact	Information
Telephone	(800) 500-3627 (U.S. Sales)
	(801) 615-2200 (U.S. Business) +44 131 226 3999 (International)
Fax	(877) 356-3627 (U.S.)
	(801) 868-3627 (International)
E-mail	<u>sales@hotdocs.com</u>
Web Site	http://www.hotdocs.com

	http://www.hotdocs.com/products
	http://www.hotdocs.com/services
Address	387 South 520 West
	Suite 210
	Lindon, Utah 84042

For Europe, Africa, Asia, Australia, and South America:

Method of Contact	Information
Telephone	
	0131 226 3999 (U.K.)
	+44 131 226 3999 (International)
Fax	
	0131 220 9024 (U.K.)
	+44 131 220 9024 (International)
E-mail	info@hotdocs.co.uk
Web Site	http://www.hotdocs.co.uk
Address	
	14 South Charlotte Street
	Edinburgh, EH2 4AX
	Scotland

Corporate Offices

Head Office U.S.

Head Office U.K.

Contact HotDocs Sales and Support

HotDocs Corporation	HotDocs Limited
387 South 520 West	14 South Charlotte Street
Suite 210	Edinburgh, Scotland
Lindon, Utah 84042	EH2 4AX
Tel: 801-615-2200	Tel (UK): 0843 208 0704
Sales: 800-500-3627	Tel (Int'l): +44 131 226 3999
Fax: 877-356-3627	Fax: +44 131 220 9024
info@hotdocs.com	info@hotdocs.co.uk

Documentation Feedback

To improve the quality of our documentation, we invite you to make comments or suggestions. When doing so, please include as much information about your experience using the documentation as possible. For example, include the product name and version, as well as the title of the topics you want to comment on. You can e-mail your comments and suggestions to publications@hotdocs.com.

The HotDocs Publications team cannot respond to technical support or project consulting questions. We are interested in suggestions that improve the documentation itself—such as those concerning erroneous information, grammatical and spelling errors, or suggestions for topics to include in the next release of the software. For other concerns, please contact Sales and Support.
Glossary

.ANS: File name extension designating a HotDocs answer file. Traditionally .ANS designates a binary answer file, but starting with the release of HotDocs 2009, all answers files (even those with the .ANS file name extension) are saved in XML format. The .ANS file name extension is retained to ensure backwards compatibility with HotDocs 2008.

.ANX: File name extension designating a HotDocs XML answer file.

.CMP: File name extension designating a HotDocs component file.

- .DOC/.DOCX: File name extension designating a Microsoft Word document. (See text document.)
- .DOT: File name extension designating a Microsoft Word DOT template. (See text document.)
- .HDA: File name extension designating a HotDocs auto-assemble file.
- .HDI: File name extension designating a HotDocs auto-install file.
- .HFD, .HPD: File name extensions designating HotDocs form documents -- .HPD designates a HotDocs PDF Form Document, and .HFD designates a HotDocs Form Document (based on the legacy Envoy file format).
- .HFT, .HPT: File name extensions designating HotDocs form templates.
- **.PDF:** File name extension designating a Portable Document Format file, a format created and supported by Adobe. PDFs are a useful way of distributing documents in a format most users can view—as long as they have Adobe Acrobat, Adobe Reader, or HotDocs Filler installed. Template developers can also create PDF-based form templates (.HPT files) from which PDF files can be assembled.
- **.RTF:** File name extension designating a Word RTF (Rich Text Format) file, or a HotDocs RTF template. (See text template.)
- .WPD: File name extension designating a WordPerfect document. (See text document.)
- .WPT: File name extension designating a WordPerfect template. (See text template.)

Α

- **accelerator:** A key or key combination that quickly performs routine tasks in HotDocs. For example, rather than click the Print button, a user can press Ctrl+P and the document will be printed. Accelerators are useful when users don't want to use the mouse.
- **addendum:** The last section of a form document that contains answers that don't fit in their allotted fields on the actual form. (See also overflow.)
- additional text: See dialog element.
- **ADO:** Short for ActiveX Data Objects, it's a data presentation layer that lets HotDocs communicate with a database so HotDocs can retrieve data from it and use it to assemble a document. (See also ODBC.)

- **ANS:** File name extension that designates that the file is a HotDocs answer file. Starting with the release of HotDocs 2009, all answers files (even those with the .ANS file name extension) are saved in XML format. The .ANS file name extension is retained to ensure backwards compatibility with HotDocs 2008.
- **answer file:** A saved file that contains the answers gathered by an interview. Often template developers save an answer file so they can reuse the answers later to assemble other documents to which the answers are pertinent.
- **Answer File Manager:** The library used to manage answer files. With Answer File Manager, users can group answer files, view histories of their usage, and so forth. (The alternative is using Windows Explorer to find, view, and use answer files.)
- answer library: See library and Answer File Manager.
- **answer management:** The system of using Answer File Manager to store and manage answer files. (The alternative is using Windows Explorer to find, view, and use answer files.)
- **answer sharing:** The process of creating and using same-named variables in multiple templates so that a user can use the same answer file to assemble multiple documents. (Can also be called variable flow-through.)
- **answer source:** An answer file that is linked to a specific dialog in an interview. Users can enter their answers in an answer source and have those answers available to them on demand. (During an interview, a Select button appears on the dialog. The user clicks this button and has access to the answers in the answer source.)
- **Answer Summary:** A brief report HotDocs generates that lists the questions asked during an interview, followed by the answers that were entered. (See also Question Summary.)
- **answer wizard:** A button attached to a form field that users can click during direct-fill assembly. When they click the Answer Wizard button a pop-up interview appears, asking one or more questions that are required in order for an answer to be merged in the field. Frequently, answer wizards are assigned to inactive fields in a form.
- **answers:** Data users enter during an interview. Answers are usually merged into the document, but sometimes they are used to calculate other answers that are used in the document. See also answer file.
- **ANX:** File name extension that designates a HotDocs answer file.
- **ascend:** The process of sorting answers in alphanumeric order, from 1 to 9, and from A to Z. (See also descend.) You can also sort items in a template library, clause library, and answer library.
- **ASK instruction:** An instruction that forces a dialog to be asked at a specific location in the script or template. Frequently, ASK instructions are used when creating an interview component. They allow developers to control the order in which dialogs are asked during the interview.
- assemble: See document assembly.
- **Assembly Queue:** A dialog box that shows a list of assemblies—pending, current, and completed. Users can open the Assembly Queue by clicking Assembly Queue in the assembly window toolbar. It is most useful when users have selected multiple templates for assembly.
- **assembly window:** The window that appears when a user selects a template to assemble. By default, it includes the Interview tab, the Document tab, the Question Summary tab, and Answer Summary

tab. Each of these tabs displays something unique about the document being assembled, such as the questions that are required to customize the document or the assembled document itself.

- **auto-assemble file:** A self-executable file that contains one or more templates and their related files. When packaged in an auto-assemble file (or HDA), the files are temporarily extracted and used to assemble the document. Once assembly is complete, the extracted files are deleted. Auto-assemble files are useful if template developers don't want users to have editing access to the template files themselves.
- **auto-install file:** A self-executable file that contains one or more templates and their related files. When extracted, the files are saved to disk and references to them are added to a library. Auto-install files provide a useful way to distribute templates or updates to template sets.
- **automate:** The process of converting any document (text or form) into an interactive template. At its very core, automation is replacing changeable text in the document with variables. Additional automation steps include making text in the template conditional, repeating sections of the template so multiple answers can be entered, and inserting other boilerplate text into the template.

Automator: See HotDocs Automator.

В

- **bar code:** A format for an answer or a group of answers so that data can be quickly scanned using an optical scanner. Bar codes are supported in both form templates and text templates. (In form templates, developers assign the PDF417 property to the field. In text templates, developers assign the preferred bar code font at the Advanced group of the Variable Field dialog box.)
- **binary files:** In versions of HotDocs prior to HotDocs 2009, represents the format HotDocs-specific files such as library files, component files, and so forth—were saved in. Binary file formats are common in most software applications. They allow information about the files to be encoded for storage and processing purposes. However, one limitation of storing information in HotDocs in binary format is that third-party application developers aren't able to inspect, edit, or otherwise make use of information contained in the files. Another limitation is that the binary formats used in HotDocs do not support the use of foreign characters (for example, international characters that are not represented in your computer system's default language).
- **browser:** A window that allows users to view HTML documents. When working with HotDocs Server, interviews are displayed in a browser window rather than the regular HotDocs assembly window.
- **built-in variable:** A predefined variable that performs a special function in a template, such as inserting either today's date or the name of the current answer file. Built-in variables include TODAY, PN#, ANSWER FILE NAME, and COUNTER.

С

- **century rollover:** A HotDocs setting that controls whether years entered as two digits appear as 1900-century years or 2000-century years.
- **check-box field:** A type of form template field that represents some sort of pre-existing option a user must select, such as a true/false value or a multiple-choice value.

- **chevrons:** The double-angle brackets (« ») that surround a variable in a text template. Together, the chevrons and variable name make up the variable field, for example, «Client Name».
- **child dialog:** A dialog that is inserted within another dialog. When it's inserted, it becomes linked to that dialog—users can't answer questions in it without first viewing the parent dialog. Usually the two dialogs are related in content or purpose.
- **clause:** Predefined sections of text that can be selected and added to an assembled document. Usually clauses are grouped together in a clause library so users can choose which ones they want to insert, although some clauses are merged in the document automatically.
- **clause archive:** A compressed file that contains all of the clauses for a given template or clause library. During assembly, clauses in the archive are extracted so they can be selected and added to an assembled document.
- **clause library:** A file that contains a listing of available clauses. Users can select the clauses they want to use from the library, designate the order they should be merged in the document, and then answer any questions contained in the clauses. The clauses are then merged in the document.
- Cloud Services: See HotDocs Cloud Services.

CMP: File name extension that designates that the file is a component file.

- **command-line option:** An instruction used to control the operation of HotDocs. These instructions, or commands, are added to any command line that causes HotDocs to run. They can alter the operation of specific templates, or they can affect the overall operation of HotDocs.
- **comments:** Notes or thoughts entered by the template developer either in a script or in a template. Comments are one way to document processes within the template. If entered correctly, they will not be visible to users in the assembled document.
- **component:** A template resource file that displays or stores information about the user input/answers to merge in the final document. Examples of components include variables, dialogs, dialog elements, merge text groups, and formats.
- **component file:** The file that stores all of the components used in a template. The component file and template file are both necessary for template development and document assembly to work correctly. Template developers use Component Manager to work with components.
- **Component Manager:** The tool used to coordinate component usage in a template. Component Manager shows all of the components used in the template and provides options for working with those components.
- **computation variable:** A type of component that performs calculations or executes other instructions within the template. Computation variable scripts are created using the HotDocs scripting language.
- **conditional text:** Text in the template that should be included in the assembled document only under certain circumstances. Conditions are controlled using IF instructions and expressions.
- **control field:** A type of form template field that is used for behind-the-scenes tasks, including inserting related templates and assigning values to variables, just to name a few.
- **COUNTER:** An expression that keeps track of the current number of repetitions in a repeated dialog. Each time a new repetition is added, the COUNTER is increased.

custom interview: A script that controls how and the order in which variables and dialogs are asked during an interview. The template developer creates this script.

D

database: A file that contains a collection of data. Template developers can map variables in templates to fields in a database table so that answers can be retrieved from it and merged in the assembled document.

Database Connection: See HotDocs Database Connection.

- **date detection:** The HotDocs setting that controls how HotDocs interprets and merges dates entered during the interview—for example, whether the date appears as DAY MONTH YEAR (British), or MONTH DAY YEAR (United States).
- **Date variable:** A type of component that merges a date in the document.
- **DEBUG:** An instruction developers can insert in a template or script that lets them troubleshoot problems they are experiencing with their automation. While testing the script or template in debugging mode, HotDocs walks the developer through it, step by step, so he or she can see exactly how the script or template is producing the unexpected result.
- **default interview:** The interview HotDocs automatically generates based on the order variables are asked in the template.
- **default word processor:** When multiple word processors (for example, Word and WordPerfect) or when multiple versions of a single word processor (for example, Word 2000 and Word XP) are installed, the word processor HotDocs uses by default for automation and document assembly.
- **delimiter:** A character, such as a tilde (~) or vertical bar (|), that delineates answers or values in a script or instruction.
- **descend:** The process of sorting answers in reverse alphanumeric order, from 9 to 1, and from Z to A. (See also ascend.)
- detect: In a form template, the process of aligning a variable field with its surrounding field borders.

developer: See template developer.

- **dialog:** In template development, represents the component in which the developer groups variables and other components. In document assembly, represents the group of questions in the Interview tab of the assembly window where users enter their answers.
- **dialog element:** A component that lets developers more easily add additional text, hyperlinks, buttons, graphics, lines, and spacing to dialogs. These can help make the dialog more visually pleasing and informative.
- **dialog script:** A set of instructions that dynamically change how variables are used in a dialog. For example, questions about a client's children can be scripted so they become available only if the client indicates that he or she has children. If the client does not, the questions will not be available.
- **direct-fill assembly:** The process of entering answers directly at the Form Document tab of the assembly window rather than answering questions at the Interview tab.

- **DOC:** File name extension that designates that the file is a Microsoft Word document. (See text document.)
- **document:** The file that is created after a template has been assembled.
- **document assembly:** The process HotDocs goes through as it processes scripts in the template and merges answers into the document produced from the template. At the end of the assembly process, the user has a document tailored to his or her needs.
- **document manager:** A third-party application that stores various data files, including documents and answer files. Using a document manager, users can track versions and show histories of the document as well as enter other physical data about the files being stored, such as the date they were created, who created them, and so forth.
- **Document Preview tab:** A tab of the assembly window that shows how the text document has been assembled using the answers entered in the interview. (See also Form Document tab and Interview tab.)
- **DOT:** File name extension that designates that the file is a Microsoft Word DOT template. (See text document.)
- **dot code:** A property that can be assigned to text in a prompt, dialog element, resource, or computation script that either formats the text or inserts characters not commonly supported in plain text. For example, dot codes can be used to apply a bold format to a word in a prompt, or they can be used to insert the subsection (§) symbol.

double-angle bracket: See chevrons.

driver: See printer driver.

duplicate: The process of copying a variable to create a new one.

Ε

Edit field: A type of form template field that is used for entering text, dates, and numbers. It is the most commonly used type of field on a form.

ELSE IF / ELSE: See IF instruction.

- **End of Interview dialog:** The last dialog displayed in an interview, which contains a report of the number of questions that are still unanswered. It also provides options for working with the assembled document.
- **Envoy:** A third-party product used for creating HotDocs form templates (or HFTs). Support for the Envoy printer driver ended with the release of HotDocs 2006 (Envoy required 16-bit compatibility, which was only supported on Windows 98 and earlier). HotDocs still supports automation of HFT files just not creation of these files. All new form templates must now be created in PDF format.
- **example format:** A predefined format for how an answer should look when it is merged in the assembled document. This allows the user to enter the answer however they want in the interview, but forces it to appear a specific way in the finished document.
- **explicit index:** A reference to a specific answer in a list of answers. For example, to merge the third answer from a list, a template developer would assign the index number of [3] to the variable that is being merged, like this: «Service Date[3]». The third date in the list would then be merged.

expression: A command in a script that retrieves a special value. Expressions help calculate dates, sums, and so forth.

F

- **field:** A place in the template that denotes where users' answers should be merged, or where a specific instruction should be executed. In a text template, a field is denoted by chevrons. In a form template, a field is denoted by a colored box that is overlaid on the form's static text.
- **file name extension:** Three characters appended to a file name that identify the type of file so Windows knows what program to use to work with the file.
- fill: The process of assembling a form document.
- **fill character:** Character that is used to pad an answer that doesn't meet the required width of the field. For example, perhaps an answer needs to be 15 characters long. If the answer is only 10 characters, a fill character can be used to bring the answer to 15 characters.
- **fillable field:** In Adobe Acrobat or Reader, represents a dynamic field in which a user can enter data while viewing the document. Using HotDocs, users can create fillable PDF templates from these PDF documents that contain fillable fields.
- Filler: See HotDocs Filler.
- **filter:** A script that removes unrelated or unwanted answers from a list of answers. For example, perhaps there is a list of a client's children but only the names of minor children should be merged. A filter can extract just this data from the list.
- **fixed value:** A predefined answer, such as a date, number, or string of text. When working with instructions and expressions, placeholders are replaced either with fixed values or with variables.
- **foreign language DLL:** A file that allows template developers and users to access Date variable and Number variable formats in languages other than English. This allows these dates and numbers to be formatted correctly in the assembled document. Supported languages include French, Spanish, German, Swiss German, Austrian German, Dutch, and Italian. From HotDocs 10 onwards, the functionality provided by the Foreign Language DLL's is integrated with HotDocs.
- **form document:** The file that is created from an assembled form template. Form documents are distinguished from text documents by the design of the document—forms are static in nature, meaning the underlying text of the document cannot be changed or modified. (See also text document.)
- **Form Document tab:** A tab of the assembly window that shows how the form document has been assembled using the answers entered in the interview. When viewing the Form Document tab, users can enter or change their answers by clicking on the form fields and changing the answer.
- **form template:** A template that is created and automated in HotDocs Automator. It is distinguished from a text template by the fact that the underlying text cannot be modified because it is static. (See also form document.)

form wizard: See custom interview.

format example: See example format.

function: In programming, a function is a piece of code that operates on one or more inputs, and produces an output.

G

- **GRAY:** An instruction used in a dialog script to disable a variable unless the user answers another variable a specific way. For example, questions asking for children's name can be grayed until the user answers a question indicating he or she has children. (See also UNGRAY.)
- **group:** The process of designating that two or more fields in a form template be linked together so that an answer can flow from one field to the next. For example, if a text answer needs to span two or more lines in a form, the fields can be created on each line and then grouped. When the answer extends beyond the first line, it will flow to the second line. (See also run-on group.)

Н

HDA: File name extension that designates that the file is a HotDocs auto-assemble file.

- HDI: File name extension that designates that the file is HotDocs auto-install file.
- HFD, HPD: File name extension that designates that the file is a HotDocs form document.
- HFT, HPT: File name extension that designates that the file is a HotDocs form template.
- **HIDE:** An instruction used in a dialog script to hide variables in the dialog. (See also SHOW.) Usually this instruction is conditioned so that questions in the dialog hide and show dynamically, based on answers the user enters.
- **host application:** A (usually third-party) web application that "hosts" HotDocs Server or Cloud Services features on the back end to present interviews to end users in a web browser. A host application enables template (and related file) uploading and storage, as well as interview generation and presentation, final document assembly, and answer file handling and storage.
- **HotDocs Automator:** The tool used to automate form templates, or those templates whose underlying static text cannot be changed. Examples of form templates include tax preparation forms, applications, and so forth.
- **HotDocs Cloud Services:** The cloud-based version of HotDocs that displays interviews and your customers' final assembled documents within a web browser. HotDocs Cloud Services does not require end-users to install any HotDocs software on their computers.
- **HotDocs Compare:** Starting with the release of HotDocs 2009, HotDocs Compare is no longer available. A HotDocs add-in tool that is used to compare different versions of an assembled document. Using HotDocs Compare, users can take a "snapshot" of an assembled document, change some answers in the interview, and then compare the two versions.
- **HotDocs Database Connection:** A tool that provides the mapping needed to connect templates to a database. Answers can be retrieved from the database during the interview, which keeps users from manually having to enter their answers. (Starting with the release of HotDocs 2008, HotDocs Database Connection (the separate product) was fully integrated into all editions of HotDocs.)
- **HotDocs Debugger:** A tool used to systematically step through a script or template in order to examine how the script or template produced the result it did. The Debugger is typically used when the

result is unexpected—it helps a template developer pin-point exactly what is happening so he or she can correct the problem creating the error.

- **HotDocs Developer:** A version of HotDocs that contains the tools necessary to automate a simple to highly complicated set of both text and form templates. It is also used to assemble both text and form templates. (See also HotDocs Developer LE.) Prior to the release of HotDocs 10, Developer was named Professional Edition.
- **HotDocs Developer LE:** A version of HotDocs that contains the tools necessary to automate a simple to moderately complicated set of text templates. HotDocs Developer LE can also be used to assemble text and form documents. (See HotDocs Developer.) Prior to the release of HotDocs 10, Developer LE was named Standard Edition.
- HotDocs Filler: The application used to view assembled form documents.
- **HotDocs Options:** A section of the software where template developers and end users can set their preferences for working with HotDocs.
- **HotDocs Outliner:** A tool that generates an outline of scripting in a Word template. This outline can include just a list of instructions used in the template, or it can include variables as well. Items in the outline appear in the same order as they are used in the template. Viewing this outline may help developers better understand the logic used in the template.
- **HotDocs PDF Advantage:** A HotDocs add-in that allowed the creation and automation of PDF-based form templates, as well as the ability to save most types of documents as PDF (assembled or otherwise). From HotDocs 10 onwards, this is no longer a separate add-in, but is instead fully integrated into HotDocs.
- **HotDocs Player:** A version of HotDocs that is used for assembling published (and registered) templates. HotDocs Player functions differently to HotDocs User, which can assemble any template created by HotDocs Developer. For more information, see HotDocs User.
- HotDocs Player Edition: A version of HotDocs that is used for assembling published (and registered) templates.
- HotDocs Professional Edition: Please see HotDocs Developer.
- **HotDocs Server:** The web-based version of HotDocs. Interviews generated by HotDocs Server are presented in a user's web browser. Answers are then sent back to HotDocs Server so final documents can be assembled by merging the interview answers with the non-changing template text. HotDocs Server enables users to create documents and answer files without requiring them to install HotDocs on their desktop.

HotDocs Standard Edition: Please see HotDocs Developer LE.

HotDocs User: A version of HotDocs that is used for assembling any templates created using HotDocs Developer. HotDocs User functions differently to HotDocs Player, which can only assemble registered templates. For more information, see HotDocs Player.

I

IF instruction: A set of instructions and expressions that control the inclusion and exclusion of optional text in a document. IF instructions are based on either True/False variables or true/false

expressions. IF instructions can also be used to control whether certain instructions or expressions are processed in computation or dialog scripts.

- **import:** When working with libraries, the process of copying template files into the currently viewed library. These files can be imported for assembly only or for editing and assembly. When working with answer files, the process of copying an answer file to the default Answers folder and then adding it to the answer library.
- **inactive field:** A form document field on which the user cannot directly enter an answer. Fields can be inactive for any number of reasons. For example, the field may be conditioned or it may contain a Computation variable. Frequently, a template developer provides an answer wizard to help the user answer all of the questions that will make the field active.
- **infinite loop:** The process of a HotDocs script repeatedly reprocessing itself until HotDocs stops responding. For example, a computation can repeatedly scan a text string, character by character, for a specific value. As HotDocs searches for this value, it adds information to what is called the processing stack. If too much information gets added to this stack, HotDocs may get into an infinite loop and stop responding.
- **INSERT instruction:** An instruction that inserts one template into another. For example, if boilerplate text needs to be used in multiple documents, a template An instruction that inserts the contents of one template into another. For example, if you have boilerplate text you need to use in multiple documents, you can create a template that contains that text and use the Insert intruction in each template that requires the boilerplate text to insert that content. This way, if you need to make a change to the text, you only need to update one template.

inserted dialog: See child dialog.

- **inserted template:** A template whose contents you insert into another template's document assembly process using an INSERT instruction.
- **instant update:** A command in the HotDocs assembly window that, when selected, updates the interview every time a user enters or changes an answer in the interview. Sometimes this updating may cause HotDocs to behave sluggishly as users move between answer fields. In such cases, the user can turn the instant update command off. Then HotDocs will update the interview only as it needs to.
- **instruction:** A command in a script or template that performs a special task, such as inserting a template or asking a dialog at a specific place in the interview.

intake interview: See interview template.

- **interview:** A series of dialogs, each containing one or more questions which HotDocs generates from the variables you add to your template. The template's user must answer the questions in the interview before HotDocs can assemble a custom document. You can view the interview by clicking the Interview tab of the assembly window. The left pane of the assembly window displays an outline of the interview. As the user answers questions, the outline tracks the user's progress through the interview.
- **interview component:** A computation script that defines how a custom interview will look and function. An interview component usually includes ASK instructions to ask all of the dialogs/variables in the interview. The script frequently includes other instructions, such as REPEAT instructions and INSERT instructions as well as conditions for using these instructions. The name of this component is defined at the Component File Properties dialog box.

- **interview outline:** The leftmost pane of the assembly window that lists all of the dialogs in the interview. Viewing the outline shows the natural progression of the interview. Icons in the outline also indicate whether questions in the associated dialog are completely answered, partially answered, or not answered at all.
- **Interview tab:** A tab of the assembly window that shows the outline of questions in the interview as well as the dialogs that contain the questions. Users enter answers while viewing the Interview tab. (See also interview.)
- **interview template:** A template that contains a series of interview questions designed to gather information about a person (or persons) or matter. Answers are saved in an answer source file, which can then be linked to a dialog in a template that requires the same information. Generally, interview templates can be used to create a list of possible answers so users have more options to choose from

iteration: One instance of a repeated dialog.

J

JS files: Stands for JavaScript files, which are used to display interviews in a Web browser. When templates are published for use with HotDocs Server, HotDocs generates these JavaScript files for the interview.

Κ

keywords: A broad term used to describe scripting instructions, expressions, and operators. Keywords are used in a script and generate values or perform certain tasks.

L

- **label:** In a text template, an identification assigned to a REPEAT, IF, or SPAN instruction to help the template developer identify the instruction in relation to other instructions in the template. In a form document, the text that is merged in a field when an answer overflows and is sent to the addendum. (See also reference.)
- **layout:** The placement of variables in a dialog. This appearance is controlled at the Layout tab of the Dialog Editor.
- **library:** A window used to display and organize templates. The library does not store the actual files instead, it contains shortcuts (or links) to the files, which are stored on disk. In addition to the template library, HotDocs also uses an answer library, which is more commonly known as Answer File Manager.
- **LIMIT instruction:** An instruction that controls the number of iterations in a repeated dialog. For example, if a dialog should be repeated only four times, a repeat LIMIT of 4 can be assigned to the dialog.
- **line break:** A code in a Word document that indicates that text should appear on a new line within the same paragraph. For example, if the user must enter separate lines in a single paragraph (such as lines in an address), a line break should be used. (See also paragraph mark.)

linked field: Represents a HotDocs field in a fillable PDF template that is associated with an Adobe fillable field. By creating this association between a HotDocs field and a fillable field, template developers can create HotDocs fields that precisely match fields in the underlying PDF. Users who assemble the document can then edit answers associated with linked/fillable fields in the saved PDF.

list: Two or more answers to one question merged in the document.

Μ

manual index: See explicit index.

map, mapping: See variable mapping.

- **mark up, markup:** The formatting applied to a Word template or an assembled Word document that shows simplified template development marks. For example, when viewed in Markup View, variables in a template appear between brackets rather than chevrons. (See also Markup View.)
- Markup View: A view that shows a simplified version of a Word template or an assembled document. This simplified view may be useful if a non-HotDocs user must review the template or document. When viewing a template or document in Markup View, variable and answer fields are marked using brackets.
- merge: When HotDocs pulls an answer from the answer file into the document during assembly.
- **merge field:** During template development, the place in the template where a variable is inserted. During document assembly, the place where the user's answer will be inserted.
- **merge text:** The text that will be merged in a document if a user chooses a specific Multiple Choice variable option. For example, if a user chooses Male as the option, a masculine pronoun such as he or his can be merged instead of Male.
- **model:** A tool in the script editor that template developers can use in writing scripts. A model shows the full instruction or expression—including any placeholders that must be replaced for the script to work correctly. Developers can drag these models from their respective lists and then replace the placeholders with the appropriate values.
- Multiple Choice variable: A type of component that merges a predefined answer in the document.

Ν

- **navigation bar:** In an interview (at the Interview tab), the toolbar used to move from dialog to dialog. In a document (at the Document tab), the toolbar used to move between merged answers in a document.
- nested repeat: A repeated dialog that is nested within another repeated dialog.
- **non-breaking space / hyphen:** A property that can be assigned to a variable that keeps the answer from being split across two lines in the assembled document.
- **notation:** An identification assigned to a variable name to help identify what type of variable it is. For example, Client Name TE would indicate that the variable is a Text variable. (Typical component notations include TE (Text), DA (Date), NU (Number), MC (Multiple Choice), TF (True/False), CO (Computation), and DI (dialog).)

Number variable: A type of component that merges a numeric value in the document.

0

- **ODBC:** Short for Open Database Connectivity, it's a data presentation layer that lets HotDocs communicate with a database so HotDocs can retrieve data from it and use it to assemble a document. (See also ADO.)
- **operator:** A symbol or word that causes either an operation (such as addition) or a comparison to be performed in a computation script or expression.
- **order:** The process of designating the sequence in which form template fields are asked in the tab order. Establishing this order in a form is important for users who directly fill the form document.
- outline: See interview outline.
- **overflow:** Answers in a form document that do not fit in the allotted field space. Overflowing answers are usually sent to the addendum.
- **overlay:** The process of using the Overlay Answers command to merge existing answers into the current answer file. When answers are overlaid, the answers become a part of the current answer file. They also overwrite any existing answers in the interview.

Ρ

- **paragraph mark:** A code in a Word document that indicates that text following the mark should appear in a new paragraph. (See also line break.)
- parent dialog: A dialog that contains a child dialog.
- **pattern:** Determines how a Text variable will be displayed and formatted in the interview and in the assembled document. By default, HotDocs includes three patterns in all new templates (Social Security number, telephone number, and time of day), but template developers can create custom patterns.
- **PDF:** File name extension that designates that the file is a Portable Document Format file, a format created and supported by Adobe. PDFs are a useful way of distributing documents in a format most users can view—as long as they have Adobe Acrobat, Adobe Reader, or HotDocs Filler installed. With HotDocs, template developers can also create PDF-based form templates. They can also create PDFs from assembled documents.
- PDF417: The two-dimensional bar code format used in HotDocs Automator and HotDocs Filler.
- **Personal Information variable:** A type of component that stores basic information about a user, such as a name, a company name, and a phone number. This information is saved in the Current User key of the Windows System Registry. Once answered, users won't be prompted to enter it again.

pick list: See answer source.

placeholder: A marker in an instruction or expression model that indicates where a value must be substituted. This value must be a literal value or a variable. Instruction and expression models help the developer use the correct syntax in a script.

- **pointed component file:** When sharing components across multiple templates, represents the template's own component file, which, in turn, points to the shared component file.
- **pop-up interview:** A dialog a user can display during an interview. Usually a pop-up interview shows a different view of the dialog. For example, if a user is entering answers in a spreadsheet, he or she can click the Edit Row button and a pop-up interview appears that shows just the questions (and answers) from that particular row in the spreadsheet.
- **printer driver:** A driver that generates form templates from files designed in other applications, such as a draw program or a word processor. HotDocs includes one printer driver—the HotDocs PDF driver.
- **processing stack:** A sequential list of templates and components HotDocs is processing at any given time. Each time a new component is processed, it is added to the stack. (Once processing is finished, it is removed.) In some instances where recursion is used in a script, the same component is repeatedly added to the list. If the number of components exceeds the stack limit, an infinite loop error will occur. (The stack limit can be changed at the Component File Properties dialog box.)
- **prompt:** Text that can be assigned to a variable to help the user better understand how to answer the question.
- **publish:** The process of generating files for distribution to other users. Types of files that can be published include auto-assemble files, auto-install files, regular templates, and HotDocs Server files. During publishing, protection options can be assigned to help safeguard the templates.
- **Publishing Wizard:** The tool used to publish template files for others. The Publishing Wizard steps through the publishing process and allows different properties to be set, depending on the type of file(s) that are being published.
- punctuate: The process of formatting a REPEAT instruction so that a list of answers will appear in sentence format, like this: The client owns real estate in New York, Pennsylvania, and Montana. (New York, Pennsylvania, and Montana are the list items. The punctuation adds the commas and the conjunction and).

Q

- **Question Summary:** A brief report HotDocs generates that lists questions asked during an interview. The summary includes blank lines for handwritten answers. (See also Answer Summary.)
- queue: See Assembly Queue.

R

- **reference:** In a form document, the text that is added to the addendum to identify any overflow answers. (See also label.)
- **reference path:** A folder path for a template in which the drive letter and some or all of the folder names are represented by a keyword. At runtime, this keyword is mapped to an actual path on the user's computer so that when the user accesses the template, the keyword is replaced by the path. This allows templates saved in one central location to work on multiple workstations regardless of how the drives on the workstation are mapped.
- register: The process of publishing a template for use with HotDocs Player.

- **REPEAT instruction:** An instruction that repeatedly asks the same variable(s) so that users can provide two or more answers for a given question. REPEAT instructions are used to create lists of answers in the document.
- **repeated dialog:** A dialog that contains the variables that need to be repeated so that multiple answers can be entered. (See REPEAT instruction.)
- **repeated series dialog:** One of two representations of a dialog that is repeated. With a repeated series, the dialog is asked repeatedly until all answers in the list have been entered. (See also spreadsheet dialog.)
- **resource:** Supplemental help that can be included with a variable or dialog to help users better understand how to answer the questions they are viewing. Resources appear in the resource pane of the assembly window.
- **Resource field:** A type of form template field that functions like a hyperlink. Link fields let users get help while direct-fill assembling the form document. Links are displayed as underlined, colored text, much like links in a Web browser. They provide a resource for the form in general, instead of for a specific variable or dialog.
- RTF: File name extension that designates that the template file is a Word RTF file. (See text template.)
- **run-on group:** A series of form template fields that have been grouped and associated with each other so that the answer can flow from one field to the next. For example, if an answer must fill two or more lines on a form, separate fields can be created on each line and then grouped. Once grouped, they become a run-on group.

S

- **script:** One or more instructions and/or expressions that generate a value or execute some kind of procedure.
- **script editor:** The tool used to write a script. The script editor includes several options to make the scriptwriting process easier, including color-coding, auto-complete lists, and a toolbar for completing other tasks.
- **selection grouping:** A dialog property assigned to True/False variables, clauses, and child dialogs which presents these options as check boxes (multiple-select) or option buttons (single-select).
- **Send to Word Processor command:** A command that opens the word processor and copies the assembled document into it. Once opened in the word processor, the user can make any changes necessary to the document.

set:

- **SET instruction:** An instruction in a template or script that assigns a value to a variable. Variables that have their values set should not be asked again in the interview.
- shared component file: A common component file to which several related templates are linked. To use a shared component file, the template's own component file must be pointed to the shared file. Changes to components in the file are reflected in all templates that use it. (See also pointed component file.)

- **SHOW:** An instruction used in a dialog script to show variables that have been hidden in the dialog. (See also HIDE.) Usually this instruction is conditioned so that variables hide and show dynamically, based on answers the user enters.
- **sort:** The process of alphabetizing answers in a repeated list or items in a library. Sorting can be done in ascending or descending order.
- **spreadsheet dialog:** One representation of a repeated dialog. Each row in a spreadsheet represents one repetition in a dialog. (See also repeated series dialog.)
- **spreadsheet-on-parent dialog:** A repeated child dialog that appears as a spreadsheet on its parent dialog. This allows the user to view both dialogs at once.
- **static text:** The underlying text in a form template or document that does not change. To enter answers on a form, form fields must be created and overlaid on the static text.
- strike-through field: A type of form template field that is used for crossing out static text on the form.
- summary: See Question Summary and Answer Summary.
- **supplemental component:** A term used to define components such as patterns, example formats, dialog elements, and merge text. Supplemental components are associated with regular components, but they can be created and edited as standalone components.
- **syntax:** The language used in writing scripts. For a script to work properly, the script must be written in a way that HotDocs can understand. This language consists of instructions, expressions, operators, and values (such as text, numbers, dates, or answers users enter).

Т

- **template:** A word processor or form document that has been converted to HotDocs format so that it can be automated. When in template format, changeable text in the template can be replaced with variables. Other instructions can be added as well, such as instructions that create lists, condition text, and insert other templates.
- **template developer:** The person responsible for automating your firm's templates. The template developer creates and inserts the variables in the template, arranges variables in dialogs, and performs other custom tasks in the template. (See also user.)

template development: See automate, automation.

- **Template Manager:** The tool used to manage a library of templates and their associated components. This includes converting or renaming templates, copying and pasting components across multiple templates, renaming components, and changing component file properties for multiple templates.
- test: The process of testing a variable or other component to make sure it looks right and works correctly.
- **test assemble:** The process of assembling a document for the purpose of ensuring the interview works correctly and the automation within the template produces a correctly assembled document. During a test assembly, developers can easily edit components and have the test assembly window updated with changes.

- **text document:** A document that is viewed in either Word or WordPerfect. It can represent a document before it is automated as well as a document after it has been assembled. When in document format, it is not associated with (or linked to) HotDocs in any way. (See also text template.)
- **text template:** A template that is created and automated in Microsoft Word or WordPerfect. It is distinguished from a form template by the fact that the underlying text of the template can be modified. (See also text document.)
- Text variable: A type of component that merges text in the document.
- **thumbnail:** Small images of each page in the form template. Developers can use thumbnails as a way to see an overview of the template. Thumbnails can also be used to move around quickly in a form.
- **title:** A property of a variable or dialog that specifies a more user-friendly name for the component. For example, if project standards require components be named using notations, names like Employee Name TE may not make sense to a user. However a title like Employee Name can be used instead.
- **True/False expression:** A script that must result in either true or false. Expressions are used for merging or excluding optional text in a document. They are also used for determining which parts of a script will be executed, based on answers or other values entered by a user. Expressions are often used when a simple True/False variable doesn't convey the condition needed. (See IF Instruction.)
- **True/False variable:** A type of component that determines a true/false status of some condition and then merges the appropriate answer or text.

U

- **unanswered text:** Text in a text document that indicates that a question is unanswered. By default, unanswered questions appear as ***Variable Name***, but this can be customized.
- **UNGRAY:** An instruction used in a dialog script to enable a variable that has been grayed based on a user's answers. (See also GRAY.)
- **Unicode:** Computer specification that makes it possible for computers to represent and manipulate characters used in most of the world's written languages. Unicode support in HotDocs makes it possible to automate and assemble documents in non-native, left-to-right-reading languages. This includes automating and assembling Microsoft Word templates as well as PDF-based form templates.
- **upload:** The process of moving HotDocs files to a server so they can be accessed via an intranet or Internet site.
- **user:** The customer, client, or person who assembles documents from templates. (See also template developer.)

V

value: In an interview, it represents a user's answer. In a script, it represents data that must be used in executing the script. (The value can either be a literal value or a user's answer.)

- **variable:** A component that is used to represent changeable text (such as names, dates, numbers, etc.) in the template. Types of variables include Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information.
- variable field: A place in the template that denotes where an interview answer merges with the template's non-changing text. At assembly, HotDocs merges interview answers into the variable fields that represent corresponding variables. In a text template, a variable field is enclosed by chevrons.
- variable flow-through: See answer sharing.
- **variable mapping:** The process of associating two HotDocs variables so that they can share answers. In some cases, this mapping defines the relationship between a HotDocs variable and a field in a third-party application file, such as a database table or a field in an Outlook Contacts list.

W

- **warnings:** A list of cautions that appear in the test assembly window that indicate problems created by scripting in the template.
- wizard: See answer wizard.
- WPD: File name extension that designates that the file is a WordPerfect document. (See text document.)
- WPT: File name extension that designates that the file is a WordPerfect template. (See text template.)

Х

XML: Stands for eXtensible Markup Language. It is a computer language designed to store and transmit data between applications. Like HTML (HyperText Markup Language), it contains customized markers, or tags, that identify the information in an XML file. However, while HTML describes the way a page looks, XML controls the way data is structured, making it easy for diverse programs to access the same information. (For example, in HTML, to indicate a book title, you would italicize it using the <i> tag. In XML, you could mark the title using a <body>

booktitle> tag. The HTML tag simply formats the text (making it italic), while the XML tag actually defines what the text is (a book title).) In HotDocs, you can save libraries, component files, and answer files in XML format.

Index

/

/af 1135 /as 1135 /cl 1135 /da 1136 /df 1136 /ed 1137 /ex 1137 /fia 1137 /ha 1138 /hi 1138 /hl 1138 /is=u 1139 /kig 1139 /la 1140 /lf 1140 /|| 1141 /mo 1139 /na 1141 /ni 1142 /nw 1141 /of 1142 /ov 1143 /pa 1145 /pb 1146 /pc 1146 /pd 1146 /po 1147 /pr 1145 /ps 1144 /pt 1145 /pw 1147 /qs 1148 /sa 1148 /sap 1148 /sig 1150 /sl 1149 /ss 1150 /ssn 1151 /stw 1149 /sw 1151 /tf 1151

1

1900-century years 212

2

2000-century years 212

А

A, B, and C lists formatting in REPEAT instructions 349 formatting in scripts 712, 850 Abbreviated icons in component lists 263 About answer file uploading 1035 answer files 181 answer management 186 assembling a form document 134 clauses and clause libraries 388 command-line options 1127 conditional text 328 database components 927 dialogs 551 dot codes 959 fillable PDFs 476 HotDocs Automator 405 HotDocs Filler 135 HotDocs Models 1041 inserted templates 360 interview process 97 libraries 65 lists within lists (nested repeats) 609 publishing templates 991 SPAN fields 385 Template Manager 664 templates and component files 289 Test Panel 986 testing variables and assembly 971 uploading HotDocs files 1025, 1031 using files located on the Internet 193 variables 487

ABSOLUTE VALUE(NUM) 734, 872 Accelerators, assigning to characters in a prompt 513 Accessing Answer File Manager 188 answer files 186 HotDocs files on the Internet 193 HotDocs libraries 69 records in a database 940 records in a database using custom command 945 ActiveX Data Objects, used in Database Connection 933 Add Another button customizing 346 Add Answer File button 189 Add Item button 71 ADD TEXT TO MULT_CHOICE 703, 840 Addendum changing properties of at form 172 changing properties of, using Filler 172 creating default references to 246 customizing text within 475 numbering codes to use in 475 specifying default properties of 250 splitting answers between form and 176 starting on new page 474 Addendum label text defining for regular field overflow 174, 466 defining for run-on field overflow 467 defining for table overflow 468 Adding answer files to Answer File Manager 189 auto-assemble files on the Internet to HotDocs libraries 195 auto-install files to libraries 196 buttons to dialogs 576, 577 clause library to a template 402 dialogs to dialogs 589 existing clauses to clause library 398 files to published template set 1009 folders to clause library 394 folders to the answer library 192 multiple items to a library 71 new clauses to clause library 396 options to Multiple Choice variable 703, 840 pages to a form template 415 prompts to variables 513

resource information to variables 520 scripts to dialogs 581 templates to libraries 71 text to a custom dialog 575 Web pages to HotDocs libraries 199 Adding (values) number of days to Date variable 741, 879 number of months to a Date variable 742, 880 number of years to Date variable 742, 880 repeated number values 773, 911 Additional Text creating 575 placing variables in 532 Adjusting toolbars 95 width or height of form field, Automator 430 width or height of form field, Filler 152 windows and panes sizes 95 ADO connections to database, specifying 933 Adobe automatically send assembled forms to 236 saving assembled form documents to 130 saving text documents to 130 using for fillable PDF templates 476 using to print assembled form documents 238 Advanced options for variables 515, 516, 517 AGE(DATE) 735, 873 Alerting when editing shared component file 267 when template sets have updates 1010 Align Fields button, Automator 430 Align Fields button, Filler 152 Aligning answers left or right justified 517 form fields, Automator 430 form fields, Filler 152 prompts and answer fields in dialogs 564 text in edit fields 241 text in multi-line form fields 242 Allowing users to upload answers 1036 Alphabetizing answers in a REPEAT field 350 answers in a spreadsheet 112 answers using a script 704, 842 lists of answers using a computation 606 Always stop at this dialog when moving to next unanswered option 564

American dates, using 211 Ampersands, using to specify accelerator keys 513 Annotations, using to mark up templates 227 Answer field widths, specifying 516 Answer File command-line option 1135 Answer file folder, changing location of default 279 Answer File Manager about 186 adding answer files to 189 adding folders to 192 creating new answer files using 181 displaying 188 setting option for always using 282 using 190 Answer File Manager button 188 **ANSWER FILE NAME variable 509** Answer file uploading, about 1035 Answer files about 181 accessing 188 adding to answer library 189 assembling documents with 99, 111, 181, 183 attaching to e-mail messages 185 automatically prompting user for 222 automatically suggesting the answers used last 99 creating 111, 181, 183 default 182 deleting 190 editing titles or descriptions 190 enabling for uploading 1036 linking to specific templates 182 managing 190 merging answers into another answer file 101 overlaying 101 searching for 190 suggesting for every assembly 183 uploading 1035 uploading to Web servers 199 using for testing 975 Answer Font button, Automator 452 Answer Font button, Filler 160 Answer sources for dialogs mapping to an Outlook Contacts list 587 mapping to Time Matters records 588 sharing answers between 586

suggesting for dialogs 583 tips on using 584 Answer summaries changing appearance of 253 viewing 179 Answer Summary command-line option 1135 Answer wizards, attaching to form fields 441 ANSWERED(DIALOG) 736, 874 ANSWERED(VAR) 736, 874 Answer-field reformatting 211 Answer-gathering dialogs, navigating 106 Answering Date variables 116 questions in an interview 106 questions to create Question Summary 254 Answers about 181, 186 applying standard information 182 assembling documents with 99 assigning conditionally to form fields 426 checking for any that overflow 171 creating lists of in forms 462, 463 editing at Document Preview tab 121 flowing across two or more form fields, Automator 435 flowing across two or more form fields, filler 157 flowing across two or more form pages, Automator 465 flowing across two or more form pages, Filler 169 for personal information stored in registry 508 gathering information to assemble other documents 100 loading automatically using default answer file 182 organizing 192 overwriting with another answer file 101 previewing formatting of in form fields 450 saving 112 saving back to a database record 942 saving back to a Time Matters record 588 spell checking 119 too long for answer field 245 typing directly in form fields 136 ANX file name extension 49 Appearance of assembly window, changing 215

of edit fields in forms 241 of fields in a text template 230 Application link, adding to a dialog 576 Apply Colors button 290 Applying colors to fields in a text template 301 Arrange button 398, 617, 789, 974 Arranging clause library window with other windows 398 columns in database table 946 Component Manager with template window 617, 789 contents of dialog 561 items in a library 80 order dialogs are asked 591 order of documents to be assembled 103 windows and panes 95 ASCEND VAR 704, 842 Ascending a list of answers in a REPEAT field 350 a list of answers using a computation 606 a list of database records 950 ASK ALL instruction 593 Ask automatically option for dialogs 571 for variables 515 ASK DATABASE 705, 842 ASK Database instructions 955 ASK DIALOG 705, 843 ASK Field button 290 **ASK** instructions attaching to form template fields 425 using directly in the template 591 using in a computation 592 ASK NONE 593 ASK VAR 706, 843 Asking dialogs automatically 571 only unanswered questions 1139 variables answered by database 953 ASSEMBLE 706, 844 Assemble button 97, 974 Assembling about 89 clause libraries 105 creating new answer files while 181 documents 97 documents without showing the interview 100 fillable PDFs 139

from Web browsers 193 headers and footers 530, 663, 821 information from a database application 200 interview templates 100 multiple documents 102 multiple documents from one template 381 previewing templates before 96 switching answer files while 112 templates using HotDocs Server 1020 using the Assembly Queue 103 warn when adding more templates to gueue 224 Assembly Queue about 103 always listing completed assemblies 225 assembling multiple documents 102 displaying warning when adding to 224 Assembly Queue button 103 Assembly window bypassing to view assembled document 100 closing after assembly session 117 customizing the look of 215 Assigning answer files to templates 182 form check-box field properties 242 merge text to Multiple Choice options 528 new variable names 638, 808 resource button labels 635, 805 resources to variables and dialogs 520 values to variables using SET instructions 783, 921 variables to form template fields 425 Asterisks using as placeholders for unanswered questions 223 using to mark required questions 215 Attaching answer files to e-mail messages 185 ASK instructions to form template fields 425 INSERT instructions to form template fields 427 Multiple Choice variables to form fields 438 REPEAT instructions to form template tables 462, 463 text or form documents to e-mail messages 177 variables to fillable fields 482 variables to form template fields 425

Austrian German templates, creating 311 Auto Format button, using 506 Auto-assemble files adding Internet link to library 195 creating 996 downloading files for use offline 195 MIME types information 193 updating 195 Auto-complete functionality use as you type component names 262 use as you write scripts 269 Auto-install files accessing from the Internet 193 automatically updating 1010 creating 999 installing a template set 196 MIME types information 193 Automatic interviews, generating 601 Automatically aligning dialog prompts 564 checking for updated template sets 198 disabling irrelevant variables in an interview 601 expanding child dialogs in interview outline 208 filling in answers for users 783, 921 generate names for dialog elements 262 merge filtered database record into document 952 numbering paragraphs 509 recovering form template files 239 selecting records based on selection of other records 953 selecting text when tabbing between fields 209 updating published template sets 1010

В

Background colors changing for dialog pane 215 changing for form fields 236 Backing up form files 239 shared component files 651, 813 Bar codes displaying in forms 167 displaying in forms in Automator 460

displaying in text template fields 517 Basic user information, specifying 508 Birth dates, using to calculate ages 735, 873 **BMPs** adding to dialogs 580 assigning to library splash screens 1010 including in published template sets 1009 inserting in form fields 165 inserting in form fields in Automator 459 Bookmarks, creating in text templates 656, 818 Boolean operators, using 687, 825 Boolean operators, using in Models 1095 **Borders** changing properties of field 153 changing properties of field in Automator 431 changing the size of detectable 242 detecting to create or resize fields 150 detecting to create or resize fields in Automator 428 Brands of word processors, changing 298 Brazillian Portuguese templates, creating 311 British dates 211 Browsers launching from HotDocs 199 registering HotDocs as a helper application for 193 viewing assemblies in 1020 Buttons customizing layout of toolbar 95 viewing resource 210 word processor toolbar 290 **Bypassing** End of Interview dialog to view assembled document 217 entire interview to view document 100

С

Caches, updating versions of auto-assemble files 195 Calculating answers 506 Calendars, using to answer date questions 116 Canceling assembly from End of Interview dialog 117 assembly of a document 103 script processing 721, 859 Catalog files, using to update published template sets 1010 CDs, referencing templates on 1007 CEILING(NUM) 737, 875 Centimeters as a unit of measure in forms 235 Century-rollover dates 212 Changing a form field's size 155 a form field's size in Automator 433 a form field's type 159 a form field's type in Automator 451 appearance of variable names and background colors for fields 450 component file properties 635, 805 contents of a text template 300 dialogs on the fly 582 dialogs shown in interview outlines 96 file location for templates in a library 87 file location of answer files 189 fonts used for form fields 160 fonts used for form fields in Automator 452 form field borders and margins 153 form field borders and margins in Automator 431 order of documents awaiting assembly 103 overflow properties for regular form fields 174 overflow properties for regular form fields in Automator 466 overflow properties for run-on form fields 467 overflow properties for table fields 468 paragraph formatting in a form field 161 paragraph formatting in a form field in Automator 453 properties for multiple items in library 78 rotation of a form field 162 rotation of a form field in Automator 454 tab order of form fields 156 tab order of form fields in Automator 434 variable names 638, 808 word processor brands 298 word processor file locations 276 Characters, using specific for check-box fields 242 Check Fillable fields command 480 Check Spelling button 119 Check-box fields attaching Multiple Choice variables to 438 changing field type to 159 changing field type to, in Automator 451 creating 149

creating in Automator 424 detecting 242 specifying properties for 242 Checking form fields for answers that overflow 171 spelling of answers 119 spelling of components 648 templates for Web server compatibility 1019 variables for accuracy 972 Chevrons, HotDocs 532 Child dialogs assigning in nested repeats 610 controlling appearance in interview outline 208 creating 589 displaying on parent when repeating 611 grouping multiple in parent dialog 590 removing icon for 564 Choosing answer files for assembly 99 database records during assembly 200 End of Interview options 217 Circling answers on a form 165 Circling answers on a form in Automator 458 Clause archive file 392 Clause component files 392 Clause field, editing 403 Clause identifier field, using 392 Clause libraries about 388 adding clauses to 398 adding folders to 394 assembling 115 creating at template library 393 creating clauses and adding to 393 creating in template and adding to 400 editing 398 files used in 392 inserting into template 402 moving on local disks or networks 87 opening and closing 393 selecting clauses from 115 Clause Library button 290 Clause Name command-line option 1135 Clauses about 388 adding to library 398 assembling 105

creating and inserting in template 401 creating at clause library 396 creating in template and adding to clause library 400 displaying selections in one dialog 403 editing 403 inserting into template 400 underlying functionality 392 CLEAR MULT CHOICE 703, 840 Clearing answers from a Multiple Choice variable 703, 840 answers from variables 779, 917 variables from component files 636, 806 Client cursors, using to connect to database 955 Closina assembly window after interview 117 clause libraries 393 component editors warning 267 Component Manager 617, 789 form templates 420 Template Manager 668 CMP file name extension 49 Codes dot 959 LANGUAGE instruction 311 text patterns 531 Collapsing all folders in a library 72 child dialogs in outline 114 Color themes applying to text template fields 301 changing default for dialog pane 215 changing default for form fields 236 changing default for text template fields 230 Columns of listed answers 357 specifying widths of database tables 946 specifying widths of in repeated spreadsheets 516 using in summaries 253 Command-line options about 1127 Answer File 1135 Answer Summary 1135 Clause Name 1135 Default Answer File 1136 **Discard Answers 1136**

Edit Template 1137 Exit HotDocs 1137 Finish Interview Action 1137 full list of 1129 Hide Library 1138 HotDocs Auto-Assemble File 1138 HotDocs Auto-Install File 1138 HotDocs Model 1139 Interview Scope 1139 Keep Interview Group 1139 Library File 1140 Lock Answer File 1140 Lock Library 1141 New Answer File 1141 No Assembly Window 1141 No Interview 1142 Output File 1142 **Overlay Answer File 1143** Paper Size 1144 Paper Tray 1145 Print 1145 Print Answers Only 1145 Print Both 1146 Print Copies 1146 Print Duplex 1146 Print Form Only 1147 Print Without Dialogs 1147 **Question Summary 1148** Save Answers 1148 Save Answers Prompt 1148 Send to Word Processor 1149 Show Library 1149 Start Interview Group 1150 Suggest Save 1150 Suggest Save New 1151 Suppress Unanswered Warning 1151 Template File 1151 Commands in Component Manager 618, 790 in HotDocs Automator 408 in HotDocs editing toolbar 290 in the assembly window 106 in the clause library 398 in the End of Interview dialog 117 in the HotDocs Debugger 981 in the template library 81 Comments adding to component editors 524, 569

adding to scripts 534, 692, 721, 829, 859 adding to variable and instruction fields 525 Common information in templates, gathering 603 Comparing fillable fields with HotDocs fields 480 Comparison operators, using 687, 825, 1095 Completed assemblies, displaying in gueue 225 Completing component names automatically 262 Component database, creating 668 Component file properties changing default dialog titles 635, 805 changing default template titles 635, 805 checking for server compatibility 635, 805 controlling answer file selection 635, 805 defining interviews 600 hiding End of Interview dialog 655, 817 moving cursor to location in assembled document 656, 818 setting for new templates 224 sharing 654, 816 specifying hidden data to remove 659, 819 Component File Properties button 635, 805 Component files about 289 adding to library 603 changing file properties 635, 805 copying components between 636, 681, 807 exploring several at once 664 looking at contents of 618, 790 managing 617, 789 opening 617, 789 opening at template library 618, 790 pointing 651, 813 printing contents of 651 searching 646 sharing information in 651, 813 spell checking contents of 648 stop sharing 655, 817 Component icons, changing appearance of 263 **Component Manager** copying components using 636, 807 creating components using 322 deleting components using 636, 806 editing components using 510 inserting variables into templates using 322 opening and closing 617, 789

opening and closing at template library 618, 790 pointing component files using 651, 813 renaming components using 638, 808 viewing 618, 790 Component Manager button 617, 789 Component names, changing in several files 682 Component types, modifying 642, 813 Components generating list of searched 646 generating list of spell checked 648 grouping into dialogs 561 Compressing published files 996, 999 Computation variables creating 506 debugging 981 executing from a dialog 577 methods of writing 534, 692, 829 testing results of 972 tips for writing correctly 546, 685, 823 using as an answer wizard for field 441 using for ASK instructions 592 using the script editor 534, 692, 829 using to create nested repeats 612 using to create REPEAT instructions 605 CONCEAL VAR 708, 845 Concealing variables in answer sources (pick lists) 708, 845 Conditional text about 328 assigning to form fields 426 creating with True/False expression 333 creating with True/False instruction 333 fully qualifying 338 nesting IF instructions in 337 using in question summaries 254 Connecting specific answers to templates 182 to database using ADO 933 Contacting technical support 1203 Contacts lists, mapping variables to fields in 587 CONTAINS 687, 775, 825, 913, 1095 Contents, specifying template 295 Control fields changing field type to 159 changing field type to, in Automator 451 creating 455 Controlling

appearance of Answer File dialog box 222 order dialogs are asked 591 text placement in multi-line field 242 variables in the interview outline 593 Convert Files button 675, 676 Converting a single template to a new file format 298 answers on a form to bar code format 167 answers on a form to bar code format, in Automator 460 form templates to PDF templates 411 HotDocs Models to HotDocs Templates 1088 HotDocs templates to HotDocs Models 1087 multiple templates to DOCX 676 multiple templates to latest HotDocs version 675 number characters into numeric values 751, 889 PDFs to form templates 410 templates, warn when 267 Copy Components button 681 Copying clause libraries across disks or networks 87 components across multiple component files 681 components between two files 636, 807 files when exporting libraries 82 form fields 154 form fields, in Automator 432 library items to new locations on disk 88 library items to new places within library 86 template files to new locations 88 template libraries from disk or network 84 variables to create new 640, 811 COUNT(DIALOG) 738, 876 COUNT(MULT_CHOICE_VAR) 738, 876 COUNTER 739, 877 Counting answers in a list 738, 876 number of characters in text values 753, 891 repetitions of a dialog 739, 877 selected Multiple Choice options 738, 876 Crashing, protecting form template files from 239 Create New Component button 618, 790 Creating ADO connection string 933 answer files during assembly 111, 183

check-box fields 149 check-box fields, in Automator 424 clause library at template 402 clauses and inserting in template 401 clauses at clause library 396 clauses in template and adding to clause library 400 components using Component Manager 618, 790 conditional text 333, 334 custom dialogs 561 dialog scripts 581 fields using Detect command 150 fields using Detect command, in Automator 428 form fields 147 form fields, in Automator 422 form templates 410 interview templates 603 new answer files 181 new libraries 69 new libraries by exporting existing libraries 82 PDF files 130 pick lists of answers for dialogs 583 questionnaires 178 summaries 178, 179 text templates 295 Crossing out static text on a document 164 Crossing out static text on a document, in Automator 457 Cross-references defining for regular field overflow 174 defining for regular field overflow, in Automator 466 defining for run-on field overflow 467 defining for table field overflow 468 specifying properties of addendum items 246 viewing for components 522, 569 Currency values 493 CURRENT ANSWER FILE, linking different variables in 586 Current open word processor 295 Cursors choosing for database 955 controlling location of in assembled document 656, 818 Custom commands

using with database component 945 Custom dialogs, creating 561 Custom splash screens for libraries 1010 Customizing colors used in dialog pane 215 Computation variables 506 Date variables 496 dialogs 564 entries on a form addendum for regular fields 174 entries on a form addendum for regular fields, in Automator 466 entries on a form addendum for run-on fields 467 entries on a form addendum for tables 468 Multiple Choice options 527 Multiple Choice variables 501 Number variables 493 options in End of Interview dialog 217 options in script editor 269 Text variables 489 the interview for a template 600 True/False variable placement in a dialog 564 True/False variables 498 Cutting a number after a decimal point 778, 916

D

Data source, specifying for database connection 935 Database applications assembling documents linked to 200 supported 927 Database components about 927 assigning keys for 941 choosing which fields will appear 946 editing 939 editing properties of 943 explanation of how they work 928 retrieving data using custom command 945 security 949 storing information for in separate file 937 using OLE DB providers for ODBC with 935 DATE - NUM DAYS 740, 878 DATE - NUM MONTHS 740, 878 DATE - NUM YEARS 741, 879 DATE + NUM DAYS 741, 879

DATE + NUM MONTHS 742, 880 DATE + NUM YEARS 742, 880 Date entry order, specifying 211 Date formats 540 DATE OF(NUM, NUM, NUM) 743, 881 Date variables creating 496 formatting correctly in scripts 733, 871 formatting in foreign languages 311 inserting today's date using 509 ways to answer 116 Dates displaying in answer fields 213 formats for 544 specifying default format for in assembled documents 222 specifying the date order 211 DAY OF WEEK(DATE) 744, 882 DAY OF(DATE) 744, 882 Day portion of a given date, finding 744, 882 Day-Month-Year date order 211 DAYS FROM(DATE, DATE) 745, 883 Days of the week, determining 744, 882 DEBUG 708, 846 Debugging templates and scripts about 977 adding instruction to template or script 980 stepping through template or script 981 watching specific variables during debugging 986 Decimals, using 493 Decreasing values of temporary counters 716, 854 DECREMENT NUM_VAR 716, 854 Default answer files 182 date formats 222 dialog titles 635, 805 form printing options, specifying 145 formats 511 interviews in form templates 248 merge text 501 size of check-box fields 242 template titles and descriptions 635, 805 word processor, changing 274 Default Answer File command-line option 1136 DEFAULT VAR TO VALUE 709, 847 Defining field borders and margins 153

Defining field borders and margins, in Automator 431 Delete Components button 636, 806 Delete Components button, in Template Manager 683 Deleting answer files 190 clauses from a clause library 398 components from current component file 636, 806 folders in Answer File Manager 192 form template pages 417 library items 74 unused components from multiple component files 683 Delivering answer files using e-mail 185 assembled document files using e-mail 177 **DESCEND VAR 704, 842** Descending a list of answers in a REPEAT field 350 a list of answers using a computation 606 a list of database records 950 Descriptions for HotDocs files 65 modifying library item 76 searching for specific templates 97 template 295 Detect Field button 150 Detect Field button, in Automator 428 Detecting fields on a form 150 fields on a form, in Automator 428 setting default properties for check-box fields 242 Developer View button 290 **Dialog** elements automatically generate names for 262 computation buttons 577 executable files 576 graphic files 580 hyperlinks 578 spacing and lines 579 text 575 Dialog navigation, tabbing between fields 209 Dialog pane, customizing 215 **Dialog scripts** about 581

CONCEAL VAR 708, 845 HIDE ALL 713, 851 HIDE VAR 713, 851 LIMIT NUM 718, 856 OMIT VAR 719, 857 **REQUIRE ALL 723, 861 REQUIRE VAR 723, 861** SHOW ALL 713, 851 SHOW VAR 713, 851 Dialogs about 551 accessing resource information while viewing 109 adding an answer source to 583 adding text to 575 asking at specific place in form templates 425 assigning titles to 561 changing layout of 568 changing properties of 564 changing the look of 215 child 589 creating 561 editing 564 grouping clauses in 564 grouping True/False variables in 564 hiding list of during interview 635, 805 navigating through during assembly 106 scripting in 581 seeing which variables are used 522, 569 testing appearance of 973 using to create answer wizard for field 441 Dialogs: 106 Differences between SET and DEFAULT instructions 786, 924 Direct-fill assembly of form documents 136 Directories for word processor files 276 Discard Answers command-line option 1136 Displaying Answer File Manager 188 child dialog on parent 611 database fields during assembly 946 forms in multiple windows 418 only selected database fields during assembly 946 Distributing templates on the Internet 996, 999

Division remainders 766, 904 DLLs, foreign language 311 Document assembly canceling 103 multiple 102 using clause libraries during 105, 115 using overlay answer files for 101 viewing list of gueued assemblies 103 viewing status of current 103 **Document Managers** setting up to work with answer files 282 Document properties, removing from assembled document 659, 819 **Document Text Editor** using 125 Documents attaching to e-mail messages 177 double-checking answers in 120 enabling all new templates for document editing 265 previewing assembled text 120 printing assembled text 129 printing form 143 saving 129 saving as PDF files 130 sending to word processor 127 viewing assembled form 139 Documents into templates, converting 295 DOCX file name extension 49 DOCX preferred over RTF, why 289, 676 Dollar amounts 493 Dot Codes about 959 using to capitalize text 965 using to change font properties of text 962 using to insert a hyperlink 964 using to insert characters in text 966 using to merge punctuation 965 using to punctuate non-repeated lists 968 DOT file name extension 49 Double-angle brackets around variables 532 Double-sided forms, printing 145 Downloading auto-assemble files for use offline 195 foreign language DLLs 311 HotDocs files 193 updated template sets 1010

Dragging variables from Component Manager into template 322 Driver, HotDocs PDF 410 Duplex printing for forms 145 Duplicate Variables button 640, 811 Duplicating form fields 154 form fields, in Automator 432 variables 640, 811 Dutch templates, creating 311 Dynamically adding options to Multiple Choice variables 703, 840 changing appearance of variables in dialogs 581 changing the contents of a document 328

Е

Edit Component button 290 Edit Document Text button 125 Edit fields (forms) changing field type to 159 changing field type to, in Automator 451 specifying properties of 241 Edit Template button 300 Edit Template command-line option 1137 Editable text creating sections of in a template 386 modifying 125 Editing answers at Document Preview tab 121 answers at the Document tab (allowing) 635, 805 appearance of dialogs 568 as you test assemble 974 assembled form documents 146 clause libraries 398 clauses 403 components using Component Manager 618, 790 custom dialogs 564 database component properties 943 database components 939 form templates 414 IF instructions 336 **INSERT** instructions 366 interview templates 603

library item properties 76 multiple variables at once 620, 792 Personal Information variable answers 257 properties in Answer File Manager 190 **REPEAT instructions 347** supplemental components 642, 813 templates 300 text in the assembled document 386 variables 510 Editing document text allowing users to 386 at the assembly window 125 ELSE IF and ELSE 333 E-mail attaching answer files to 185 attaching text or form documents to 177 Embedding dialogs 589 IF instructions 337 repeated dialogs 610 Enabling answer editing in a text document 265 Enabling answer editing in a text document, using component file properties 635, 805 End of Interview dialog bypassing to display assembled document 217 choosing which options appear in 217 hiding 117 using options at 117 Enter a Date 733, 871 Enter a Number 733, 871 Enter key using in multi-line text fields 106 using in scripts 691, 828 using to move between dialogs 209 Enter some Text 734, 872 Entering answers directly in a form field 136 ERASE VAR 710, 848 Error messages in templates and scripts 976 locating specific fields that generate 306 when starting Template Manager 668 **ERRORTEXT 746, 884** Example formats for dates 544 number 526 understanding how they're used 540 Excluding variables from answer sources (pick lists) 719, 857

Executable files, adding to dialogs 576 Exit HotDocs command-line option 1137 Exiting invalid computations 721, 859 **Template Manager 668** the assembly window 117 Expanding all folders in a library 72 child dialogs in interview outline always 208 child dialogs in outline 114 file lists in Template Manager 669 Expiring templates, setting properties for 1004 Explicit indexing answers in a list 359 answers in a sublist 613 Explorer, opening for a selected template 72 EXPONENTIAL(NUM) 747, 885 Exponents in calculations 762, 900 Exporting a Template Manager list to a spreadsheet 673 Exporting existing library to create new libraries 82 Expression models ABSOLUTE VALUE(NUM) 734, 872 AGE(DATE) 735, 873 ANSWERED(DIALOG) 736, 874 ANSWERED(VAR) 736, 874 CEILING(NUM) 737, 875 COUNT(DIALOG) 738, 876 COUNT(MULT CHOICE VAR) 738, 876 COUNTER 739, 877 DATE - NUM DAYS 740, 878 DATE - NUM MONTHS 740, 878 DATE - NUM YEARS 741, 879 DATE + NUM DAYS 741, 879 DATE + NUM MONTHS 742, 880 DATE + NUM YEARS 742, 880 DATE OF(NUM, NUM, NUM) 743, 881 DAY OF WEEK(DATE) 744, 882 DAY OF(DATE) 744, 882 DAYS FROM(DATE, DATE) 745, 883 Enter a Date 733, 871 Enter a Number 733, 871 Enter some Text 734, 872 **ERRORTEXT 746, 884** EXPONENTIAL(NUM) 747, 885 FIRST(TEXT, NUM) 748, 886 FLOOR(NUM) 749, 887

FORMAT(VALUE, 749, 887 full list of 1091 FUTURE VALUE (RATE, TERM, PAYMENT, PRESENT VALUE, TYPE) 750, 888 INTEGER(TEXT) 751, 889 LAST(TEXT, NUM) 752, 890 LENGTH(TEXT) 753, 891 LOGARITHM(NUM) 754, 892 MAX(NUM, NUM) 754, 892 MID(TEXT, NUM, NUM) 755, 893 MIN(NUM, NUM) 756, 894 MONTH OF(DATE) 756, 894 MONTHS FROM(DATE, DATE) 757, 895 MULT_CHOICE != TEXT 757, 895 MULT_CHOICE = TEXT 757, 895 NATURAL LOGARITHM(NUM) 758, 896 NOT TRUE_FALSE 759, 897 OTHER(MULT_CHOICE_VAR) 759, 897 PAYMENT (RATE, TERM, PRESENT VALUE, FUTURE VALUE, TYPE) 760, 898 POSITION(TEXT, TEXT, TRUE/FALSE) 761, 899 POWER(NUM, NUM) 762, 900 PRESENT VALUE (RATE, TERM, PAYMENT, FUTURE VALUE, TYPE) 763, 901 RATE (TERM, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE) 764, 902 REMAINDER(NUM, NUM) 766, 904 REPLACE(TEXT, TEXT, TEXT, NUM) 766, 904 RESULT 768, 906 ROUND(NUM, NUM) 768, 906 SELECTION(MULT CHOICE VAR, NUM) 769, 907 SPACE(TEXT, TEXT) 770, 908 SQUARE ROOT(NUM) 771, 909 STRIP(TEXT, TEXT, TRUE FALSE, TRUE FALSE) 771, 909 SUM(COMPUTATION VAR) 773, 911 SUM(NUM VAR) 773, 911 TERM (RATE, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE) 774, 912 TEXT CONTAINS TEXT 775, 913 TEXT ENDS WITH TEXT 776, 914 TEXT STARTS WITH TEXT 777, 915 TODAY 777, 915 TRIM(TEXT) 777, 916 TRUNCATE(NUM, NUM) 778, 916 UNANSWERED 779, 917

UNION(MULT_CHOICE_VAR, MULT_CHOICE_VAR) 779, 918 VALUE(VAR, EXPRESSION) 780, 919 YEAR OF(DATE) 781, 919 YEARS FROM(DATE, DATE) 782, 920 ZERO(NUM_VAR) 783, 921 Expressions, creating conditional 334 Extended inserts, using 380 External applications Web browsers 193 Extracting records from database using custom command 945

F

Favorites folder, create in your template library 82 Feedback, providing for HotDocs documentation 1207 Field borders changing properties of 153 changing properties of, in Automator 431 detecting to create or resize fields 150, 428 detecting to create or resize fields, in Automator 428 Field display order, specifying 946 Field labels, using 303 Field numbering codes, using in addendum 475 **Field Properties** assigning wizards at 441 Bar code 167 Bar code, in Automator 460 check box 149 check box, in Automator 424 Control 455 field types 159 field types, in Automator 451 Image 165 Image, in Automator 459 non-printing 163 non-printing, in Automator 455 previewing 450 Resource 456 Strike-through 164 Strike-through, in Automator 457 Field Properties button 408 Field type, changing 159 Field type, changing in Automator 451

Fields (database records) assigning as keys 941 displaying in assembly window 940 linking to variables 940 Fields (forms) aligning two or more 152 aligning two or more, in Automator 430 assigning font properties to 160 assigning font properties to, in Automator 452 attaching answer wizards to 441 attaching ASK instructions to 425 attaching INSERT instructions to 427 attaching Multiple Choice variables to check boxes 438 attaching variables to 425 borders 153 borders, in Automator 431 changing colors of 236 changing types 159 changing types, in Automator 451 checking for overflow 171 circling 165 circling, in Automator 458 copying 154 copying, in Automator 432 creating check box 149 creating check box, in Automator 424 detecting borders to create or resize 150 detecting borders to create or resize, in Automator 428 globally changing properties for 419 handling overflow of 245 making conditional 426 margins 153 margins, in Automator 431 moving 151 moving, in Automator 428 previewing formatting properties for 450 resizing 155 resizing, in Automator 433 rotating answers 162 rotating answers, in Automator 454 selecting 148 selecting, in Automator 423 showing background color for 450 specifying tab order for 156 specifying tab order for, in Automator 434 ungrouping 158

ungrouping, in Automator 439 Fields (text template) applying colors to 301 applying formats to specific 511 finding syntax errors in 976 going to specific 306 inserting variable 319 navigating through 302 specifying width of 517 File name extensions changing all in a folder 78 list of HotDocs-specific 49 modifying library items 76 File uploading, about 1025, 1031 Files adding to published template set 1009 changing location of HotDocs program 279 changing location of word processor 276 converting formats of 298 copying when exporting libraries 82 listed in Template Manager 669 modifying file paths for library items 76 moving 87 organizing in libraries 81 reducing size of template and document 659, 662, 819 storing answer files 186 Fill characters, using in answer fields 517 Fill Fields button 450 Fill order, specifying in forms 156 Fill order, specifying in forms in Automator 434 Fillable PDFs assembling 139 attaching variables to fields in 482 checking fields in 480 creating 410 overview 476 tips for working with 484 Filling answers in automatically 783, 921 FILTER COMPUTATION_VAR 711, 849 Filtering allow for database users 943 database records 948 database records during assembly 202 lists of components in Template Manager 669 lists of repeated answers 350 lists of repeated answers after sorting 350

lists of repeated answers using a computation 607 tips for repeated database records 951 Find and Replace in Components button 646 Find in Components button 646 Finding a person's age 735, 873 clauses in a clause library 398 dates based on separate date values 743, 881 exact position of a character in an answer 761, 899 missing IF instructions 327 missing REPEAT instructions 327 specific templates in libraries 97 Finding and replacing text strings in component files 646 variable text throughout a template 322 Finish button, defining what happens when used 217 Finish Interview Action command-line option 1137 FIRST(TEXT, NUM) 748, 886 **Fixed values** in SET instructions 784, 922 setting variables to 783, 921 FLOOR(NUM) 749, 887 Focus button 671 Folders adding all included items to a library 71 adding to clause library 394 changing all items in 78 deleting from libraries 74 for word processor files 276 in HotDocs libraries 65 locked 74 using in Answer File Manager 192 Fonts assigning to form fields 160 assigning to form fields, in Automator 452 assigning to text template fields 517 changing properties of in dialog pane 215 changing properties of in plain text scripts, prompts, dialog text 962 specifying default for edit fields (forms) 241 Footers defining in inserted templates 376 using variables in 530, 663, 821 Forcing

databases to be asked 955 dialogs to be asked 564 results of a script 768, 906 users to enter answers 723, 861 Foreign language templates creating 311 instructions to use 717, 855 Form documents assembling 97 editing 146 including resources for 456 printing 143 saving 129 saving as PDF files 130 spanning answer across multiple pages 169 spanning answer across multiple pages, in Automator 465 specify default printing options for 145 viewing 139 Form templates adding pages to 415 changing properties for multiple 419 closing 420 creating 410 creating from fillable PDFs 477 deleting pages from 417 editing 414 saving 420 updating the content of 417 viewing in multiple windows 418 FORMAT 712, 850 Format examples assigning non-breaking spaces and hyphens to 511 date 544 editing 642, 813 number 526 understanding how they're used 540 using 511 Format of uploaded answers 1037 FORMAT(VALUE, 749, 887 Formatting lists of answers 349 Multiple Choice answers 501 plain text in scripts, prompts, dialog text 962 previewing for form fields 450 variables 511 Four-digit years 212

Fractions in Number variables 526 French templates, creating 311 Frequently used templates, grouping 82 Full list of expression models 727, 865 FUTURE VALUE (RATE, TERM, PAYMENT, PRESENT VALUE, TYPE) 750, 888

G

Gathering variables and clauses into one dialog 403 Generating a list of database records in an assembled document 951 default dialog titles 635, 805 interview outline for form documents 248 outline of scripting in the template 309 German templates, creating 311 GIFs inserting in Word templates 379 Globally changing form field properties 419 Go To Answer in Document button 106 Go to Answer in Interview button 121 Go To Field button 290 Go to Template command 990 Going to a specific field in a template 306 to errors from error message 976 Graphic icons in component lists 263 Graphics inserting in dialogs 580 inserting in form fields 165 inserting in form fields, in Automator 459 inserting in Word templates 379 Greater than, less than operators 687, 825, 1095 Greatest value of two numbers 754, 892 Group name property, assigning for form fields 169 Group name property, assigning for form fields in Automator 465 Grouping ASK instructions in a computation 592 child dialogs in parent dialog 590 clauses in a dialog 403 fields as check boxes 438 form fields 157 form fields, in Automator 435 True/False variables 564

undoing a form field 158 undoing a form field, in Automator 439 variables in a dialog 561

Н

HAL file name extension 49 Handling field overflow 245 Hard and soft returns in computation scripts, using 691, 828 Hard returns in answers, using 489 HCL file name extension 49 HCL files 392 HDA file name extension 49 HDA files, creating 996 HDI file name extension 49 HDI files creating 999 installing 196 using a shortcut for 1007 using with templates on CDs 1007 HDL file name extension 49 HDP file name extension 49 HDP files 1009 Headers defining in inserted templates 376 modifying text of form addendum 475 using variables in text template 530, 663, 821 Height or width of form field, adjusting 155 Height or width of form field, adjusting in Automator 433 Help (Resources) assigning for an entire form template 456 assigning to variables and dialogs 520, 567 displaying buttons for viewing 210 viewing during assembly 109 Helper applications 193 HFD file name extension 49 HFT file name extension 49 Hidden data removing from assembled Word documents 659, 819 removing from word templates 662 Hide Library command-line option 1138 HIDE VAR, HIDE ALL 713, 851 Hiding End of Interview dialog 117 End of Interview dialog from users 655, 817

field labels and comments 303 form field backgrounds 450 interview outline in assembly window 635, 805 unanswered variable warnings 515 variable names in form fields 450 variables in an answer source (pick list) 708, 845 Highlight Fields button 121, 125 Highlighting form fields 148 form fields, in Automator 423 text when tabbing between answer fields 209 Histories for answer files 190 Honoring IF instructions in a question summary 254 Horizontally aligning form fields 152 Horizontally aligning form fields, in Automator 430 HotDocs what is it? 38 HotDocs Auto-Assemble File command-line option 1138 HotDocs Auto-Install File command-line option 1138 HotDocs Automator about 405 creating form templates for 410 launching to edit form templates 414 toolbars 408 HotDocs button 290 HotDocs Core Services publishing templates for 1028 uploading templates for 1028 HotDocs files, uploading 1025, 1031 HotDocs Filler 146 HotDocs Help button 290 HotDocs Model command-line option 1139 HotDocs Models allowing non-standard names in 233 converting templates to 1087 converting to HotDocs templates 1088 creating 1041 naming standards to use in 1089 HotDocs Options button 205 HotDocs Outliner button 290 HotDocs Outliner, using 309 HotDocs Player, registering templates for use with 1005

HotDocs Server about 1013 creating templates for 1019 enabling all new templates for use with 266 feature support 1015, 1016 publishing templates for 1022 testing a template for 1020 HPD file name extension 49 HPL file name extension 49 HPT file name extension 49 HTML converting summaries to 253 creating documents for uploading 1033 HTTP POST, using for uploaded answers 1037 HVC files, publishing 1022, 1028, 1035 **Hyperlinks** including in dialogs 578 including in prompts, dialog element text 964 including on forms 456 to published files on the Internet 193 Hyphens and spaces, using non-breaking 511

L

Icons changing appearance of component 263 in interview outline 106 using custom for auto-install files 1007 Identifier, clause 392 Identifying correct templates for assembling 96 IF EXPRESSION, ELSE IF, ELSE, END IF 714, 852 IF Field button 290 IF instructions about 328 attaching to form fields 426 creating more complex 334 creating using True/False expressions 334 creating using True/False variables 333 editing 336 matching with END IF instruction 327 nesting 337 Image files inserting in dialogs 580 inserting in form fields 165 inserting in form fields, in Automator 459 Images inserting in Word templates 379 Import Answer File button 189
Importing answers from another answer file 101 library files from other libraries 84 templates or clause libraries from other libraries 71 Inactive form fields, displaying a warning when typing in 237 Inches as a unit of measure in forms 235 Including resources for variables 520, 567 Increasing values of temporary counters 716, 854 INCREMENT NUM_VAR 716, 854 Indenting answers in a form field 453 answers in form addendum 246 Infinite loops, preventing 627, 635, 799, 805 Information about variables, including 513 Information-gathering dialogs, using 89 INSERT 716, 854 **INSERT Field button 290 INSERT** instructions about 360 attaching to form fields 427 editing 366 for clauses 401 for graphics 379 inserting an ASSEMBLE instruction 381 tips on shared component files 365 Insert Variable button 618, 790 Inserted dialogs answering during assembly 114 removing icons for 564 using during development 589 Inserting answer files in e-mail messages 185 ASK instructions 592 ASSEMBLE instructions in templates 381 clause libraries in templates 402 clauses in assembled text documents 105, 115 current date 509 dialogs for nested repeats 610 document files in e-mail messages 177 files using variable INSERTs 380 form templates 427 hyperlinks in prompts, dialog element text 964 images in Word templates 379 individual clauses in templates 401 lists within lists 610

multiple templates in main template 381 new pages in a form template 415 repeating dialogs 346 special characters in plain text scripts, prompts, additional text 966 templates into templates 716, 854 templates that have different headers/footers 376 variables directly into templates 319 variables using Component Manager 322 Insertingdialogs into dialogs 589 Installing new word processors after HotDocs is installed 274 template sets (HDIs) 196 Instant Update button 96 Instant Update, using 96 Instruction and Expression models, using at Computation variable 506 Instruction models ADD TEXT TO MULT_CHOICE 703, 840 ASCEND VAR 704, 842 ASK DATABASE 705, 842 ASK DIALOG 705, 843 ASK VAR 706, 843 ASSEMBLE 706, 844 CLEAR MULT_CHOICE 703, 840 CONCEAL VAR 708, 845 DEBUG 708, 846 DECREMENT NUM VAR 716, 854 DEFAULT VAR TO VALUE 709, 847 **DESCEND VAR 704, 842** END REPEAT 722, 860 ERASE DIALOG 710, 848 ERASE VAR 710, 848 FILTER COMPUTATION_VAR 711, 849 FORMAT 712, 850 full list of 700, 838 HIDE ALL 713, 851 HIDE VAR 713, 851 IF EXPRESSION, ELSE IF, ELSE, END IF 714, 852 **INCREMENT NUM VAR 716, 854** INSERT 716, 854 LANGUAGE 717, 855 LIMIT NUM 718, 856 OMIT VAR 719, 857 PLAY 720, 858 QUIT 721, 859

REPEAT DATABASE 722, 860 REPEAT DIALOG 722, 860 **REQUIRE ALL 723, 861 REQUIRE VAR 723, 861** SET VAR TO VALUE 724, 862 SHOW ALL 713, 851 SHOW VAR 713, 851 WHILE EXPRESSION 725, 863 INTEGER(TEXT) 751, 889 Internet adding auto-assemble files to HotDocs libraries 195 adding Web pages to HotDocs libraries 199 launching browsers from HotDocs 199 publishing HotDocs files on 193 registering browser helper applications 193 starting HotDocs assemblies from browsers 193 uploading answer files to Web servers 199 using HotDocs to assemble documents on 1013 using to distribute templates 991 Interview outlines icons in 106 updating during assembly 96 Interview Scope command-line option 1139 Interview templates assembling 100 creating 603 Interviews about 595 defining for form templates 421 defining for templates 600 displaying repeated dialog icon in 207 End of Interview dialog 106 hiding outlines in assembly window 635, 805 speeding up processing of 96 verifying information before ending 120 Invalid computations 721, 859 Invisible variables in a dialog 713, 851 Italian templates, creating 311 Item Properties button 76

J

Joining database tables 953 JPGs adding to dialogs 580 inserting in form fields 165 inserting in form fields, in Automator 459 inserting in Word templates 379 JS files, publishing 1022, 1028, 1035

Κ

Keep Headers/Keep Footers property 376 Keep Interview Group command-line option 1139 Keyboard shortcuts, list of 53 Keys, database 941

L

Label Fields button 290 Labels assigning to addendum items 246 assigning to REPEAT and IF fields 303 assigning to repeat icons 564 customizing for repeated lists 348 LANGUAGE 717, 855 LANGUAGE instructions 311 LAST(TEXT, NUM) 752, 890 Launching HotDocs Filler 135 HotDocs Outliner 309 **Template Manager 668** Layout changes for variables in dialogs 568 customizing windows and panes 95 library windows 65 Left-aligning dialog prompts 564 LENGTH(TEXT) 753, 891 Lengthening answers in the document 517 Lesser of two numbers, finding 756, 894 Letting users specify Multiple Choice options 703, 840 Libraries about 65 adding auto-assemble files to 195 adding items to 71 adding web pages to 199 changing items in 76, 78 commands available in 72 copying items in 86 creating 69 creating by exporting existing 82

deleting items in 74 dividing into smaller libraries 82 exporting 82 importing other libraries into 84 layout of 65 limiting access to 84 opening 69 organizing items in 81, 86 printing contents of 85 references to actual files 65, 87 saving 85 saving contents as text file 86 searching for specific templates 97 sorting items in 80 switching to a different library 69 Library File command-line option 1140 LIMIT NUM 718, 856 Limiting database records that appear for selection 948 dialog repetitions 352 lists of components in Template Manager 669 Line breaks forcing in Text variables 489 in computation scripts, using 691, 828 Line format properties in form fields 453 Line spacing, determining for form fields 453 Lines per inch option for fields 241 Lines, adding to dialogs 579 Linked answers, refreshing between assemblies 943 Linking answer files to templates 182 database tables together 953 variables to database fields 940 variables to dialogs 564 warning about breaking 267 List formats 540 Lists alphabetizing answers in a REPEAT field 350 choosing specific answers from 359 counting answers in 358 creating 346 creating using records in database 951 displayed for completed assemblies 225 displaying in forms 462, 463 displaying in word processor tables 355 filtering answers from 350 filtering answers from using computation 607

formatting to appear as A, B, and C 349 limiting answers in 352 of answers in a document 179 of components, generating 646 of questions in a template 178 printing of library contents 85 providing answers to create 112 punctuating non-repeated 968 totalling numbers in 352 using columns to display 357 using particular answer from 359 Lists within lists about 609 creating 610 using a computation to create 612 using a particular answer from 613 using one dialog to display 611 Loading template sets 196 Local disks, saving libraries to different locations on 69 Local Variable 694, 832 Locating correct templates 96 items in a library 80 Locations, template (default) 276 Lock Answer File command-line option 1140 Lock Library command-line option 1141 Locking component files 1004 LOGARITHM(NUM) 754, 892 Looping through an answer 725, 863

М

Macros, defining post-assembly 653, 815 Maintaining templates on a network 84 Making changes to templates 300 Managing answer files 282 answer files through library commands 190 assembled documents 282 components from the template library 618, 790 components in a component file 618, 790 components in multiple component files 664 Mandatory answers, requiring 723, 861 Manual indexing items in a list 359 Mapping variables to fields in a database table 940

variables to fields in a Time Matters record 588 variables to fields in an Outlook Contacts list 587 variables to other variables in the template 586 Margins changing for form fields 153 changing for form fields, in Automator 431 defining in inserted templates 376 specifying for form edit fields 241 Marking merge fields for assembled documents 635, 805 Markup Syntax about 1041 options for working with 233 requirements for using 1089 Markup View looking at an assembled document in 126 setting options for 227 Markup View button 290 Match Fields button 290 Matching fillable fields with HotDocs fields 480 IF or REPEAT instructions with END instructions 327 Mathematical operators, using 687, 825, 1095 MAX(NUM, NUM) 754, 892 Maximum number of lines for a text field 242 Measurement, defining units of 235 Merge fields creating in text templates 319 marking in assembled text documents 635, 805 Merge Text adding more information to 527 assigning names to 530 default vs. field-specific 528 editing groups 642, 813 Merging answer file information into template 509 answers into other answer files 101 filtered database records into document automatically 952 lists of answers 346 lists of answers in a form table 462, 463 Microsoft Data Link files, creating 937 MID(TEXT, NUM, NUM) 755, 893

MIME types, registering HotDocs as a helper application 193 MIN(NUM, NUM) 756, 894 Minimize All button 620, 792 Minimum shrink-to-size option for form fields 241 values for numbers 493 Modifiable text, creating sections of in a text template 386 Modifying component file properties 635, 805 component file properties, in Template Manager 683 custom dialogs 564 IF instructions 336 **INSERT** instructions 366 Multiple Choice options 703, 840 **REPEAT** instructions 347 templates 300 the properties for a single library item 76 the properties for multiple library items 78 variables 510 MONTH OF(DATE) 756, 894 Month portion of a given date, returning 756, 894 Month-Day-Year date order 211 MONTHS FROM(DATE, DATE) 757, 895 Mouse-over text for resource buttons 635, 805 Move to TypeHere bookmark 635, 805 Moving around in a form 413 between fields in a text template 302 cursor to location in assembled document 656, 818 files on local disk or network 87 form fields 151 form fields, in Automator 428 from test assembled document to template 990 library items 86 templates or clause libraries from other libraries 71 through answer-gathering process 106 windows and panes 95 MULT_CHOICE != TEXT 757, 895 MULT_CHOICE = TEXT 757, 895 Multi-line form fields formatting 453

splitting between form and addendum 176 splitting between form and addendum, in Automator 473 Multiple lines in fields, typing in 209 versions of paragraphs, inserting 333 word processors, using 274 Multiple Choice variables adding more text to prompts 527 attaching to check-box fields 438 combining answers from 779, 918 copying and pasting options between 529 creating 501 dynamically adding options during assembly 703, 840 extracting data from 769, 907 modifying options at runtime 703, 840 naming groups of merge text in 530 using in conditional expressions 334 Multiple component files copying components between 681 deleting components from 683 editing 620, 792 modifying preferences in 683 renaming components in 682 viewing the contents of 669 Multiple form fields, splitting answers across 157, 435, 437

Ν

Names assigning to merge text groups 530 assigning to variables 538 searching for specific templates 97 tips for assigning to variables 584 NATURAL LOGARITHM(NUM) 758, 896 Navigating through fields in a text template 302 through interview 106 Navigation bar controlling tab order in 208 using to get through interview 106 Nested repeats, about 609 Nestina IF instructions 337 repeated dialogs 610, 612 Networks

access to answer files 190 copying libraries from 84 maintaining templates on 84 New answer files, creating 181 dialogs, creating 561 form fields, creating 147 form fields, creating in Automator 422 form templates, creating 410 libraries, creating 69 text templates, creating 295 variables, creating 319 New Answer File button 181 New Answer File command-line option 1141 New features of HotDocs 10 11 117 2005 22 2005 SP1 22 2005 SP2 22 2005 SP3 22 2006 18 2006 SP1 18 2007 17 2007 SP1 17 2008 15 2008 SR1 15 2008 SR2 15 2009 13 6.0 33 6.1 30 6.1 SP1 30 6.2 26 6.2 SP1 26 New features of HotDocs 10 11 New features of HotDocs 10 HF1 11 New features of HotDocs 10 HF2 11 New features of HotDocs 10 HF3 11 New features of HotDocs 10.1 11 New features of HotDocs 10.2 11 New features of HotDocs 11 7 New features of HotDocs 2005 22 New features of HotDocs 2006 18 New features of HotDocs 2007 17 New features of HotDocs 2008 15 New features of HotDocs 2009 13 New features of HotDocs 2009 HF1 13 New features of HotDocs 2009 HF2 13

New features of HotDocs 6 33 New features of HotDocs 6.1 30 New features of HotDocs 6.2 26 New Form Template button 410 New Library button 69 New Template button 295 Next button first in Navigation Bar tab order 208 Next Field button 290 Next Page command, using 413 No Assembly Window command-line option 1141 No Interview command-line option 1142 Non-breaking spaces and hyphens, assigning to formats 511 None of the Above option testing whether selected 759, 897 using in Multiple Choice variables 501 Non-English templates, using 311 Non-printing fields, assigning 163 Non-printing fields, assigning in Automator 455 Nonrelational databases, connecting to 945 Non-repeated lists, creating 968 Not asking dialogs automatically 571 NOT TRUE_FALSE 759, 897 Notes, adding to components 524, 569 Notifying users of published template updates 1010 Number formats 540 Number of days between two dates 745, 883 Number of lines allowed in a single-line field 242 Number of months between two dates 757, 895 Number variables creating 493 formatting in foreign languages 311 using HotDocs paragraph numbering 509 Numbering formatting correctly when scripting 733, 871 lists of answers 358 paragraphs in text templates 509

0

ODBC-based connections using 935 ODMA DLL file locations 282 Offline, using downloaded auto-assemble files 195 OLE DB providers, list of supported 927 OMIT VAR 719, 857 Omitting buttons for child dialogs 564 On the fly dialog scripting 582 inserting 380 One component file for multiple templates 651, 813 Online compatibility, checking for 1019 templates, creating 1019 testing of templates 1020 Open Library button 69 Opening Answer File Manager 188 assembled documents in other applications 127 clause library 393 Component Manager 617, 789 HotDocs libraries 69 **Template Manager 668** Web pages from HotDocs 199 Operators, using 687, 825, 1095 Options tab at a dialog 564 at Multiple Choice variable 527 Options, specifying for Multiple Choice variables 501 Ordering answers alphanumerically 704, 842 fields in a form 156 fields in a form, in Automator 434 Organizing answer files 192 HotDocs files 81 Orientation, changing for form fields 162 Orientation, changing for form fields in Automator 454 Other option chosen in Multiple Choice, testing whether 759, 897 OTHER(MULT_CHOICE_VAR) 759, 897 Outline of dialogs, updating 96 Outliner, HotDocs 309 Outlook, retrieving data from 587 Output File command-line option 1142 Output format of assembled document 274 Overflow checking form fields for 171

defining properties for form tables 468 defining properties for regular form fields 174 defining properties for regular form fields, in Automator 466 defining properties for run-on form fields 467 warning about answer 237 Overlay Answer File command-line option 1143 Overlay answer files 101 Override Field command, using 141 Overriding answers or formats in form documents 141 form answers, displaying a warning when 237 Overwriting answers with other answer files 101 existing components when copying 636, 807

Ρ

Packaging templates or clause libraries to send to other users 82 Page numbering codes, using in addendum 475 Pairing IFs and END IFs 327 Panes accessing resources 109 customizing layout 95 Paper size and source for printing, selecting 248 Paper Size command-line option 1144 Paper size, choosing for forms 145 Paper Tray command-line option 1145 Paper tray, choosing for forms 145 Paragraphs formatting in multi-line form fields 453 numbering 509 using breaks in computation scripts 691, 828 using breaks in Text variables 489 Parameter 697, 835 Passwords assigning to template sets 1004 specifying for database connection 949 Paste Components button 681 Pasting components across multiple component files 681 form fields 154 form fields, in Automator 432 Patterns (text) assigning 489 creating 531

editing 642, 813 PAYMENT (RATE, TERM, PRESENT VALUE, FUTURE VALUE, TYPE) 760, 898 PDF files creating from assembled documents 130 printing 238 sending assembled documents automatically to 236 using fillable 477 **PDF** templates converting existing HFTs to 411 creating 410 PDF417 bar codes, creating 167 PDF417 bar codes, creating in Automator 460 Percentages 526 Personal Information variables creating 256 editing 257 using 508 Pick lists (answer sources) creating 583 gathering information for 100 linking to an interview template 603 linking to dialog in another template 586 Pictures inserting in dialogs 580 inserting in form fields 165 inserting in form fields, in Automator 459 inserting in Word templates 379 Placeholders default text for all unanswered variables 223 text for specific unanswered variable 517 Plain text help for variables 520 plain text template 313 PLAY 653, 720, 815, 858 PN# code 509 PNGs adding to dialogs 580 inserting in form fields 165 inserting in form fields, in Automator 459 inserting in Word templates 379 Pointed component files defining which component file provides properties 654, 816 unpointing 655, 817 using 651, 813 using with inserted templates 365 warn when editing 267

Points as a unit of measure in forms, using 235 Position of errors in a template or script 976 POSITION(TEXT, TEXT, TRUE_FALSE) 761, 899 Positionina clause library with other open windows 398 Component Manager with template window 618, 790 test assembly window with template window 975 Positive expressions of negative numbers, finding 734, 872 Post-assembly macros, storing in special templates 653, 656, 815, 819 POWER(NUM, NUM) 762, 900 Pre-answering variables 709, 847 Pre-determined lists of answers, using 583 Preferences, component file 635, 805 PRESENT VALUE (RATE, TERM, PAYMENT, FUTURE VALUE, TYPE) 763, 901 Pressing Enter to insert lines 209 Preventing infinite processing of instructions in scripts 635,805 unanswered variable warnings 515 use of answer files 515, 635, 805 variables from appearing during assembly 515 variables from appearing in pick lists 708, 845 Previewing assembled text document before sending to word processor 120 form documents 139 form template field properties 450 templates before assembly 96 Previous and Next buttons in a text template 302 in the dialog pane 208 Previous Field button 290 Previous Page command, using 413 Print Answers Only command-line option 1145 Print Both command-line option 1146 Print command-line option 1145 Print Components button 651 Print Copies command-line option 1146 Print Document button 129, 143 Print Duplex command-line option 1146 Print Form Only command-line option 1147 Print Without Dialogs command-line option 1147

Printing assembled form documents 143 assembled text documents 129 blank copies of templates 72 component lists 651 fields, keep from 163 fields, keep from, in Automator 455 form templates 410 list of clauses 398 lists of files in a library 85 specifying default for forms 238 specifying default options for forms 145 to create PDF files 130 using Adobe or Filler functionality 238 warn when using default options 221 Processing stack controlling instructions that get added to 635, 805 debugging items on 981 Product titles, creating 635, 805 Prompt before printing assembled document via word processor option 221 Prompts assigning for Multiple Choice options 501 creating 513 placing variables in 532 positioning in dialogs 564 suppressing 513 Properties assigning font to form template fields 160 assigning font to form template fields, in Automator 452 changing clause 398 changing component file 635, 805 modifying library item 76 specifying for form edit fields 241 Property Replicator tool, using 419 Protecting published files 1004 Providers.ini file, using 927 Provisional sections, creating in template 333 PUB reference path keywords 999 Published settings files 1009 Publishing about 991 adding files while 1009 auto-assemble files 996 auto-install files 999 HotDocs Core Services template files 1028

HotDocs Server template files 1022 HTML Destination Page 1033 HTML Form Page 1033 protection options 1004 registered templates 1005 saving settings for 1009 standard template files 994 Publishing Wizard button 991 Punctuating answers in plain text scripts, prompts, additional text 965 lists of answers 349 lists of answers using a computation 606

Q

Question summaries changing appearance of 253 controlling asked variables in 254 creating 178 Question Summary command-line option 1148 Questions, not asking during assembly 100 QUIT 721, 859

R

RATE (TERM, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE) 764, 902 Records customizing list of returned 948 sorting lists of returned 950 Recovering form template files 239 Red markers in file and component lists in Template Manager 669 Re-displaying suppressed warning messages 267 References to answer overflow in addendum 246 Refining lists of answers in a repeat 350, 711, 849 Reformatting in answer fields 211 Refresh Database button 671 Refresh linked answers every assembly option 947 Refresh Outline button 309 Refreshing cache for auto-assemble files 195 component lists in Template Manager 669 data from database between assemblies 947 Registering

HotDocs to work with Web browsers 193 templates for use with HotDocs Player 1005 Registry creating Personal information variables whose values are saved in 508 Relationships between components, viewing 522, 569 REMAINDER(NUM, NUM) 766, 904 Remembering answer file used last 99 records selected from database 941 Remove Item button 74 Removing an answer wizard from a field 441 answers from a repeated dialog 112 clauses from a clause library 398 components from current file 636, 806 entries from a list using a filter 711, 849 form template pages 417 hidden data from assembled documents 659, 819 hidden data from word templates 662 options from a Multiple Choice variable 703, 840 Personal Information variables 256 variables from an answer source (pick list) 719, 857 Rename button 678, 682 Rename Components button 638, 808 Renaming components across multiple component files 682 components in one component file 638, 808 components when copying 636, 807 supplemental components 638, 808 templates 678 variables 638, 808 REPEAT DATABASE, END REPEAT 722, 860 REPEAT DIALOG, END REPEAT 722, 860 **REPEAT Field button 290 REPEAT** instructions attaching to tables in form templates 462, 463 filtering 350 filtering using computation 607 inserting in template 346 limiting repetitions 352 matching with END REPEAT instructions 327 punctuating 349

punctuating using a computation 606 using for database components 951 using in computations 605 Repeated dialogs (answering) removing answers from 112 using special icons to denote 207 Repeated dialogs (editing) assigning custom titles to 348 creating 346 creating using a computation 605 displaying 347 editing 347 filtering 350 filtering using computation 607 limiting answers in 352 punctuating 349 punctuating using a computation 606 sorting 350 sorting using a computation 606 styles for 347 tips on using in a computation 608 Repeating an answer until a condition is met 725, 863 database records 951 dialogs 346 tips for filtering and sorting database records 951 Replace Static Text command, using 417 REPLACE(TEXT, TEXT, TEXT, NUM) 766, 904 Replacing existing libraries 69 multiple instances of text with a variable 322 template text with variables 319 Reporting errors in templates and scripts 976 REQUIRE ALL, REQUIRE VAR 723, 861 Requiring four-digit years in answers 212 variables to be answered 582 Resizing Component Manager window 617, 789 fields 155 fields using Detect command 150 fields using Detect command, in Automator 428 fields, in Automator 433 windows and panes 95 Resolving answer overflow 171 Resource buttons, using 210

Resource fields, changing field type in Automator 451 Resource fields, changing field type to 159 Resource fields, creating 456 Resource Pane button 109 Resources assigning to a component 520, 567 assigning to an entire form template 456 customizing button label for 635, 805 viewing for dialogs and variables during assembly 109 Restore All button 620, 792 Restoring default colors used in dialogs 215 Restricting number of answers in lists 352 RESULT 768, 906 Retrieving data from a REPEAT instruction 359 data from a Time Matters record 588 data from an Outlook Contacts list 587 filtered records from database automatically 952 Returning characters first in a text string 748, 886 last in a text string 752, 890 middle in a text string 755, 893 Returning value of unanswered Number variable 783, 921 Returns after instructions, understanding 263, 323 Reviewer, displaying templates for a 227 Reviewing answers used in an interview 179 Rotating answers in a form field 454 ROUND(NUM, NUM) 768, 906 Rounding numbers up or down 768, 906 Rows to display setting, using 351 Rows, controlling how many are viewable in dialog 351 Rows, viewing individually during interview 115 RSS feeds, using to update published template sets 1010 RTF file name extension 49 Run-on fields creating 157 creating, in Automator 435 specifying overflow properties for 467

S

Sample code Publishing Destination Page 1024 Publishing Form Page 1023 Save All button 620, 792 Save and Close All button 620, 792 Save and Close Template button 290 Save Answer File button 111 Save Answers command-line option 1148 Save Answers Prompt command-line option 1148 Save Component File button 618, 790 Save Document button 129 Save Template button 290 Saving ADO connection string in separate file 937 answers 111, 182, 183 assembled documents always 221 assembled documents as PDF files 130 assembled text or form documents 129 changed answers back to the database 942 components using Component Manager 618, 790 database records in answer file 203 fillable PDF documents 139 form templates 420 from End of Interview dialog 117 information about used printers 238 libraries 85 library contents as text file 86 options selected for publishing 1009 PDF documents 130 Script editor customizing appearance of 269 using 534, 692, 829 Script link, adding to dialogs 577 Scripts adding to dialogs 581 conditional 333, 334 creating computation 506 customizing appearance of editor 269 debugging 981 improper use of 989 local variables 694, 832 parameters 697, 835 testing functionality of 972 using an editor to write 534, 692, 829

using the correct syntax 546, 685, 823 viewing an outline of 309 writing 534, 692, 829 Scrolling around in a form 413 Searching and replacing text strings in component files 646 clause library for clauses 398 for specific answer files 190 for specific templates in libraries 97 Secondary dialogs, using 589 Security for database connections 949 for PDF documents 133 Select Tool button 148 Select Tool button, in Automator 423 Selecting all filtered records in database automatically 952 All That Apply options 501 answer files 99 answer files before assembly option 222 clauses from clause libraries 115 existing answers when tabbing between fields 209 form fields 148 form fields, in Automator 423 One Only option 501 pre-specified answers 583 records in a database during assembly 200 SELECTION(MULT CHOICE VAR NUM) 769, 907 Send Document button 127 Send to Word Processor command-line option 1149 Sending answer files in e-mail messages 185 answers to an addendum 245 assembled documents to default word processor 274 clauses to an open word processor document 105 document files in e-mail messages 177 document to word processor without displaying End of Interview 655, 817 text documents to word processor 127 Sending answers to addendum to resolve form overflow 174

Sending answers to addendum to resolve form overflow, in Automator 466, 467, 468 Sentence-style lists creating 349 creating using a computation 606 Separate addendums, creating 474 Separating specific answers from a list 350 Separators, using in non-English dates and numbers 311 Series of answers, presenting 349 Server cursors, using to connect to database 955 Servers uploading answer files to 199 uploading templates to 1035 SET instructions creating 783, 921 using when writing computations 785, 923 values in 784, 922 SET VAR TO VALUE 724, 862 Setting styles for repeated dialogs 347 Shared component files defining which component file provides properties for template 654, 816 unpointing 655, 817 using 651, 813 using with inserted templates 365 warn when editing 267 Sharing answers between templates 538 components between component files 651, 813 Shortcut keys for the keyboard 53 Shortcuts, creating for template sets 1007 Shortening a number after a decimal 778, 916 Show Fields button 450 Show Fillable Fields command 480 Show Library command-line option 1149 Show Variables button 309 Show Variables button, in a form 450 Showing already linked variables when mapping 940 comments and field labels 303 form field backgrounds 450 levels of nesting in REPEATs and IFs 303 variable names in form fields 450 Shrinking answers to fit in fields 245 answers to resolve regular field overflow 174

answers to resolve regular field overflow, in Automator 466 Silverlight viewing interviews in a browser 1020 Simultaneous editing and assembly 975 editing of variables 620, 792 Single-character boxes, attaching variables to 437 Singled-sided forms, printing 145 Single-line fields, typing in 209 Skipping components when copying 636, 807 Slowness, fixing in interviews 96 Smart returns, inserting with instructions 263 Software e-mail applications 185 Sort & Filter tab of Database Editor 950 Sort Components button 618, 790 Sorting allow for database users 950 answers before filtering 350 answers in a spreadsheet 112 clauses in a clause library 398 components by name or by type in Component Manager 618, 790 components in Template Manager 669 database records during assembly 201 library items 80 lists of answers in a REPEAT field 350 lists of answers using a computation 606 tips for database records 951 SPACE(TEXT, TEXT) 770, 908 Spaces and hyphens, using non-breaking 511 Spacing, adding to dialogs 579 SPAN Field button 290 SPAN fields about 385 inserting for existing text 386 inserting for new text 387 Spanish templates, creating 311 Specifying user name and password for database connection 949 Speed, increasing interview processing time during assembly 96 Spell Check Components button 648 Spell checking answers 119 components 648

options for 259 Splash screens, creating custom 1010 Splitting answer across multiple form fields 157 answer across multiple form fields, in Automator 435 answer across single-character form fields 437 answer across two or more form pages 169 answer across two or more form pages, in Automator 465 multi-line answers between form and addendum 176 multi-line answers between form and addendum, in Automator 473 Spreadsheet exporting Template Manager list to 673 viewing individual rows of during the interview 115 Spreadsheets controlling number of viewable rows 351 copying and pasting information in Multiple Choice 529 defining widths of 516 lists of questions and answers in a document 180 Spreadsheet on Parent option 611 using additional text only in 575 using to gather lists 347 SQUARE ROOT(NUM) 771, 909 Standard answers, assigning to templates 182 Standard template files, publishing 994 Start Interview Group command-line option 1150 Starting assembly sessions 103 HotDocs Filler 135 Template Manager 668 **STARTUP** computations 603 Static content circling 165 circling, in Automator 458 detecting underlying to create new fields 150 detecting underlying to create new fields, in Automator 428 updating form templates with new 417 Status of templates selected for assembly 103

Status bar, information about gueued assemblies 103 Stepping through an interview or script to isolate problems 981 Stop sharing component files 655, 817 Stopping assembly sessions 103 variables from appearing during assembly 515 Storing answer files 190 answers for a single template 182 assembled documents 129 typical answers 101 Striking through static text on a form document 164 Striking through static text on a form document, in Automator 457 STRIP(TEXT, TEXT, TRUE_FALSE, TRUE_FALSE) 771, 909 Styles of repeating lists 347 Subtracting (values) number of days from Date variable 740, 878 number of months from Date variable 740, 878 number of years from Date variable 741, 879 one value from another 687, 825, 1095 Suggest Save command-line option 1150 Suggest Save New command-line option 1151 Suggesting answers for users 709, 847 frequently used answer files 183 Sum of repeated numbers 352 SUM(COMPUTATION_VAR) 773, 911 SUM(NUM_VAR) 773, 911 Summaries lists of answers 179 lists of questions 178 spreadsheet of questions and answers 180 Summing values in computations 773, 911 in Number variables 773, 911 Supplemental components renaming 638, 808 viewing in Template Manager 669 Support, getting for HotDocs 1203 Suppress Unanswered Warning command-line option 1151 Suppressed warnings, redisplaying 267

Swiss German templates, creating 311 Switches, full list of command-line 1129 Switching answer files during assemblies 112, 184 Synch Template button 309 Syntax finding errors and resolving 976 using correct in scripts 546, 685, 823 System failure, protecting form template files from 239

Т

Tab order controlling in Navigation Bar 208 specifying for direct-fill assembly 156 specifying for direct-fill assembly, in Automator 434 Table of Contents, updating in Word after assembly 120 Tables (form templates) creating nonstandard 463 creating standard 462 sending overflow answers to addendum 245 specifying overflow properties for 468 Tables (text templates) 355 Tabs (assembly and library windows) Answer Summary 179 choose assembly window placement of 106 choose library placement of 96 **Document Preview 120** Form Document 139 Interview 106 Preview 96 **Properties 76** Question Summary 178 Variable Sheet 180 Tabs (in Component Editors) Advanced 515, 516, 517 Lavout 568 Options 527, 564 Resource 520, 567 Script 581 Used In 522, 569 Technical Support in the United States 1203 outside the United States 1203 Telephone numbers, using 489

Template File command-line option 1151 Template files about 289 adding to libraries 71 assigning names to sets of 635, 805 assigning titles and descriptions 295 changing default file locations of 276 converting to new formats 298 creating based on existing template 297 creating form 410 creating interview 603 creating text 295 descriptions in libraries 65 editing 300 grouping in Favorites folder in library 82 installing sets of 196 linking to specific answer files 182 moving files on local disks or networks 87 organizing groups into separate libraries 69 previewing before an assembly 96 publishing auto-assemble files from 996 publishing auto-install files from 999 publishing HotDocs Server files from 1022 publishing standard files from 994 renaming 678 searching for specific in libraries 97 using LANGUAGE instructions in 311 using one component file for multiple 651, 813 **Template Manager** about 664 copying and pasting components using 681 customizing the view of 669 deleting components using 683 exporting list from 673 modifying properties using 683 renaming components using 682 setting display options 271 starting and exiting 668 toolbar 671 using to rename templates 678 viewing the contents of 669 Template Manager button 668 Template set updates checking for 198 setting properties for 285 Template sets, updating 1010

TERM (RATE, PAYMENT, PRESENT VALUE, FUTURE VALUE, TYPE) 774, 912 Test answer files 975 Test answers, entering for form fields 450 Test assembly 974 **Test Panel** about 986 Go to Template tab 990 Variable Usage tab 987 Warnings tab 989 Testing about 971 custom dialogs 973 document assembly 974 for improper scripting 989 online assembly 1020 scripts 972 True/False expressions 972 using answer files 975 variables 972 Text formatting correctly when scripting 734, 872 specifying labels in addendum 246 using additional in dialogs 575 Text boxes, using variables in 530, 663, 821 **TEXT CONTAINS TEXT 775, 913** Text documents printing 129 saving as PDF files 130 TEXT ENDS WITH TEXT 776, 914 Text files, saving libraries as 86 Text formats 540 text only template 313 Text patterns adding 489 creating 531 editing 642, 813 TEXT STARTS WITH TEXT 777, 915 Text strings in component files, searching for 646 Text templates, creating 295 Text variables controlling answer field height of 489 controlling number of answer characters in 489 creating 489 creating patterns for 531 Tiling form windows 418

Time answering questions about 106 converting values to numbers 751, 889 creating variables for inserting 526 Time Matters, linking variables in dialogs to fields in 588 Time of day, inserting 526 Time patterns, using 526 Titles dialog 561 modifying library items 76 template 295 variable 548 TODAY 777, 915 **TODAY** variables 509 Toolbars Component Manager 615, 786 displaying full in assembly window 220 HotDocs assembly window 90 HotDocs Automator 408 HotDocs Edit 290 HotDocs Navigation 290 **Template Manager 671** Totaling answers in a list 358 Tracking answer file histories 190 variable usage in a template 987 Trading answer files during assembly 112, 184 Transferring answer files to other users 185 assembled documents to editing application 127 document files to other users 177 library items into a new library 82 library items into existing library 84 library items to other libraries 71 Treat linked variables as asked option 953 TRIM(TEXT) 777, 916 Troubleshooting template automation 977 True/False expressions, creating 334 True/False formats 540 True/False variables creating 498 grouping in dialog 564 specifying options 498 TRUNCATE(NUM, NUM) 778, 916

Two text values containing same text string, determining 775, 913 Two-digit years 212 Two-dimensional bar codes, creating 167 Two-dimensional bar codes, creating in Automator 460 TypeHere bookmark 656, 818 Types of templates 289 Typing answers directly in a form field 136

U

UDL files, creating 937 **UNANSWERED 779, 917** Unanswered variable text specifying at Component Manager 635, 805 specifying at variable 517 Unanswered variables asking only 1139 formatting 223 reported at End of Interview dialog 117 warning about 220 Underscore as a placeholder 223 Ungrouping fields 158 Ungrouping fields, in Automator 439 UNION(MULT_CHOICE_ MULT_CHOICE) 779, 918 Universal Data Link files, creating 937 Unpointing component files 655, 817 Unsupported characters in variable names 538 features in HotDocs Server (detecting and eliminating) 1019 Unused components removing from component file 683 Updating all item properties in a library folder 78 computation results as you script 768, 906 database fields in interview 943 database records with answers entered during interview 942 dialogs shown in interview outlines 96 downloaded auto-assemble files 195 existing answer files with new answers 111 library items located on a CD 1007 static content of form templates 417 template sets automatically 1010 template sets using auto-install files 999

template sets via the Internet 198 test assemblies 975 Uploading answer files 199 answers to Web servers 199 answers, format for 1037 enabling for users 1036 HotDocs files 1025, 1031 template files to a server 1035 URLs adding to HotDocs libraries 199 correct format for adding auto-assemble files from the Internet 195 Use Defaults option 511 Used In tab 522, 569 User names, specifying for database connection 949 Using existing variables in templates 636, 807 properties stored in the shared component file 654, 816 variable names in summaries 635, 805

۷

VALUE(VAR, EXPRESSION) 780, 919 Values computed 506 in SET instructions 784, 922 setting using scripts 785, 923 Variable Field button 290 Variable INSERT instructions, using 380 Variable mapping 586 Variable name, determining for question in interview 117 Variable properties adding comments 525 adding example formats 511 adding prompts 513 assigning resource text 520, 567 assigning titles 548 seeing associated components 522, 569 specifying advanced options 515, 516, 517 Variable types Computation 506 Date 496 Multiple Choice 501 Number 493

Personal Information 508 Text 489 True/False 498 Variable usage, determining 987 Variables about 487 adding comments to 525 adding resources to 520 adding to prompts, additional text 532 asking specifically 706, 843 attaching for fillable fields 482 attaching to form template fields 425 attaching to single-character boxes in a form 437 automatically disabling in interview 601 controlling how answers appear in document 517 controlling how HotDocs processes 515 copy existing to create new 640, 811 creating and inserting in template 319 creating in Component Manager 322 deleting from current component file 636, 806 deleting from multiple component files 683 editing 510 fields associated with 319 grouping into dialogs 561 irrelevant 601 naming 538 renaming 638, 808 repeating 346 seeing which components use which 522, 569 setting answer field widths for 516 showing names for in form fields 450 viewing resources during assembly 109 Verifvina accurate information in form document before ending interviews 139 accurate information in text document before ending interviews 120 Vertically aligning form fields 430 Viewing a test assembly in a browser 1020 answer file histories and contents 190 answers used to assemble a document 179 assembled form documents 139 assembled text documents 120 contents of Template Manager window 669 form page height or page width 413

form page in more than one window 418 interview outline for form documents 248 lists of documents recently assembled 103 questions used in templates 178 saved text or form documents 128 spreadsheets of questions and answers 180

W

Warn when unanswered option 515 Warnings displaying after choosing not to 267 when adding assembly to queue 224 when answers are reformatted 211 when editing shared component files 267 when fields overflow 237 when overriding a field's answer 237 when printing via the word processor 221 when typing in inactive fields 237 when using unsupported features in HotDocs Server 1019 when variables are unanswered 220 Warnings tab viewing 989 Web accessing from variable prompts, dialogs 964 accessing published files from 193 adding addresses to libraries 394 registering browser helper applications 193 starting HotDocs assemblies from browsers 193 uploading answer files to Web servers 199 viewing templates in browser 1020 Web destinations for published files 1035 Web-based HotDocs, overview 1013 Weekdays, determining dates based on 744, 882 When irrelevant variable property 601 WHILE EXPRESSION 725, 863 White space, adding to a dialog 579 Width or height of form field, adjusting 155 Width or height of form field, adjusting in Automator 433 Widths specifying for database table columns 946 specifying for regular answer fields 516 specifying for spreadsheet columns 516 Windows accessing answer files with Explorer 186

accessing saved documents with Explorer 128 customizing HotDocs layout 95 viewing forms in multiple 418 Windows Explorer command, issuing for templates 72 Wizards, attaching to form fields 441 Word clause libraries based on 393 creating templates for 295 template conversions 298 Word processor tables, using to display lists 355 Word processors changing settings for working with 276 installing new ones after installing HotDocs 274 sending assembled clauses to open 105 sending text documents to 127 viewing saved documents in 128 WordPerfect clause libraries based on 393 creating templates for 295 template conversions 298 template conversions, in template manager 676

WPT file name extension 49
Wrapping an answer across two pages 169
Wrapping an answer across two pages, in Automator 465
Write access to answer files, allowing 186
Write-back allowing in Time Matters answer sources 588 allowing with database components 942

Y

YEAR OF(DATE) 781, 919 Year portion of a given date, finding 781, 919 Years between current date and given date 735, 873 Years between two dates 782, 920 YEARS FROM(DATE, DATE) 782, 920 Yes/no questions 498

Ζ

ZERO(NUM_VAR) 783, 921 Zooming in on a form 413