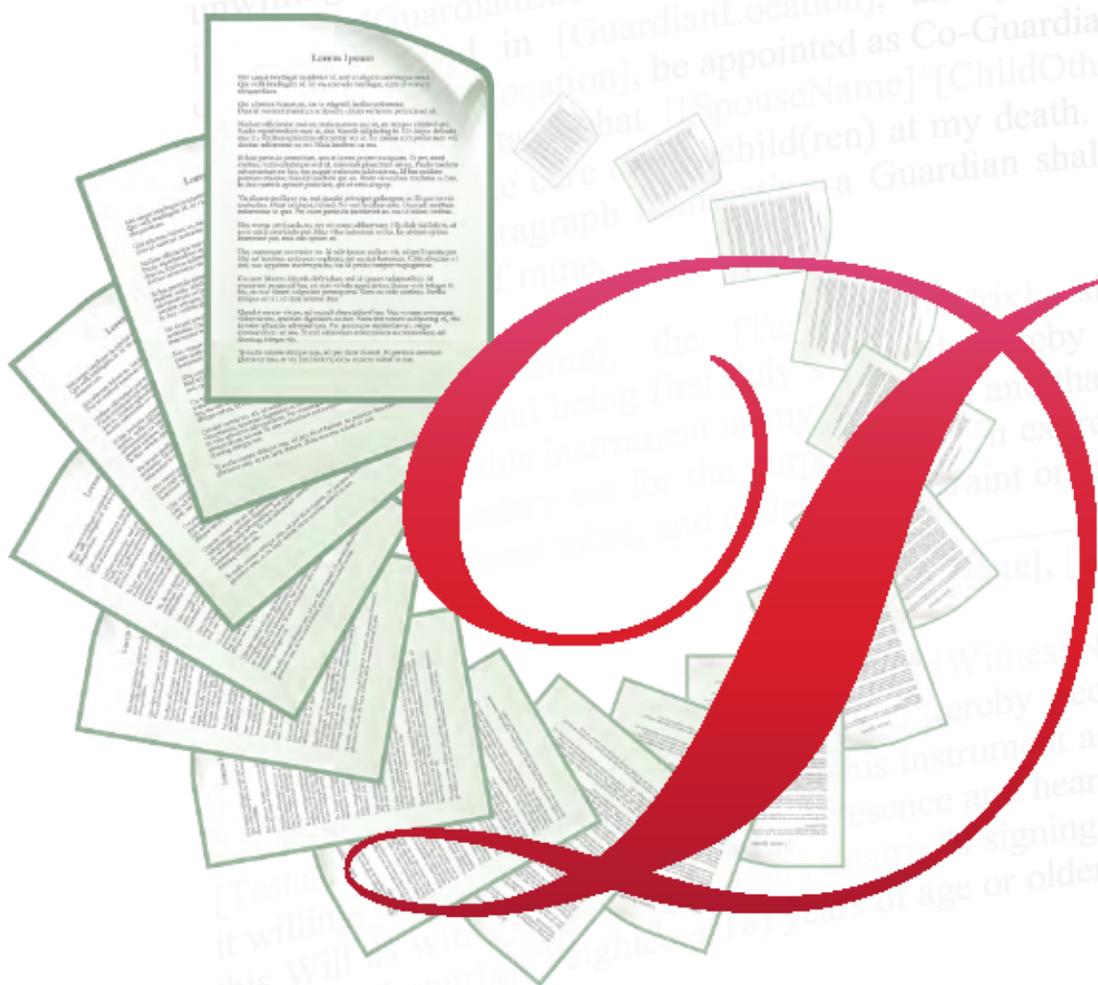


# HotDocs®

## 2008 Professional Edition

### HotDocs Help



# Copyright Information

## Copyright

Copyright © 2010 HotDocs Limited.

All rights reserved. No part of this product may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form by any means without the express written permission of HotDocs Limited ("HotDocs")

## Warranty Information

HotDocs makes no representations or warranties with respect to the contents or use of this product and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Information in this document is subject to change without notice and does not represent a commitment on the part of HotDocs.

## Government Use

Use, duplication, or disclosure by the Federal Government is subject to restrictions as set forth in FAR clauses 52.227--14, "Rights in Data--General"; 52.227--19, "Commercial Computer Software--Restricted Rights"; and subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause DFAR 252.227--7013; and the limitations set forth in the standard commercial license agreement for this software. Unpublished rights are reserved under the copyright laws of the United States.

## Trademark Information

HotDocs is a registered trademark of HotDocs Limited. Other product names may be trademarks or registered trademarks of their respective companies.

## Sample Templates and Forms

This product includes sample templates and documents. Using them may have significant legal implications in some situations, and these implications vary by state and depending on the subject matter. Before using these templates and documents or adapting them for your business, you should consult with a lawyer and financial advisor.

# HotDocs Help Table of Contents

Copyright Information .....	i
<b>HotDocs Help .....</b>	<b>1</b>
Get Help While Using HotDocs .....	1
<b>New Features Lists .....</b>	<b>2</b>
New and Enhanced Features of HotDocs 6 .....	2
New and Enhanced Features of HotDocs 6.1 .....	6
New and Enhanced Features of HotDocs 6.1 SP1 .....	8
New and Enhanced Features of HotDocs 6.2 .....	9
New and Enhanced Features of HotDocs 6.2 SP1 .....	12
New and Enhanced Features of HotDocs 2005 .....	13
New and Enhanced Features of HotDocs 2005 SP1 .....	15
New and Enhanced Features of HotDocs 2005 SP2 .....	17
New and Enhanced Features of HotDocs 2005 SP3 .....	18
New and Enhanced Features of HotDocs 2006 .....	19
New and Enhanced Features of HotDocs 2006 SP1 .....	23
New and Enhanced Features of HotDocs 2007 .....	24
New and Enhanced Features of HotDocs 2007 SP1 .....	25
New and Enhanced Features of HotDocs 2008 .....	26
<b>Getting Started .....</b>	<b>28</b>
What is HotDocs? .....	28
System Requirements .....	29
Default File Locations in HotDocs 2008 .....	30
Install HotDocs .....	32
Understand HotDocs Installation .....	33
Uninstall HotDocs .....	35
Start and Exit HotDocs .....	36
Compatibility of HotDocs 2008 Files with Earlier Versions of HotDocs .....	37
Contact HotDocs Sales and Support .....	38
Verify HotDocs 5 Templates for Use with HotDocs 2008 .....	39
List of HotDocs File Name Extensions .....	40
Use the Keyboard to Work in HotDocs .....	42
List of Keyboard Shortcuts .....	44
Using HotDocs with Microsoft Word 2007 .....	61
Complete the HotDocs Tutorials .....	62
Documentation Feedback .....	63
<b>Setting HotDocs Options .....</b>	<b>64</b>
Overview: Customize HotDocs .....	64
Display Repeated Dialogs Using a Special Icon .....	65
Control How Child Dialogs Appear in the Interview Outline .....	66
Control the Tab Order of Buttons in Navigation Bar .....	67
Select Existing Answers When Tabbing Between Answer Fields .....	68
Control the Functionality of the Enter Key During Assembly .....	69
Display Resource Buttons Next to Answer Fields .....	70
Warn When HotDocs Reformats Variable Answers .....	71
Specify How HotDocs Should Process the Date Order .....	72
Control How HotDocs Handles Two- and Four-Digit Years .....	73
Change the Way Dates Appear in Answer Fields .....	74
Customize the Look of the Dialog Pane .....	75
Customize the End of Interview Dialog .....	76
Control What Happens When You Finish an Interview .....	77
Make Assembly Window Toolbars Automatically Wrap .....	78
Display Warning When Questions are Unanswered .....	79
Prompt to Save the Assembled Document When Closing the Assembly Window .....	80
Warn When Printing Assembled Text Documents .....	81

Prompt to Select an Answer File Before Assembly .....	82
Specify a Default Date Format .....	83
Format Unanswered Variables in a Document .....	84
Set Properties for Viewing Answers and Editable Text in the Assembled Document .....	85
Warn When Adding Assemblies to Assembly Queue .....	86
Include List of Completed Assemblies in Assembly Queue .....	87
Specify How Documents Should Be Marked Up .....	88
Define Colors for Text Template Fields and Instructions .....	91
Change the Units of Measurement .....	92
Change Colors in HotDocs Automator .....	93
Control How Forms Appear when Opened .....	94
Control Warnings During Assembly .....	95
Save Printer Information for Printing Forms .....	96
Specify How PDF-based Forms Should Be Printed .....	97
Protect Forms By Backing Them Up .....	98
Set Properties for New Edit Fields .....	99
Set Properties for New Check-Box Fields .....	101
Set Rules for Handling Answer Overflow .....	103
Specify Addendum References and Labels for Answer Overflow .....	104
Generate Default Interviews for Form Templates .....	105
Select Paper Size and Paper Source For Printing Forms .....	106
Customize Appearance of Addenda .....	107
Set Preferences for Saving Comparison Snapshots .....	108
Save Edited Document Text in Comparison Snapshots .....	109
Specify Colors for Marking Changes in Comparisons .....	110
Change the Way Question and Answer Summaries Appear .....	111
Control Which Variables are Asked in a Question Summary .....	112
Enter or Edit Your Personal Information .....	113
Change Your Spell Checking Options .....	115
Have HotDocs Automatically Complete Component Names .....	117
Have HotDocs Always Generate Dialog Element Component Names .....	118
Choose Icon Style for Component Lists .....	119
Control Whether Returns Are Inserted After Instructions in Text Templates .....	120
Enable Document Editing Options for All New Templates .....	122
Enable All New Templates for Use with HotDocs Server .....	123
Display HotDocs Warning Messages .....	124
Change Script Editing Options .....	125
Set Template Manager Options .....	127
Install Support for New Word Processors .....	128
Change Your Default Word Processor .....	129
Change Word Processor File Locations .....	130
Change HotDocs Program File Locations .....	131
Assign Reference Paths to HotDocs Files .....	132
Manage Answer Files .....	134
Manage Assembled Documents .....	135
Determine How Frequently HotDocs Checks for Template Set Updates .....	136
Control How PDF Documents Are Saved .....	137
Set Properties for the HotDocs PDF Driver .....	138
Create and Edit PDF Watermarks .....	140
<b>Using HotDocs Libraries .....</b>	<b>141</b>
Overview: Use HotDocs Libraries .....	141
Create a Library .....	143
Open a Library .....	144
Add Templates and Other Files to a Library .....	145
Work with Templates and Other Files in a Library .....	146
Remove Items from a Library .....	148
Change the Properties of a Library Item .....	149
Change the Properties for Multiple Items in a Library .....	150
Sort Items in a Library .....	152
Organize the Contents of a Library .....	153

Create a Favorites Folder in the Template Library .....	155
Create a New Library by Exporting Part of an Existing Library .....	156
Import One Library into Another .....	157
Print a List of Items in a Library .....	158
Save the Contents of a Library as a Text File.....	159
Move Items Within a Library .....	160
Move Items to New Locations on the Local Disk or Network.....	161
Copy Templates to New Locations.....	162
Check for Template Set Updates .....	163
Add a Web Page to a HotDocs Library.....	164
<b>Automating Text Templates .....</b>	<b>165</b>
Overview: Template and Component Files.....	165
HotDocs Editing and Navigation Toolbar Buttons.....	167
Create a New Text Template File.....	170
Create a Text Template Based on an Existing Template .....	172
Convert a Single Template to a New File Format.....	173
Edit a Template .....	174
Assign Colors to Fields and Instructions in Templates .....	175
Move Between Fields in a Template .....	176
Use Labels to Identify Instructions.....	177
Go to a Specific Field in the Template.....	179
View the Template in Markup View .....	180
View an Outline of Scripting in the Template.....	181
Create a Foreign Language Template.....	183
Open and Close Component Manager.....	185
Open Component Manager at the Template Library .....	186
Use Component Manager to Work with Components .....	187
Create and Edit Multiple Components Simultaneously.....	190
Change Component File Properties .....	191
Delete a Component from the Current Component File .....	196
Copy Components From One File to Another .....	197
Rename Components in a Single Template .....	198
Make a Duplicate Copy of a Variable .....	200
Edit Formats, Merge Text, Dialog Elements, and Patterns.....	201
Search, Print, and Spell Check Components .....	203
Use One Component File for Multiple Templates.....	206
Specify Whether Component File Properties are Shared .....	208
Make Templates Stop Sharing Component Files .....	209
Control Whether Users See the End of Interview Dialog .....	210
Position the Cursor in the Assembled Document .....	211
Specify a Word Template for Storing Post-Assembly Macros .....	212
Remove Hidden Data from Assembled Documents .....	213
Remove Hidden Data from Word RTF Templates.....	215
<b>HotDocs Interviews.....</b>	<b>216</b>
Overview: HotDocs Interviews.....	216
Create an Interview Template .....	218
Have HotDocs Generate a Default Interview.....	220
Define a Custom Interview .....	221
Automatically Disable Irrelevant Variables in Interviews .....	223
<b>Inserting Variables.....</b>	<b>225</b>
Overview: Create and Customize Variables .....	225
Insert a Variable Field in a Text Template.....	226
Replace Other Instances of Selected Text with a Variable.....	228
Create and Insert a Variable Using Component Manager.....	229
Add a Comment to a Variable or Instruction Field .....	230
Edit a Variable .....	231
Customize a Text Variable .....	232
Customize a Number Variable.....	234

Customize a Date Variable.....	236
Customize a True/False Variable.....	238
Customize a Multiple Choice Variable.....	240
Customize a Computation Variable.....	243
Insert a Personal Information Variable.....	245
Create a Date Variable that Inserts the Current Date.....	246
Merge the Name of the Answer File in the Document.....	247
Use Automatic Paragraph Numbering.....	248
Count the Number of Entries in a List.....	249
Format the Variable.....	250
Create a Prompt for a Variable.....	252
Control How HotDocs Processes a Variable.....	253
Specify the Width of Answer Fields in the Interview.....	254
Control How Answers Appear in the Assembled Document.....	255
Automatically Disable Irrelevant Variables in Interviews.....	257
Add Resource Information to a Variable or Dialog.....	259
View Relationship Between the Current Component and Other Components.....	262
Add a Comment to a Variable or Instruction Field.....	263
Create a Pattern for a Text Variable.....	264
Specify Number Variable Formats, Percentages, and Fractions.....	265
Work with Multiple Choice Options.....	266
Specify Merge Text Options as Default or Field-Specific Properties.....	267
Copy and Paste Columns in a Multiple Choice Variable Spreadsheet.....	268
Name a Group of Merge Text Values.....	269
Work with Variables in Headers, Footers, Footnotes, and Text Boxes.....	270
Use Variables and Scripts in Prompts, Dialog Element Text, and Plain Text Resources.....	271
Automatically Assign Answers to a Variable.....	272
Values in SET Instructions.....	273
Use SET Instructions to Help Write Computations.....	274
Tips on Naming Your Variables.....	275
How Example Formats are Interpreted.....	277
Understand How Component Titles and Prompts Are Used.....	285
<b>Overview: Dot Codes.....</b>	<b>287</b>
Add Punctuation and Capitalization to Sentences.....	288
Change Font Properties of Text.....	290
Insert Characters in Text Strings.....	292
Punctuate Non-Repeated Lists of Answers.....	294
<b>Grouping Variables in Dialogs.....</b>	<b>296</b>
Overview: Create Your Own Dialogs.....	296
Gather Questions into a Custom Dialog.....	298
Edit a Custom Dialog.....	301
Change a Dialog's Options.....	302
Change the Layout of Variables in a Dialog.....	305
Use the Same Variable or Clause in Two or More Dialogs.....	306
Control Whether Dialogs are Asked Automatically.....	307
Add Text to Your Dialogs.....	308
Add the Ability to Launch an Application from a Dialog.....	309
Add a Link to a Dialog That Runs a Computation.....	310
Add a Hyperlink to a Dialog.....	311
Add Spacing and Separator Lines to a Dialog.....	313
Add a Graphic File to a Dialog.....	314
Control When Your Dialogs Appear.....	315
Put ASK Instructions in a Computation Variable.....	316
Use ASK Instructions to Control the Interview.....	317
Insert Dialogs into Dialogs.....	319
Group Child Dialogs in a Parent Dialog.....	320
Use Scripts to Add Power to Your Dialogs.....	321
Special Instructions for Dialog Scripts.....	322
Suggest an Answer Source for Dialogs.....	323

Share Answers Between Two Dialogs .....	325
Let Users Retrieve Answers from a Time Matters Record.....	326
Let Users Retrieve Answers from an Outlook Contacts List .....	327
<b>Using Repeat Instructions.....</b>	<b>328</b>
Overview: Include Lists in Your Documents .....	328
Create a REPEAT Instruction to Gather a List of Answers.....	330
Edit a REPEAT Instruction .....	331
Choose a Presentation Style for the Repeated Dialog .....	332
Customize Repeat Titles .....	333
Punctuate a Sentence-Style List .....	335
Sort a List of Answers .....	336
Filter a List of Answers .....	337
Control How Many Rows are Visible in a Spreadsheet Dialog .....	338
Limit the Number of Answers Allowed for a Repeated Dialog .....	339
Get Sum Totals for Repeated Number Variables .....	340
Use a Word Processor Table to Display a List .....	341
Make a List Appear as One or More Columns.....	342
Count the Number of Entries in a List.....	343
Retrieve a Specific Answer From a List.....	344
Retrieve Information Other Than Answers from a REPEAT Instruction .....	345
Understand How Smart Returns Are Inserted After Instructions .....	346
Create a REPEAT Instruction Using a Computation Variable .....	349
Punctuate a List Using a Computation Variable .....	350
Sort a List Using a Computation Variable.....	351
Filter a List Using a Computation Variable .....	352
Tips on Using Computations to Create Lists .....	353
Overview: Create Lists Within a List.....	354
Create a List Within a List.....	355
Display a Child Dialog Directly on the Parent Dialog .....	356
Nest REPEAT Instructions Using a Computation Variable .....	357
Use a Particular Answer from a Sublist .....	358
<b>Using IF Instructions and Expressions.....</b>	<b>359</b>
Overview: Make Parts of Templates Conditional.....	359
Condition Text Using True/False Variables .....	360
Condition Text Using a True/False Expression .....	362
Edit an IF Instruction or Expression.....	363
Nest IF Instructions .....	364
Use Conditional Instructions in a Computation Script.....	365
Tips on Using IF Instructions.....	366
<b>Using INSERT Instructions .....</b>	<b>368</b>
Overview: Insert Templates into Templates .....	368
Make Part of a Template into an Inserted Template.....	369
Insert an Existing Template .....	370
Edit an INSERT Template Instruction.....	371
Inserted Templates and Shared Component Files .....	372
Use Headers, Footers, and Margins in Inserted Word Templates.....	373
Start a New Assembly From a Template .....	375
<b>Using SPAN Instructions.....</b>	<b>377</b>
Overview: SPAN Fields .....	377
Allow Users to Edit the Text of an Assembled Document.....	378
Allow Users to Enter New Text in the Assembled Document .....	379
<b>Using Clauses .....</b>	<b>380</b>
Overview: Use Clauses and Clause Libraries .....	380
Understand How Clauses Work .....	381
Open and Close a Clause Library .....	382

Create a Clause Library at the HotDocs Library .....	383
Add a Folder to a Clause Library .....	384
Create Clauses at the Clause Library .....	385
Add Existing Clauses to a Clause Library .....	386
Customize a Clause Library .....	387
Create Clauses at the Template .....	389
Insert a Clause Library into a Template .....	391
Edit an Existing Clause .....	392
Group Clause Components in One Custom Dialog .....	393
Convert Clauses to New File Formats .....	394
<b>Using Instruction and Expression Models .....</b>	<b>396</b>
Overview: Instruction and Expression Models .....	396
Understand the HotDocs Scripting Language .....	398
Use Operators When Scripting .....	400
Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts .....	403
Use the Script Editor .....	404
<b>Instruction Models .....</b>	<b>407</b>
ADD TEXT TO MULT_CHOICE; CLEAR MULT_CHOICE .....	407
ASCEND VAR; DESCEND VAR .....	409
ASK DIALOG .....	410
ASK VAR .....	411
ASSEMBLE "FILENAME" .....	412
CONCEAL VAR .....	413
DEBUG .....	414
DEFAULT VAR TO VALUE .....	415
ERASE VAR ERASE DIALOG .....	416
FILTER COMPUTATION_VAR .....	417
FORMAT "LIST_FORMAT" .....	418
GRAY ALL, GRAY VAR UNGRAY ALL, UNGRAY VAR .....	419
HIDE ALL, HIDE VAR SHOW ALL, SHOW VAR .....	420
IF EXPRESSION ELSE IF ELSE END IF .....	421
INCREMENT NUM_VAR DECREMENT NUM_VAR .....	423
INSERT "FILENAME" .....	424
LANGUAGE CODE .....	425
LIMIT NUM .....	426
OMIT VAR .....	427
PLAY "MACRO" .....	428
QUIT .....	429
REPEAT DIALOG END REPEAT .....	430
REQUIRE ALL REQUIRE VAR .....	431
SET VAR TO VALUE .....	432
WHILE EXPRESSION END WHILE .....	433
<b>Expression Models .....</b>	<b>435</b>
Enter a Date .....	435
Enter a Number .....	436
Enter some Text .....	437
Enter True or False .....	438
ABSOLUTE VALUE( NUM ) .....	439
AGE( DATE ) .....	440
ANSWERED( DIALOG ) .....	441
ANSWERED( VAR ) .....	442
COUNT( DIALOG ) .....	443
COUNT( MULT_CHOICE_VAR ) .....	444
COUNTER .....	445
DATE - NUM DAYS .....	446
DATE - NUM MONTHS .....	447
DATE - NUM YEARS .....	448
DATE + NUM DAYS .....	449

DATE + NUM MONTHS .....	450
DATE + NUM YEARS .....	451
DATE OF( NUM, NUM, NUM ) .....	452
DAY OF( DATE ) .....	453
DAY OF WEEK( DATE ) .....	454
DAYS FROM( DATE, DATE ) .....	455
FIRST( TEXT, NUM ) .....	456
FORMAT( VALUE, "EXAMPLE" ) .....	457
INTEGER( TEXT ) .....	458
LAST( TEXT, NUM ) .....	459
LENGTH( TEXT ) .....	460
MAX( NUM, NUM ) .....	461
MID( TEXT, NUM, NUM ) .....	462
MIN( NUM, NUM ) .....	463
MONTH OF( DATE ) .....	464
MONTHS FROM( DATE, DATE ) .....	465
MULT_CHOICE = TEXT; MULT_CHOICE != TEXT .....	466
NOT TRUE_FALSE .....	467
OTHER( MULT_CHOICE_VAR ) .....	468
POSITION( TEXT, TEXT ) .....	469
REMAINDER( NUM, NUM ) .....	470
POWER( NUM, NUM ) .....	471
REPLACE( TEXT, TEXT, TEXT, NUM ) .....	472
RESULT .....	473
ROUND( NUM, NUM ) .....	474
SELECTION( MULT_CHOICE_VAR, NUM ) .....	475
SPACE( TEXT, TEXT ) .....	476
STRIP( TEXT, TEXT, TRUE_FALSE, TRUE_FALSE ) .....	477
SUM( COMPUTATION_VAR ) .....	478
SUM( NUM_VAR ) .....	479
TEXT CONTAINS TEXT .....	480
TODAY .....	481
TRUNCATE( NUM, NUM ) .....	482
UNANSWERED .....	483
UNION( MULT_CHOICE_VAR, MULT_CHOICE_VAR ) .....	484
VALUE( VAR, EXPRESSION ) .....	485
YEAR OF( DATE ) .....	486
YEARS FROM( DATE, DATE ) .....	487
ZERO( NUM_VAR ) .....	488
<b>Testing Automation .....</b>	<b>489</b>
Overview: Test HotDocs Templates .....	489
Test Individual Variables .....	490
Test a Computation Script .....	491
Test a Custom Dialog .....	492
Test Assemble a Document .....	493
Use a Test Answer File .....	494
Simultaneously Edit a Template and Test Assemble a Document .....	495
Resolve Syntax Errors in a Template or Script .....	496
Go to a Specific Field in the Template .....	497
Overview: Use the Test Panel .....	498
Validate Variable Usage in a Template .....	499
Check Interviews for Improper Scripting .....	500
Move From Document to Template During Testing .....	503
<b>Use the HotDocs Debugger .....</b>	<b>504</b>
Overview: Debugging Templates .....	504
Insert Debugging Instructions in Templates and Scripts .....	505
Step Through a Template or Script .....	507
Add Variables to the Debugger Watch List .....	510

<b>Automating HotDocs Server Templates.....</b>	<b>511</b>
Overview: Create Templates for Use with HotDocs Server .....	511
Differences Between Desktop and Browser-based Interviews .....	512
Enable Templates for Use with HotDocs Server .....	517
Add a Button to a Dialog that Links to a Database .....	519
View an Interview in a Web Browser .....	520
Publish Templates for Use with HotDocs Server .....	521
Upload HotDocs Server Files to a Web Server .....	522
<b>Automating Form Templates .....</b>	<b>523</b>
HotDocs Automator Toolbar Buttons .....	523
Start and Exit HotDocs Automator .....	527
Move Around in a Form .....	528
Create a Form Template .....	529
Save Existing HFT Files as PDF-based Form Templates .....	531
Edit a Form Template .....	532
Add Pages to a Form Template .....	533
Remove Pages from a Form Template .....	534
Update the Underlying Text in a Form Template .....	535
View the Form in More than One Window .....	536
Globally Change Properties in a Template or Group of Templates .....	537
Save and Close a Form Template .....	539
Add Templates and Other Files to a Library .....	540
Define the Interview for a Form Template .....	541
Have HotDocs Generate a Default Interview .....	542
Define a Custom Interview .....	543
<b>Form Fields.....</b>	<b>545</b>
Create a Form Field .....	545
Attach a Variable to a Field .....	546
Attach a Variable to Single-Character Boxes .....	547
Remove a Variable from a Field .....	548
Show and Hide Variable Names and Field Backgrounds .....	549
Create a Form Field .....	550
Select a Field .....	551
Create a Check-Box Field .....	552
Attach a Variable to a Field .....	553
Detect Borders to Create or Resize a Field .....	554
Move a Field on a Form .....	555
Align Two or More Fields .....	556
Change the Borders and Margins of a Field .....	557
Copy One or More Fields .....	558
Resize a Field .....	559
Change the Tab Order of Fields .....	560
Group Fields So Answers Can Flow From One Field to Another .....	561
Ungroup Fields .....	562
Attach an Answer Wizard to a Field .....	563
<b>Change Field Properties.....</b>	<b>564</b>
Preview the Formatting of Answers in a Field .....	564
Show and Hide Variable Names and Field Backgrounds .....	565
Change the Field Type .....	566
Change the Font Used for a Field .....	568
Format Lines and Paragraphs of Text in a Field .....	569
Rotate Answers in a Field .....	570
Keep Contents of Fields from Printing .....	571
Create a Field for Behind-the-Scenes Tasks .....	572
Create a Resource Link on a Form .....	573
Cross Out Static Text on the Form .....	574
Circle Static Text on a Form .....	575

Insert a Graphic File in a Form .....	576
Convert Answers on the Form to Bar Code Format .....	577
Flow a Single Answer Across Two or More Pages in a Form .....	580
Merge a List of Answers in a Standard Table .....	581
Merge a List of Answers in a Nonstandard Table .....	582
Change the Tab Order of Fields .....	584
Make a Field Conditional .....	585
Attach an INSERT Instruction to a Field .....	586
<b>Field Overflow and Addenda.....</b>	<b>587</b>
Define Overflow Properties for a Field.....	587
Split a Multi-Line Answer Between the Form and the Addendum.....	591
Customize the Look of the Addendum .....	592
Use Answer Overflow and Addendum Text Codes .....	594
<b>Automating Interview Templates .....</b>	<b>598</b>
Create an Interview Template .....	598
Define a Custom Interview .....	600
<b>Linking Templates to a Database .....</b>	<b>602</b>
Overview: Link Templates to a Database.....	602
Use Supported Databases .....	603
Understand Database Connectivity .....	604
Specify Connection Information Using ADO.....	606
Specify Connection Information Using ODBC .....	608
Save ADO Connection Information in a Separate File .....	610
Convert ODBC-based Connections to ADO .....	611
Edit a Database Component .....	612
Link Variables to Database Fields .....	613
Remember Selected Records .....	614
Save Changed Answers Back to the Database.....	615
Change Database Component Properties.....	616
Use a Custom Command to Retrieve Data from a Database .....	618
Choose Which Database Fields Appear During Assembly.....	619
Cause HotDocs to Update Selected Records Between Assemblies .....	620
Limit the Number of Database Records from Which a User Chooses.....	621
Specify a User Name and Password for a Database .....	623
Sort Records in a Database Table .....	624
Generate a List of Records in an Assembled Document .....	625
Tips on Filtering and Sorting Repeated Database Components.....	626
Merge Filtered Records into the Document Automatically.....	627
Ask Variables Already Answered by the Database .....	628
Select Related Database Records .....	629
Control When Your Database Tables Appear .....	630
Choose a Database Cursor .....	631
<b>Using Template Manager.....</b>	<b>633</b>
Overview: Template Manager .....	633
Open and Close Template Manager .....	635
View the Contents of a Template Manager Window.....	636
Work with Template Manager.....	639
Export Contents of a Template Manager Window to a Spreadsheet.....	640
Convert Multiple Templates to Work with HotDocs 2008.....	641
Convert Templates and Clauses to Microsoft RTF.....	642
Rename Templates Using Template Manager.....	644
Copy and Paste Components Across Multiple Component Files .....	645
Rename Components Across Multiple Component Files .....	646
Delete Components from Multiple Component Files .....	647
Modify Component File Properties Across Multiple Files.....	648

<b>Publishing Templates .....</b>	<b>649</b>
Overview: Publish HotDocs Templates .....	649
Publish a Template as a Standard File .....	650
Publish a Template or a Template Set as an Auto-Install File (HDI) .....	651
Publish a Template or a Template Set as an Auto-Assemble File (HDA) .....	654
Publish Templates for Use with HotDocs Server .....	656
Protect Published Files .....	657
Register Published Templates for Use With HotDocs Player .....	658
Create a Shortcut for an Auto-Install Library .....	659
Create an HDI File for a CD-based Template Set .....	660
Add Additional Files to a Published Set .....	661
Save Publishing Settings .....	662
Use a Custom Library Splash Screen .....	663
Customize the Properties Tab of the Library Window .....	664
Automatically Update Published Template Sets .....	667
<b>Uploading Published Files to a Web Server .....</b>	<b>671</b>
Overview: Upload HotDocs Files to a Web Server .....	671
Create Custom HTML Documents for Template Uploading .....	672
Upload Templates to a Web Server .....	674
<b>Uploading Answer Files .....</b>	<b>675</b>
Overview: Upload HotDocs Answer Files .....	675
Specify Options to Upload Answers .....	676
Handle Data in Uploaded Answers .....	677
<b>Using Command-Line Options .....</b>	<b>679</b>
Overview: Command-Line Options .....	679
Answer File .....	681
Answer Summary .....	682
Clause Name .....	683
Default Answer File .....	684
Discard Answers .....	685
Edit Template .....	686
Exit HotDocs .....	687
Finish Interview Action .....	688
Hide Library .....	689
HotDocs Auto-Assemble File .....	690
HotDocs Auto-Install File .....	691
Interview Scope .....	692
Keep Interview Group .....	693
Library File .....	694
Lock Answer File .....	695
Lock Library .....	696
Model Document .....	697
New Answer File .....	698
No Assembly Window .....	699
No Interview .....	700
Output File .....	701
Overlay Answer File .....	702
Paper Size .....	703
Paper Tray .....	704
Print .....	705
Print Answers Only .....	706
Print Both .....	707
Print Copies .....	708
Print Duplex .....	709
Print Form Only .....	710
Print Without Dialogs .....	711
Question Summary .....	712

Save Answers .....	713
Save Answers Prompt .....	714
Send to Word Processor .....	715
Show Library .....	716
Start Interview Group .....	717
Suggest Save .....	718
Suggest Save New .....	719
Suppress Unanswered Warning .....	720
Template File .....	721
<b>Assembling Documents .....</b>	<b>722</b>
Overview: Assemble a Text or Form Document .....	722
Adjust HotDocs Windows and Panes .....	724
Update Your Interview Outline and Document .....	725
Preview a Template Before Assembling It .....	726
Search for a Specific Template in a Library .....	727
Assemble a Text or Form Document .....	728
Select an Answer File for Assembly .....	729
Assemble an Interview Template .....	730
Assemble a Document Without Answering Any Questions .....	731
Merge Existing Answers into the Current Answer File .....	732
Assemble Multiple Documents .....	733
Use the Assembly Queue .....	734
Assemble a Clause Library from the Template Library .....	736
Navigate Through Answer-Gathering Dialogs .....	737
Access Resources During Assembly .....	740
Save an Answer File During Assembly .....	741
Switch Answer Files During Assembly .....	743
Answer Questions in a Dialog Using an Answer Source .....	744
Use Repeated Dialogs During Assembly .....	746
Use Inserted Dialogs During Assembly .....	748
View Individual Rows in a Spreadsheet .....	749
Select Clauses During an Interview .....	750
View an Assembled Document in Markup View .....	751
Answer Date Questions .....	752
View an Answer's Variable Name .....	753
Use the End of Interview Dialog .....	754
Enter Answers in Non-English Languages .....	756
Spell Check Answers .....	757
<b>Assemble Documents Linked to a Database .....</b>	<b>758</b>
Overview: Assemble Documents Linked to a Database .....	758
Select Records During Document Assembly .....	759
Sort Database Records During Assembly .....	760
Filter Database Records During Assembly .....	761
Save Selected Records in an Answer File .....	762
<b>Working with Assembled Documents .....</b>	<b>763</b>
Print an Assembled Text or Form Document .....	763
Save an Assembled Text or Form Document .....	765
View an Assembled Document in Markup View .....	766
Save a Document as a PDF File .....	767
Assign Security Options to a PDF Document .....	769
Attach a Text or Form Document to an E-mail Message .....	771
Preview the Assembled Text Document .....	772
Edit Answers at the Document Preview Tab .....	774
Edit Document Text at the Document Preview Tab .....	776
Send the Assembled Text Document to the Word Processor .....	778
Open a Saved Text Document for Editing .....	779

<b>Comparing Assembled MS Word Documents .....</b>	<b>780</b>
Overview: HotDocs® Compare.....	780
Compare Different Versions of a Document .....	782
Navigate Through Changes in Compared Documents .....	784
<b>Working with an Assembled Form Document .....</b>	<b>785</b>
Overview: Assemble a Form Document .....	785
Enter Answers Directly at the Form Document Tab .....	786
View the Assembled Form Document .....	788
Check Fields for Answer Overflow .....	789
Override an Answer in a Form.....	791
Print a Form Document .....	792
Choose Default Form Printing Options for a Form .....	794
Troubleshooting Form Printer Problems.....	795
Edit a Saved Form Document .....	796
<b>Create and Edit Form Fields .....</b>	<b>798</b>
Create a Form Field .....	798
Select a Field.....	799
Create a Check-Box Field .....	800
Detect Borders to Create or Resize a Field .....	801
Move a Field on a Form.....	802
Align Two or More Fields.....	803
Change the Borders and Margins of a Field .....	804
Copy One or More Fields .....	805
Resize a Field.....	806
Group Fields So Answers Can Flow From One Field to Another .....	807
Ungroup Fields .....	808
Change the Field Type .....	809
Change the Font Used for a Field .....	811
Format Lines and Paragraphs of Text in a Field.....	812
Rotate Answers in a Field.....	813
Keep Contents of Fields from Printing.....	814
Cross Out Static Text on the Form .....	815
Circle Static Text on a Form .....	816
Insert a Graphic File in a Form .....	817
Convert Answers on the Form to Bar Code Format .....	818
Flow a Single Answer Across Two or More Pages in a Form.....	821
<b>Customize Form Addenda.....</b>	<b>822</b>
Define Overflow Properties for a Field.....	822
Customize the Look of the Addendum .....	826
Use Answer Overflow and Addendum Text Codes .....	828
<b>Using Answer Files.....</b>	<b>832</b>
Overview: Save Answers.....	832
Create a New Answer File.....	833
Create an XML Answer File.....	835
Create a Default Answer File.....	836
Suggest an Answer File for Every Assembly.....	837
Save an Answer File During Assembly .....	838
Switch Answer Files During Assembly .....	840
Attach an Answer File to an E-mail Message .....	841
Spell Check Answers .....	842
<b>Managing Answer Files .....</b>	<b>843</b>
Overview: Use Answer Management .....	843
Open and Close Answer File Manager.....	844
Add an Answer File to Answer File Manager.....	845
Work with Answer File Manager.....	846

Add, Modify, and Delete Folders in Answer File Manager.....	849
<b>Creating Template and Document Summaries.....</b>	<b>850</b>
Overview: Create a Summary .....	850
View a Question Summary .....	853
View an Answer Summary .....	854
View the Variable Sheet .....	855
<b>Using Published and Web-based Files .....</b>	<b>856</b>
Overview: Use Published Files on the Internet.....	856
Register HotDocs to Work with Web Browsers .....	857
Add an Auto-Assemble File on the Internet to a HotDocs Library .....	859
Download Auto-Assemble Files for Use Offline.....	860
Install a Template Set.....	861
Check for Template Set Updates .....	862
Upload Answers .....	863
Add a Web Page to a HotDocs Library.....	864
<b>Creating Model Documents.....</b>	<b>865</b>
Overview: Create a Model Document.....	865
Convert Model Documents to HotDocs Template Format .....	866
Convert HotDocs Templates to Model Documents.....	867
<b>Troubleshooting: Template Development .....</b>	<b>868</b>
'The Component File Cannot Be Used With This Version of HotDocs' Message Appearing.....	868
Can't Dynamically Add Merge Text to a Multiple Choice Variable .....	869
Changing the Order Questions are Asked in an Interview.....	870
Checking the Spelling of Library Entries.....	871
Child Dialog Always Appearing in Interview Outline Even When it Shouldn't .....	872
Choosing Which Word Processor Will Be Used By Default.....	873
Comparing the Contents of Two or More Component Files.....	874
Converting WordPerfect Templates to Word .....	875
Dot Code Formatting Doesn't Seem to Work in True/False and Multiple Choice Options .....	876
Editing Answer Source Records Causes Problems.....	877
Enabling Multiple Templates for Use with HotDocs Server At Once .....	878
Entered Answers Getting Overwritten .....	879
Entering Multiple Choice Option Prompts and Merge Text.....	880
Fixing Warnings Listed at Warnings Tab.....	881
Getting Missing END IF or Missing END REPEAT Errors .....	883
Hiding the End of Interview Dialog During the Interview.....	884
HotDocs is Creating Database File with Library File .....	885
HotDocs is Prompting to Convert the Template to HotDocs 2008.....	886
HotDocs Showing Unanswered Result When Repeating a Number Variable .....	887
Keeping Users from Editing Templates .....	888
Limiting Visible Rows but Not Limiting Number of Answers .....	889
Matching Pairs of Instructions in a Computation Script .....	890
Merging Check Boxes in a Text Template.....	891
Merging Just a Portion of a Date Instead of the Whole Date.....	892
Optional Variables Showing Up as Unanswered in Interview and Document.....	893
Punctuating a Repeated List of Answers.....	894
Receiving 'Invalid Variable Name' Errors When Creating Components .....	895
Reducing Template File Size.....	896
Renaming Components Correctly .....	897
Repeating a Single Variable in a Dialog .....	898
Repeating the Contents of a Single Cell in a Table.....	899
Test Assembling Template with ASSEMBLE Instructions .....	900
Trying to Save an Invalid Computation Script.....	901
Understanding the Difference Between Multiple Choice Options and Merge Text .....	902
Using DEFAULT Instruction Causes Extra Repetition.....	903
Using Headers and Footers of Inserted Templates .....	904
Using INSERT Instructions in a Table .....	905

Using One Variable in Two Different Dialogs .....	906
Variables Not Working in Word Text Boxes .....	907
<b>Troubleshooting: Document Assembly .....</b>	<b>908</b>
Can't Attach Answer File or Document to Outlook 2007 E-mail Message .....	908
Moving to the Next Dialog After Entering a Date or Number .....	909
Changing Unanswered Text Markers .....	910
Discerning Between Template Preview and Actual Template .....	911
Downloading HDIs and HDAs as ZIP Files.....	912
Editing Answers in Assembled Document.....	913
Entering a Two-Digit Year .....	914
HotDocs Skips Questions When Navigating from Child Dialog .....	915
Increasing the Font Size of Dialog Text.....	916
Installing Support for New Word Processor.....	917
Moving Between Questions in Interview Slow .....	918
Organizing Files in the Library vs. on the Disk .....	919
Overriding an Answer in a Form.....	920
Printing Form Document is Off-Center .....	921
Printing Form Document Takes Incredibly Long Time.....	922
Receiving Error Messages When Launching HDAs, HDIs .....	923
Restoring System Defaults to HotDocs .....	924
Saving Answers During Interview.....	925
Specifying Printing Options When Printing Text Documents .....	926
Understanding What Happens When Deleting Templates from the Library .....	927
Updating Changes Made in Word Processor Document in the Answer File, Template .....	928
Updating Cross-References in an Assembled Document .....	929
Viewing File Names Instead of Template Titles at Library.....	930
<b>Troubleshooting: HotDocs Automator.....</b>	<b>931</b>
Double-Clicking Doesn't Create Field as Expected .....	931
Fixing Page Rotation Problems in PDF Templates .....	932
Getting Errors When Automating Table in Form.....	933
Keeping Control Field Text From Appearing on Form .....	934
Modifying a Single Field in a Group.....	935
Moving Form Fields Using Keyboard Takes Too Long.....	936
Needing Characters in Answer to Appear in Individual Boxes.....	937
No Interview When Assembling Form Document.....	938
Setting Form Properties Not Taking Effect in Existing Templates .....	939
Tabbing Through Fields Sequentially in Form.....	940
Want Check Box to Use Actual Check Mark .....	941
<b>Addendum .....</b>	<b>942</b>
Names of Repeated Answers.....	942
Custom Library Properties Page Markup.....	943
Sample Code for Uploading an Answer File Using HotDocs Server .....	944
Publishing Destination Page Sample Code .....	945
Publishing Form Page Sample Code .....	946
URL-Encoded Text.....	947
Use a Custom Library Splash Screen .....	948
What is XML? .....	949
Document Text Editor Toolbar.....	950
Assembly Queue Statuses .....	952
Troubleshooting Form Printer Problems.....	953
Understand Bar Code Settings.....	954
Custom Command Sample .....	955
Tips on Filtering and Sorting Repeated Database Components.....	956
Component Properties .....	957
Tips for Working With Answer Sources .....	958
Differences Between ASSEMBLE and INSERT Instructions.....	959
Differences Between SET and DEFAULT Instructions.....	960
Foreign Language Number and Date Formats.....	961

View the Component's Status .....	962
View the Template Status.....	963
Currency Symbols .....	964
Additional Support Information for Folio Views 4.2.....	965
Tips on Naming True/False Variables .....	966
Tips on Using Unanswered Variable Placeholders.....	967
Inserted Template Locations .....	968
<b>HotDocs Glossary.....</b>	<b>969</b>
<b>Third-Party Software Acknowledgments .....</b>	<b>985</b>
<b>End-User License Agreement.....</b>	<b>986</b>

# HotDocs Help

---

## Get Help While Using HotDocs

You can use the HotDocs Help system to find information about HotDocs—including explanations of HotDocs features as well as step-by-step instructions for completing most tasks in HotDocs. The HotDocs Help system also provides dialog-level help and  **What's This** help. Dialog-level help explains the overall purpose for the dialog box you are viewing, while  **What's This** help explains specific options or properties of the dialog box.

### To access task and reference information

1. Select **HotDocs Help** at any **Help** menu. (If you are working in a text template, click the  **Help** button in the HotDocs toolbar.) The **HotDocs Helps** window appears.
2. Perform any of the following tasks:
  - Click the **Contents** tab, double-click the book icons to expand (and collapse) them, and then click on a topic to display its contents.
  - Click the **Index** tab, and type the keyword or phrase for which you are searching in the box. As you type, HotDocs displays relevant topics. At any time, select the topic that fits your search criteria and either double-click the entry or click **Display**.
  - Click the **Search** tab, and type the word or phrase for which you are searching in the box. Click **List Topics** to display the search results in the **Select Topic to display** list. Once you have located the topic you want to view in this list, double-click it or click **Display**.
    - Search for a literal phrase by including quotation marks around the phrase (for example, "add text to"). The help system will display all the topics that contain that exact phrase. It will also include any grammatical variations.
    - Use a wildcard character by including an asterisk or question mark in the word or phrase for which you are searching (for example, *explore\** or *HotDocs 5.?*). The help system will display all the topics that contain that phrase with any modifications as expressed by the wildcard character.
    - Search using the Boolean operators of *AND*, *OR*, *NOT*, or *NEAR*.

### To access dialog-level help

- At any dialog box that contains a **Help** button, click the button. HotDocs displays the information about the contents of that specific dialog box.

### To access What's This help

- At any dialog box where What's This help is available, click the **What's This** button, either in the title bar () or the toolbar (), and then click the item for which you want help.

#### Notes:

- You can bookmark topics you frequently refer to. To do this, click the **Favorites** tab and then click **Add**. The topic is added to the **Topics** list. To view it later, select it from the **Favorites** list and click **Display**.
- To print a help topic, click the  **Print Topic** button in the top navigation pane.
- To increase or decrease the font size used for viewing the help file, click the  **Increase Size** or  **Decrease Size** buttons in the top navigation pane, respectively.

# New Features Lists

---

## New and Enhanced Features of HotDocs 6

The following new features and enhancements were implemented for the release of HotDocs 6.0:

### Changes to 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, the **Edit Component** button, and the **HotDocs Help** button:

- 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  **Edit 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.

## Changes to 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.

**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.

# New and Enhanced Features of HotDocs 6.1

The following features and enhancements were implemented for the release of HotDocs 6.1:

## Changes to 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.

**Warning:** 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.

## Changes to 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.

### Changes to 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 Automator and HotDocs Filler both include 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 have PDF Advantage installed and 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 6.1 SP1

The following features and enhancements were implemented for the release of HotDocs 6.1 SP1:

## Changes to 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 (¶)**. 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.

## Changes to 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 6.2

The following features and enhancements were implemented for release of HotDocs 6.2:

## Changes to Default File Locations

Several changes were made to default file locations HotDocs uses. See [Changes to File Locations in HotDocs](#) for a complete list.

## Changes to 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 loops.

**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. 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  **Label 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  **Match Fields** button to quickly find corresponding IF, ELSE, END IF, REPEAT, and END REPEAT fields. For example, if you place your cursor in a REPEAT field and click the  **Match Fields** button, HotDocs will move your cursor to the matching END REPEAT field. This can help you troubleshoot problems you may experience as you work with these types of instructions—particularly when there are multiples levels of instructions.
- 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  **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.)

## Changes to HotDocs Automator / 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, Automator 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. (Requires HotDocs PDF Advantage.)

## Changes to 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.

# 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 2005

The following new features and enhancements were implemented for the release of HotDocs 2005:

## Changes to Template Development

**Database Connection:** HotDocs Database Connection is now included with all licenses for HotDocs Professional. 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. (See [View an Interview in a Web Browser](#).)

**Warning:** 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. (See [Use Headers and Footers in Inserted Templates](#).)

**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. (See [Control How Answers Appear in the Assembled Document](#).)

**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. (See [Format the Variable](#).)

**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. (See [Let Users Retrieve Answers from an Outlook Contacts List](#).)

## Changes to 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.)* (See [Edit Answers at the Document Preview Tab](#).)

**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.)* (See [Edit Answers at the Document Preview Tab](#).)

**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.*) (See [Overview: HotDocs Compare](#).)

**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. (See [Copy Templates to New Locations](#).)

# New and Enhanced Features of HotDocs 2005 SP1

The following new features and enhancements were implemented for the release of HotDocs 2005 Service Pack 1:

## Changes to 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). See [Change Component File Properties](#).

**Warning:** 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. See [Change Component File 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. (See [Understand How Component Titles and Prompts Are Used](#).)

**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. (See [Customize Repeat Titles](#).)

**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). (See [Copy and Paste Columns in a Multiple Choice Variable Spreadsheet](#).)

**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** box, either at the **Multiple Choice Variable Editor** or at the **REPEAT Field** dialog box. (See [How Example Formats are Interpreted](#).)

## Changes to Document Assembly

**Sort spreadsheets (including answer sources) during the interview:** You can now sort the contents of a spreadsheet in alphanumeric order. See [Answer Questions in a Dialog Using an Answer Source](#) and [Use Repeated Dialogs During Assembly](#).

**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. See [Work with Templates and Other Files in a Library](#).

**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. (Select your option at the library's **View** menu.) 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.) (See [Search for a Specific Template in a Library](#).)

## Changes to 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 SP2

The following new features and enhancements were implemented for the release of HotDocs 2005 Service Pack 2:

## 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 folders 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 SP3

The following new features and enhancements were implemented for the release of HotDocs 2005 Service Pack 3:

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

# 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. See [Allow Users to Edit the Text of an Assembled Document](#) for details.

**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. See the following topics for details: [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](#), and [Add a Graphic File to a Dialog](#). *(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. See [Remove Hidden Data from Word RTF Templates](#) for details. Additionally, you can have this data removed from a document after it is assembled. See [Remove Hidden Data from Assembled Documents](#) for details.

**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. See [Overview: Dot Codes](#) for details.

**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. See [View an Outline of Scripting in the Template](#) for details.

**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. See [Specify How Documents Should Be Marked Up](#) and [View the Template in Markup View](#) for details.

**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. See [Have HotDocs Generate a Default Interview](#) and [Define a Custom Interview](#) for details.

**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. See [Specify Whether Component File Properties are Shared](#).

**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.) See [Control Whether Returns are Inserted After Instructions in Text Templates](#) and [Understand How Returns are Inserted After Instructions](#) for details.

**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 several 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.) See [View Relationship Between the Current Component and Other Components](#) for details.

**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. See [Specify a Template for Storing Post-Assembly Macros](#) for details.

**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. See [Change Component File Properties](#) for details.

**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. See [Use Variables and Scripts in Prompts, Dialog Element Text, and Plain Text Resources](#) for details.

**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. See [Install Support for New Word Processors](#) for details.

**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. See [Customize the Look of the Dialog Pane](#) for details.

**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. See [Edit Document Text at the Document Preview Tab](#) for details.

**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. See [View an Answer's Variable Name](#) for details.

**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. See [Work with Answer File Manager](#) for details.

**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. See [Customize the Look of the Dialog Pane](#).

**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. See [Check for Template Set Updates](#) for details.

## HotDocs Automator

**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, to create a form template, you must use HotDocs PDF Advantage, Professional Edition, which lets you create and save form templates in PDF format.

**Warning:** While PDF Advantage is required for the creation of all *new* form templates, PDF Advantage is not required for editing *existing* .HFT files.

**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. See [Circle Static Text on a Form](#) for details.

**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. See [Define Overflow Properties for a Field](#) for details.

# New and Enhanced Features of HotDocs 2006 SP1

## Template Development and Document Assembly

**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. For details, see [Use the Keyboard to Work in HotDocs](#) and [List of Keyboard Shortcuts](#).

## HotDocs Automator

**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. See [Split a Multi-Line Answer Between the Form and the Addendum](#) for details.

# New and Enhanced Features of HotDocs 2007

The following new features and enhancements were implemented for the release 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. See [Using HotDocs with Microsoft Word 2007](#) for a list of some of the differences between Word 2007 and earlier versions of Word.

**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. See [View an Assembled Document in Markup View](#) for details.

**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 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. See [Insert a Web Link in a Prompt or Dialog Text Element](#) for details.

# New and Enhanced Features of HotDocs 2008

The following new features and enhancements were implemented for the release 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 to generate custom documents for your clients. See [Overview: Create a Model Document](#) for details.

## 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. See [Go to a Specific Field in the Template](#).

**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. See [Validate Variable Usage in a Template](#) for details.
- **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. See [Check Interviews for Improper Scripting](#) for details.
- **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. See [Move From Document to Template During Testing](#) for details.

**Publishing licenses now include annual expiration:** Licenses for publishing and registering templates for use with HotDocs Player now expire annually. See [Register Published Templates for Use with HotDocs Player](#) for information.

**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. See [Automatically Disable Unused Variables in Interviews](#) for details.

**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. See [Enable Templates for Use with HotDocs Server](#) for details.

## 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. For details, see [Enter Answers in Non-](#)

English Languages.

**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. See [Create a Favorites Folder in the Template Library](#) for details.

## 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.

# Getting Started

## What is HotDocs?

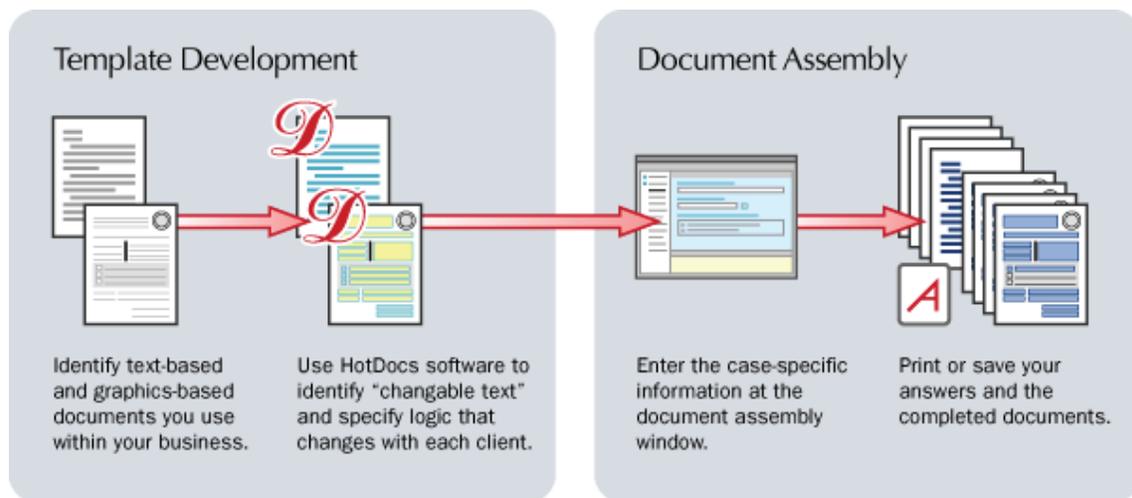
HotDocs is an award-winning software application that significantly reduces the time spent generating customized documents, such as contracts, sales proposals, government and court forms, legal documents, loan applications, and medical forms.

Using HotDocs, you can transform any PDF document or word processor file into an interactive template by marking changeable text with HotDocs variables. Then, the next time you want to generate a completed form or text 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, including lengthy and complex documents. 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.

The following diagram shows the HotDocs process, from template development to document assembly:



# System Requirements

To ensure the best performance using HotDocs, please note the following requirements for hardware and software:

## Hardware

- 900 MHz processor (2GHz, recommended)
- 256 MB RAM
- 50 MB hard disk storage
- 1024 x 768 screen resolution with at least 16-bit color

## Software

- Microsoft Windows XP, or Windows Vista
- Microsoft Internet Explorer 6.0 or later
- Microsoft Word 97, 2000, XP, 2003, or 2007; or Corel WordPerfect 8, 9, 10, 11, 12, or X3

**Warning:** Microsoft Word 97 is not supported for use with HotDocs on Windows Vista. Additionally, WordPerfect X3 SP2 is the only version of WordPerfect supported for use on Windows Vista.

## Databases

If you plan to integrate HotDocs with a database, you must have one of the following applications or servers installed:

- Microsoft Access
- Microsoft SQL Server
- Oracle 8i or 9i with the OraOLEDB OLEDB provider
- Most other ODBC-compliant databases

## Default File Locations in HotDocs 2008

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.

The following is a list of these default locations:

**Warning:** In Windows Vista, the *My Documents* folder has been renamed as *Documents*. The *All Users Documents* folder has been renamed as *Public Documents*.

File Type	Default Location
Program Files	C:\Program Files\HotDocs 6
JavaScript Files	<Program Files>\HotDocs 6\JavaScript
Help Files	<Program Files>\HotDocs 6
Spelling Dictionary File	<My Documents>\HotDocs\Spelling
Library Files	<My Documents>\HotDocs\Libraries
Template Files Includes text templates, form templates, interview templates, and auto-assemble files <b>Note:</b> Even though HotDocs saves all template types to this folder, it still maintains separate default settings for text templates, form templates, and interview templates.	<My Documents>\HotDocs\Templates
WordPerfect Macro Files	Word Processor Macro Folder
MS Word Startup Files	Word Processor Startup Folder
Template Sets	<All Users Documents>\HotDocs\Templates

Text Document Files	Word Processor Documents Folder
Form Document Files	<My Documents>
Answer Files	<My Documents>\HotDocs\Answers
Catalog Files	<All Users Documents>\HotDocs\Catalogs
Publish Settings Files	<My Documents>\HotDocs\Publish

If you need to change these default locations after HotDocs is installed, you can do so at HotDocs Options. See [Change Word Processor File Locations](#) and [Change HotDocs Program File Locations](#) for details.

**Warnings:**

- If you are upgrading from HotDocs 6.1 or earlier and your libraries are saved in the HotDocs 6 Program Files folder, HotDocs will set this folder as the default location for libraries. If you do not have library files saved here, HotDocs will use *<My Documents>\HotDocs\Libraries*.
- If you perform a custom installation, HotDocs will install files and set registry entries using the information you specify during installation. Whatever settings you specify during installation will be used for all users of the workstation.

# Install HotDocs

When you insert the HotDocs installation disk in your CD-ROM drive, a CD browser window should automatically appear. From this window, you can view the Readme and install HotDocs. Once installation has started, HotDocs prompts you for all of the information it needs to complete the process.

Before installing HotDocs, make sure your computer meets the minimum requirements necessary for using HotDocs. (See [System Requirements](#).)

## To install HotDocs

1. Close all open programs, especially any word processors or existing versions of HotDocs you may have open.
2. Insert the HotDocs installation CD into your CD-ROM drive. The CD browser window appears.
3. Click **Install HotDocs 2008 Professional Edition**. The installation program begins installing HotDocs.
4. Follow the on-screen prompts to complete the installation.

### Notes:

- To manually install HotDocs, select **Run** from the **Start** menu and type **D:\setup**. (Substitute the correct drive letter if your CD-ROM drive is not mapped to *D:*.) Follow the on-screen prompts to complete the installation.
- For a description of how HotDocs is installed, see [Understand HotDocs Installation](#).

# Understand HotDocs Installation

When you install HotDocs, two different installations actually happen—the *HotDocs Setup* and the *Current User Setup*. This two-part installation process not only allows the program be installed just once for multiple users on a single workstation, but it also allows each user who logs on to the workstation to have his or her own custom files and settings.

## HotDocs Setup

When you first install HotDocs from *Setup.exe* (see [Install HotDocs](#)), HotDocs installs the program files needed to use HotDocs, including any executable files used to run HotDocs. Among others, it also installs the HotDocs Help files and JavaScript files (which are needed for developing HotDocs Server templates). These files are installed, by default, to *C:\Program Files\HotDocs 6*. This location is not user-specific, which allows all users of a single workstation access to the files.

As HotDocs installs these files, it registers information about them in the *Local Machine* key of the Windows System Registry. For example, the Registry stores information about where the executable file for HotDocs is located on disk, as well as which word processors HotDocs supports. It also sets some preliminary file location information for the various files HotDocs uses, including information about library files and template set files. All of this information must be registered in order for HotDocs to work correctly.

## Current User Setup

When you install HotDocs, the installation creates a folder named *Source* in the HotDocs program folder. This folder contains several files, including word processor macro or startup files, the user spell-checking dictionary, and the HotDocs tutorial files. These files must be in user-specific folders on the disk. However, because multiple users may use a single workstation, HotDocs Setup doesn't install these files. Instead, when a user logs on to the workstation and runs HotDocs, the Current User Setup copies these files to the correct locations for the user. (See [Default File Locations in HotDocs](#).)

The Current User Setup also registers information about each user in the *Current User* key of the System Registry. This information includes default file locations for such things as templates, answer files, and library files. As each user specifies preferences for working with HotDocs, these settings are likewise written to the *Current User* key. This allows each workstation user to have his or her own HotDocs settings without other users overriding those preferences with their own.

The Current User Setup happens each time a user starts HotDocs. This means that each time you run HotDocs, HotDocs checks to make sure all of the required files (such as word processor startup files) are located where it expects to find them. If the file isn't found, HotDocs replaces it using a copy it finds in the *Source* folder.

**Warning:** Even if you delete the tutorial files that are installed to your default *Templates* folder, because of the Current User Setup, the next time you launch HotDocs, HotDocs will recopy the files to that folder. To keep this from happening, open the *Source* folder, select the *Tutorials* subfolder and press **Delete**.

## Restore Installation Defaults

At times, you may want to restore default settings in HotDocs. You can accomplish this by clearing the *Current User* key of the Registry.

**Warning:** 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 restore default settings

1. At the **Start** menu, select **Run**. The **Run** dialog box appears.
2. In the **Open** box, type **regedit** and click **OK**. The **Registry Editor** appears.
3. Navigate to HKEY\_CURRENT\_USER > Software > LexisNexis > HotDocs 6.
4. Select the **HotDocs 6** subfolder and press **Delete**. Click **Yes** to confirm the deletion.
5. Close the Registry Editor.

# Uninstall HotDocs

You can remove existing versions of HotDocs from your computer by using the HotDocs uninstall program. When you uninstall, all of the program files are removed from your computer. However, the program does not remove any custom templates, libraries, or answer files you have created.

## To uninstall HotDocs

1. At the **Start** menu, select **Settings > Control Panel**. The **Control Panel** dialog box appears.
2. Click **Add or Remove Programs**. The **Add or Remove Programs** dialog box appears.
3. Select the version of HotDocs that needs to be removed, then click **Remove**. The HotDocs uninstall process starts.
4. Follow the on-screen prompts to remove the program.

**Warning:** If you plan to reinstall HotDocs to restore default settings, you must delete the user-specific settings HotDocs entered in the System Registry. For details on doing this, see [Understand HotDocs Installation](#).

# Start and Exit HotDocs

To start HotDocs so you can edit templates or assemble documents, you must first open the HotDocs library. The library includes commands for performing most tasks in HotDocs.

## To start HotDocs

- Choose **Start > Programs > HotDocs 2008 > HotDocs Professional**. The HotDocs library window appears.

## To exit HotDocs

- Click **Exit** (File menu).

**Note:** To start HotDocs, you can also click the  **HotDocs** button in the word processor toolbar. This will open and display the HotDocs library window.

# Compatibility of HotDocs 2008 Files with Earlier Versions of HotDocs

## Installation of Program Files

When you install HotDocs 2008 over HotDocs 6, HotDocs 2005, HotDocs 2006, or HotDocs 2007, it overwrites that installation.

## Templates and Component Files

If you are upgrading from HotDocs 6 or HotDocs 2005, you will be prompted to convert your component files to HotDocs 2006-2008 format. (You will not be prompted to do this if you are upgrading from HotDocs 2006 or HotDocs 2007.)

After you convert the component file to the latest version, if you need to use it with an earlier version of HotDocs, you can specify which version at the **Component File Properties** dialog box of Component Manager. (See [Change Component File Properties](#).)

If you are using HotDocs 2008 to automate templates for use with an earlier version of HotDocs, do not use any new features that aren't supported in the earlier version or your templates may not work correctly. See the New Features section of this help file for a list of features implemented in each release of HotDocs.

## Answer Files

HotDocs answer files are not version-specific. They can be created and used with any version of HotDocs.

# Contact HotDocs Sales and Support

## HotDocs Technical Support

To contact technical support, please call (800) 828-8328.

Additionally, you may find answers or solutions to questions you have in the HotDocs Knowledge Base. To view the Knowledge Base (as well as access other support options), please visit <http://www.hotdocs.com/support/>.

## 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.

<b>Method of Contact</b>	<b>Information</b>
Telephone	(800) 500-3627 (sales) (801) 615-2200 (business)
Fax	(877) 356-3627
E-mail	<a href="mailto:sales@hotdocs.com">sales@hotdocs.com</a>
Web Site	<a href="http://www.hotdocs.com/products">http://www.hotdocs.com/products</a> <a href="http://www.hotdocs.com/services">http://www.hotdocs.com/services</a>
Address	387 S 520 W Suite 210 Lindon, UT 84042

# Verify HotDocs 5 Templates for Use with HotDocs 2008

When documents are assembled in HotDocs 2008, HotDocs displays an interview outline in the assembly window. This outline represents all the dialogs in the interview. Using it, users can select any dialog and answer questions in it. When they do this, HotDocs processes the answers and then updates the entire interview outline.

This type of interview processing is different than it was in HotDocs 5, where interviews were sequential, meaning answers entered in one dialog affected answers only in subsequent dialogs. This type of processing rarely required the interview to be updated entirely and at once.

Because of this change in interview processing, some scripting and other features that were used in HotDocs 5 may now cause the interview in HotDocs 2008 to process incorrectly. This can result in inconsistent answers in the interview as well as produce an incorrectly assembled document. In order to prevent such errors from going unnoticed, templates should be tested for potential problems before users assemble documents from them.

In order to verify a template, you must first convert the template to HotDocs 2008 format. (See [Convert Multiple Templates to Work with HotDocs 2008](#).) Then you must test the template. While at the test assembly window, you can view the Test Panel, which includes a **Warnings** tab that tells you of potential problems in your template, including incorrectly used SET instructions, incorrectly asked variables, and improperly referenced variables. (See [Overview: Use the Test Panel](#) for information on using the Test Panel.)

## To verify a HotDocs 5 template for use with HotDocs 2008

1. At the template library, select the converted template you want to verify and click  **Edit**. (See [Convert Multiple Templates to Work with HotDocs 2008](#).) The template appears in the template development window.
2. Test assemble the template and verify that it does not contain any automation or scripting that may cause problems. (See [Check Interviews for Improper Scripting](#).)
3. In addition to reviewing any reported warnings, make sure the converted template looks and functions correctly. For example, you should check for the following:
  - Any syntax errors reported by HotDocs have been fixed. (See [Resolve Syntax Errors in a Template or Script](#).)
  - The interview outline looks right and operates properly.
  - Dialogs look right and operate properly.
  - The document is assembled correctly.

# List of HotDocs File Name Extensions

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

<b>File Name Extension</b>	<b>Type of File</b>
.RTF	Word RTF Template
.DOT	Word Template
.WPT	WordPerfect 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	HotDocs Answer Library
.ANS	HotDocs Answer File
.ANX	HotDocs XML Answer File
.HPL	HotDocs Answer Source  <b>Note:</b> This file name extension isn't required, but since answer source files and regular answer files are stored in the same folder, many template developers choose this extension to distinguish between answer source files and regular answer files.
.JS and .HVC	JavaScript interview file and HotDocs Variable Collection file (These files are used when generating HotDocs Server interviews.)
.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.)

# 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.

**Warning:** 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, complete the following steps:

1. Press the **Tab** key to move focus to the item list of the library.
2. Press the **Up Arrow** or **Down Arrow** key to move between items in the library.
3. Once the template you want to assemble is selected, press **Alt+M** (to view the **Template** menu).
4. 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 box (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, complete the following steps:

1. At the assembly window, press **Alt+F**. The **File** menu appears.
2. Press **S**. The **Save Answer File** dialog box appears.

**Warning:** 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**.

## 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, complete the following steps:

1. While your cursor is in an answer field, press the  **Application** key on your keyboard. (Or, press **Shift+F10**.) A shortcut menu appears.
2. 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.

**Note:** 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.)

# 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.)

## Template Library Window

To Do This	Press
Tab between the item list and the Find options	Tab / Shift+Tab
Switch between <b>Properties</b> and <b>Preview</b> tabs	Ctrl+Tab
Assemble the currently selected template	Enter
Edit the currently selected template (Professional and Standard users only)	Shift+Enter
Create a new template (Professional and Standard users only)	Ctrl+Enter
Open the <b>Item Properties</b> dialog box for the currently selected folder or template	Alt+Enter
Remove the template from the library	Delete
Collapse and expand folders in the library	Left Arrow / Right Arrow
Move through items in the library list	Up Arrow / Down Arrow
Move the selected item up or down in the list	Alt+Up Arrow / Alt+Down Arrow
Copy the selected item	Ctrl+C

Create a new library	Ctrl+N
Open an existing library	Ctrl+O
Print a copy of the item list	Ctrl+P
Cut the selected item or text and place a copy of it on the Clipboard	Ctrl+X
Open the HotDocs Help file	F1
Display the What's This Help cursor	Shift+F1
Display the What's This Help text for the currently selected item	Ctrl+F1
Close the template library	Alt+F4
Move focus to a different pane in the library	F6
Move between HotDocs windows	Alt+F6
Resize the pane you are currently viewing This option selects the split bar so you can use the arrow keys to move it. Press <b>Enter</b> to set the placement.	Ctrl+F6
Access the library menu bar	F10
Display a shortcut menu	Shift+F10
Save a new copy of the library	F12

## Component Manager

To Do This	Press
Move between the different elements in Component Manager	Tab / Shift+Tab
Edit the selected component	Enter
Open the Component File Properties dialog box	Alt+Enter
Insert the selected component in the template	Insert
Remove the component from the component file	Delete
Open the <b>Components</b> drop-down list to view all available options	Alt+Down Arrow
Copy components between two open component files	Alt+Left Arrow / Alt+Right Arrow
Select all components in the list	Ctrl+A
Create a copy of a component (or group of components, if multiple components are selected)	Ctrl+C
Edit the currently selected component	Ctrl+E
Search each component in the list for a specific word or phrase	Ctrl+F
Search each component in the list for a specific word or phrase and replace it with a new word or phrase	Ctrl+H

Create a new component	Ctrl+N
Print a list of components and their properties	Ctrl+P
Close Component Manager	Ctrl+W
Open the HotDocs Help window	F1
Display the What's This help cursor	Shift+F1
Display the What's This help text	Ctrl+F1
Rename the selected component(s)	F2
Close Component Manager	Alt+F4
Move between HotDocs windows	Alt+F6
Check the spelling of text used in components	F7
Close Component Manager	Ctrl+F8
Access the Component Manager toolbar	F10
Display the shortcut menu	Shift+F10

## Script Editor

To Do This	Press
Move throughout the elements in the component editor	Tab / Shift+Tab

Merge a paragraph break in a script	Ctrl+Enter
Merge the just highlighted component or model in the <b>Script</b> box  (First select the component or keyword)	Insert
Display an Auto Complete list of keywords and component names so you can more easily enter the correct syntax for a script	Ctrl+Space
Switch between the different tabs of the component editor	Ctrl+Tab
Highlight all of the script	Ctrl+A
Copy the highlighted script to the Clipboard	Ctrl+C
Edit the currently selected component	Ctrl+E
Search through the <b>Script</b> box for a specific word or phrase	Ctrl+F  (Use F3 to find the next instance of text for which you are searching)
Go to a specific line number or character position in the script	Ctrl+G
Search through the <b>Script</b> box for a specific word or phrase and replace it with a different word or phrase	Ctrl+H  (Use F3 to find the next instance of text for which you are searching)
Indent the script text	Ctrl+I
Place comment markers before the selected text	Ctrl+K

Remove any comment markers from the selected text	Ctrl+Shift+K
Find the matching IF or REPEAT instruction (place your cursor in the instruction before issuing this command)	Ctrl+M
Highlight a block of script starting with the opening instruction and ending with the closing instruction (place your cursor in the instruction before issuing this command)	Ctrl+Shift+M
Create a new component	Ctrl+N
Insert a tab in the selected block of script	Ctrl+T
Paste the text on the Clipboard in the script	Ctrl+V
Close the <b>Computation Editor</b> without saving your changes	Ctrl+W
Cut the selected script text and place it on the Clipboard	Ctrl+X
Reapply a change you just undid	Ctrl+Y
Undo a change	Ctrl+Z
Open the HotDocs Help	F1
Display the What's This Help cursor	Shift+F1
Display the What's This Help text	Ctrl+F1
Close the component editor	Alt+F4

View an auto-complete list of just components	F5
View an auto-complete list of scripting keywords	Shift+F5
Move between HotDocs windows	Alt+F6
View a status hint for the component or keyword you are currently viewing (where your cursor is)	F7
Indent all pairs of IF and REPEAT instructions, based on their level of insertion	F8
Close the component editor without saving changes	Ctrl+F8
Access the script editor toolbar	F10
Display the shortcut menu for the script editor	Shift+F10

## Assembly Window: Interview Tab

To Do This	Press
Tab through the answer fields in a dialog	Tab / Shift+Tab
Move to the next dialog in the interview	Enter
Insert a paragraph end in a multi-line answer	Ctrl+Enter
Move to the next dialog or the previous dialog in the interview	Page Down / Page Up
Move to the next unanswered dialog or to the previous unanswered dialog in the interview	Ctrl+Page Down / Ctrl+Page Up

Move through each unanswered question in a dialog	Ctrl+Down Arrow / Ctrl+Up Arrow
View all answers in a drop-down list	Ctrl+Down Arrow
Display the calendar for a date field	Alt+Down Arrow
Collapse or expand dialogs in the interview outline	Left Arrow / Right Arrow
Use a new, untitled answer file during the current interview	Ctrl+N
Open Answer File Manager	Ctrl+O
Save the current answer file	Ctrl+S
Indent the text in a multi-line answer field	Ctrl+T
Close the assembly window	Ctrl+W
Open the HotDocs Help window	F1
Display the What's This Help cursor	Shift+F1
Display the What's This Help text	Ctrl+F1
Display the resource for a field in a pop-up window	Alt+F1
Close the HotDocs assembly window	Alt+F4
Switch between the answer file drop-down menu, the interview outline, the dialog pane, and the resource pane	F6 / Shift+F6

Move between HotDocs windows	Alt+F6
Resize the pane you are currently viewing This option selects the split bar so you can use the arrow keys to move it. Press <b>Enter</b> to set the placement.	Ctrl+F6
Check the spelling of answers in the interview	F7
Access the assembly window menu bar	F10
Display a shortcut menu	Shift+F10
Send the assembled document to the word processor	F11
Copy the assembled document to the currently open document	Shift+F11
Copy the assembled document to the Clipboard	Ctrl+F11
Save the assembled document to disk	F12
Send the assembled document to the word processor and close HotDocs	Ctrl+F12

## Assembly Window: Document Preview Tab

To Do This	Press
Tab through the answer fields in a dialog	Tab / Shift+Tab
Switch between the tabs of the assembly window	Ctrl+Tab
Open a pop-up interview for the current answer	Enter (make sure your cursor is in an answer

field (Word users)	field)
Open the <b>Document Text Editor</b> for the section of editable text (Word users)	Enter (make sure your cursor isn't in an answer field)
Move through each unanswered question in the document (Word users)	Ctrl+Down Arrow / Ctrl+Up Arrow
Move through each section of editable text in the document (Word users)	Ctrl+Shift+Down Arrow / Ctrl+Shift+Up Arrow
Select all of the text in the document	Ctrl+A
Copy the selected text to the Clipboard	Ctrl+C
Search through the document text for a specific word or phrase	Ctrl+F (Press F3 to find the next instance)
Move your cursor to a specific line number or character position in the document	Ctrl+G
Use a new, untitled answer file	Ctrl+N
Open Answer File Manager so you can choose a different answer file for the assembly	Ctrl+O
Print a copy of the assembled document	Ctrl+P
Save the answer file	Ctrl+S
Paste the text on the Clipboard in the document	Ctrl+V
Cut the selected text and place a copy of it on the Clipboard	Ctrl+X
Close the assembly window	Ctrl+W

Open the HotDocs Help window	F1
Display the What's This Help cursor	Shift+F1
Display the What's This Help text	Ctrl+F1
Close the assembly window	Alt+F4
Zoom in or out on the document text	Ctrl+= / Ctrl+-
Move between HotDocs windows	Alt+F6
Access the assembly window menu bar	F10
Display a shortcut menu	Shift+F10
Send the assembled document to a new word processor document	F11
Send the assembled document to the currently open word processor document	Shift+F11
Copy the assembled document to the Clipboard	Ctrl+F11
Save the assembled document to disk	F12

## Assembly Window: Form Document Tab

To Do This	Press
Move between the different answer fields of the form document	Tab /Shift+Tab

View a pop-up interview for the selected field	Enter
View the <b>Field Properties</b> dialog box for the current field	Alt+Enter
Display the full list of options for a drop-down field	Alt+Down Arrow
Move between unanswered fields on the form	Ctrl+Down Arrow / Ctrl+Up Arrow
Select all of the text in a field, or, if your cursor isn't on a field, select all of the fields on the form page	Ctrl+A
Copy the selected text or fields to the Clipboard	Ctrl+C
Change the font properties of the selected field(s)	Ctrl+D
Go to a specific page in the form	Ctrl+G
Move between the fields in the document where the same answer is merged	Ctrl+M / Ctrl+Shift+M
Create a new answer file for use during the current interview	Ctrl+N
Open Answer File Manager so you can choose a different answer file for the current interview	Ctrl+O
Print a copy of the form document	Ctrl+P
Save the current answer file	Ctrl+S
Paste the contents of the Clipboard	Ctrl+V
Close the assembly window	Ctrl+W

Cut the selected text or field to the Clipboard	Ctrl+X
Zoom in or out on the form	Ctrl+= / Ctrl+-
Open the HotDocs Help window	F1
View the What's This Help cursor	Shift+F1
View the What's This Help text	Ctrl+F1
View the answer's resource	Alt+F1
Switch between the document pane and the resource pane	F6
Move between HotDocs windows	Alt+F6
Resize the pane you are currently viewing This option selects the split bar so you can use the arrow keys to move it. Press <b>Enter</b> to set the placement.	Ctrl+F6
Check the spelling of answers on the form	F7
Display the page width of the form	F9
Display the full page of the form	Shift+F9
Access the assembly window menu bar	F10
Display a shortcut menu	Shift+F10
Send the assembled document to HotDocs Filler	F11

Save a copy of the assembled document to disk	F12
---	-----

## Assembly Window: Comparison Tab

The shortcut commands for the **Comparison** tab are identical to those for the **Document Preview** tab, with the following changes:

To Do This	Press
Move to the next difference in the document	Tab
Move to the previous difference in the document	Shift+Tab
Send the comparison document to the word processor	F11
Save the comparison document to disk	F12

## Assembly Window: Clause Library

To Do This	Press
Move a clause from the <b>Available</b> list to the <b>Selected</b> list	Insert
Remove a clause from the <b>Selected</b> list	Delete
Move to the top of the clause list	Home
Move to the bottom of the clause list	End
Move the selected clause up in the <b>Selected</b> list	Alt+Up Arrow
Move the selected clause down in the <b>Selected</b>	Alt+Down Arrow

list	
------	--

## Answer File Manager

To Do This	Press
Tab between the item list and the Find options	Tab / Shift+Tab
Switch between <b>Properties</b> and <b>Contents</b> tabs	Ctrl+Tab
Open the <b>Item Properties</b> dialog box for the currently selected answer file	Alt+Enter
Remove the answer file from the library	Delete
Move the selected item up or down in the list	Alt+Up Arrow / Alt+Down Arrow
Expand or collapse a folder	Right Arrow / Left Arrow
Copy the selected answer file	Ctrl+C
Create a new answer file	Ctrl+N
Cut the selected item or text	Ctrl+X
Open the HotDocs Help file	F1
Display the What's This Help cursor	Shift+F1
Display the What's This Help text	Ctrl+F1
Close Answer File Manager	Alt+F4

Switch between the answer file list and the <b>Preview</b> pane	F6
Move between HotDocs windows	Alt+F6
Resize the pane you are currently viewing This option selects the split bar so you can use the arrow keys to move it. Press <b>Enter</b> to set the placement.	Ctrl+F6
Access the Answer File Manager toolbar	F10
View the shortcut menu	Shift+F10

## General Commands

To Do This	Press
Display all of the options in a drop-down list	Alt+Down Arrow
Move items up or down within a list (for example, the <b>Contents</b> list of the <b>Dialog Editor</b> )	Alt+Down Arrow / Alt+Up Arrow
Edit the selected component	Ctrl+E
Create a new component	Ctrl+N
Close the open component editor and cancel your changes	Ctrl+W
Move between HotDocs windows	Alt+F6
View a shortcut menu	Shift+F10
Move between the different tabs of a window	Ctrl+Tab

Access the buttons in a toolbar (when no menu is provided)

F10

# Using HotDocs with Microsoft Word 2007

If you are using HotDocs with Microsoft Word 2007, you should be aware of a few differences between earlier versions of Word and Word 2007:

- **Location of HotDocs toolbar:** To access the HotDocs editing and navigation toolbar buttons, you must click the **HotDocs** tab of the Word ribbon. This displays each of the editing and navigation commands.
- **Accessing the HotDocs button:** The HotDocs button is located on the **HotDocs** tab of the Word ribbon.
- **File formats:** Word 2007 includes two new XML-based template formats—.DOTX (indicates the file does not use macros) and .DOTM (indicates the file does use macros). ***These template formats are not supported in HotDocs.*** However, if you are using Word 2007 to save assembled documents, you can save in these formats: .DOCX and .DOCM.
- **Compatibility Mode:** When you edit existing templates in Word 2007, the template opens in Compatibility Mode. According to the Microsoft Word 2007 Help file, "Compatibility Mode ensures that no new or enhanced features in Office Word 2007 are available while you are working with a document, so that people who are using previous versions of Word will have full editing capabilities."

If you plan to use your templates with just Word 2007, you can convert your templates to Word 2007 format using the Word conversion tool. (See the Word Help file for additional information on doing this.) If you plan to distribute or share your templates with users who are using earlier versions of Word, however, you should either continue to work in Compatibility Mode, or you should take care not to use features specific to Word 2007. (The Word Help file has a full list of Word 2007 elements.)

- **Styles:** When you create a new, empty Word template, HotDocs applies whatever styles are in *HotDocs6.dotx* 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.

## Complete the HotDocs Tutorials

When you purchased your license for HotDocs, you should have received a copy of *HotDocs Installation Guide and Tutorial* in PDF format on the installation CD. In addition to providing instructions for installing HotDocs, this manual also contains several lessons that teach the basics of template automation. To complete these lessons, you should refer to this manual. (You can also find a copy of the *HotDocs Installation Guide and Tutorial* at <http://www.hotdocs.com/documentation/guides/>.)

Once you complete the lessons in the *HotDocs Installation Guide and Tutorial*, you may download additional tutorials from the HotDocs Web site. (See <http://www.hotdocs.com/documentation/tutorials/>.)

# Documentation Feedback

To improve the quality of the tutorials and the help files, 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 which version of the product you are using, as well as whether your suggestion is in regards to the *Installation Guide & Tutorial* or the electronic help file. If commenting about a specific topic, include that information as well.

**Warning:** The HotDocs Publications team cannot respond to technical support or project consulting questions. We are mainly interested in problems with the documentation itself—such as erroneous information, grammatical and spelling errors, or suggestions for topics to include in the next release of the software.

E-mail your comments and suggestions to [publications@hotdocs.com](mailto:publications@hotdocs.com).

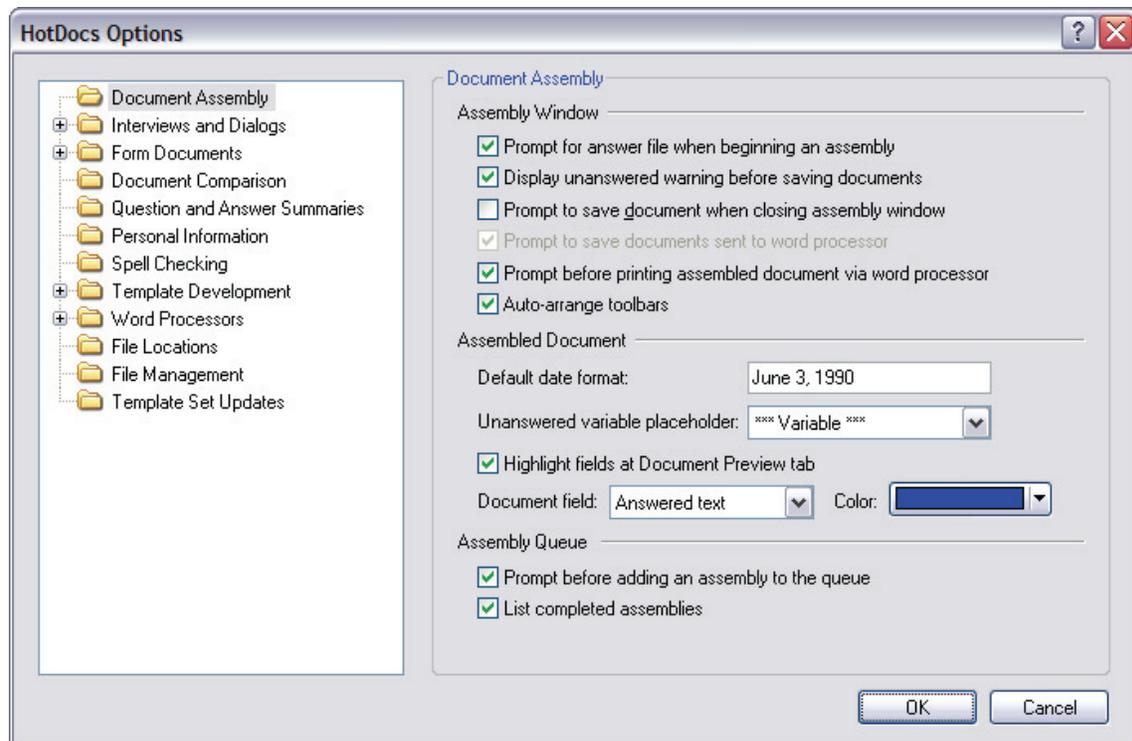
# Setting HotDocs Options

## Overview: Customize HotDocs

You can change the way HotDocs looks and works by changing your preferences at **HotDocs Options**. 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.

The following example shows the **HotDocs Options** window. You can select a specific folder and then specify options that control the way HotDocs works.



**Note:** 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.

# Display Repeated Dialogs Using a Special Icon

When HotDocs starts an assembly, it displays an interview outline as well as a dialog pane. The interview outline contains a list of all the dialogs in a given template. These 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 (  ). 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 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 **Interview Outline** group, select **Display repeat icons in interview outline**.

**Note:** 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. Some dialogs contain inserted dialogs (or child dialogs) that are also represented visually in the interview outline. 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. 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 **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.

**Note:** 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 in 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 assembly

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.

# Control the Functionality of the Enter Key During Assembly

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 assembly

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.

**Note:** 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 button 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 Variable 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 re-display 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.

**Note:** 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.

### Notes:

- 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- 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** box (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**.

# 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 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** box.
  - 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** box.
  - 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** box. (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**.

## To restore all default properties

- At the **Appearance** window, click **Restore Default Fonts and Colors**.

**Note:** 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.

# 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.

**Warning:** 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. Likewise,  **Save the assembled document as a PDF** will only be available if PDF Advantage is installed.

**Note:** 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  **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.

# 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

6. At the HotDocs library, click the  **Options** button. The **HotDocs Options** dialog box appears.
7. Click the **Document Assembly** folder. The window changes to show several assembly options.
8. 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**.

**Note:** 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.

**Note:** 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**.

**Note:** 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** box. (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.

### Notes:

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.

**Warning:** 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.

**Note:** 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.

## 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**.

**Note:** 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.

# Specify How Documents Should Be Marked Up

**Warning:** Document Markup is available to Microsoft Word users only.

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. (See [View the Template in Markup View](#) for details.)

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. See [View an Assembled Document in Markup View](#) for details.

## To specify how documents should be marked up

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:

To	Do This
Choose the character or characters that should be used to mark fields in the template	Enter the opening and closing characters in the boxes next to the <code>field type</code> . <ul style="list-style-type: none"> <li>■ <b>Answers field</b> = variable fields</li> <li>■ <b>Insertion field</b> = INSERT instructions</li> <li>■ <b>Editable text</b> = SPAN / END SPAN instructions</li> <li>■ <b>Conditional text</b> = IF/ END IF instructions</li> <li>■ <b>Alternative text</b> = ELSE IF / ELSE instructions</li> <li>■ <b>Repeated text</b> = REPEAT / END REPEAT instructions</li> </ul>
Mark variable fields with any comments entered for the variable	Select <b>Use developer comments to label answer fields</b> . If you have entered comments for a variable field, they will be merged in the markup field.
Use a Multiple Choice variable's merge text as the label for its field	Select <b>Use option text to label multiple choice answer fields</b> .

<p>If you've named your merge text, the name will be used instead.</p>	
<p>Mark text blocks (such as IF, REPEAT, and SPAN instructions) using developer comments entered for the instruction</p>	<p>Select <b>Use developer comments to label beginning of text blocks</b>.</p> <p>To merge the same comment in the closing markup field, select <b>and end of text blocks</b>.</p>
<p>Include both opening and closing markers when merging a field label (which includes developer comments)</p> <p>By default, when HotDocs marks text blocks, an opening field marker is merged before the opening instruction and the closing marker is merged after the closing instruction.</p>	<p>Select <b>Enclose text block labels in markers</b>.</p> <p>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.</p> <p>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:</p> <p><i>[*Insert paragraph if employee must complete a trial period*]</i></p> <p><i>The length of [Employee Name]'s employment will be an initial term of six months. *]</i></p>
<p>Include text block instructions in either a footnote or endnote</p>	<p>Select <b>Add Footnote/Endnote at beginning of text blocks</b>. Then select whether the annotation will be marked using <b>Letters</b> or <b>Numbers</b>.</p> <p>To annotate the closing field label, select <b>and end of text blocks</b>.</p>
<p>Apply font attributes, such as bold and italics, to a markup field</p>	<p>Select <b>Bold</b> or <b>Italic</b>.</p>
<p>Define the color scheme used for marking the different fields in the template</p>	<p>Select <b>Color</b> and click the <b>Color</b> drop-down button. Then choose the color scheme. (See <a href="#">Define Colors for Text Template Fields and Instructions</a> for details.)</p>
<p>Include SPAN instructions when showing a template in Markup View</p> <p><b>Warning:</b> By default, HotDocs does not show SPAN instructions when you switch to</p>	<p>Select <b>Mark editable text blocks</b>.</p>

Markup View.	
Show only basic markup when previewing the template at the HotDocs library	<p>Select <b>Use minimal markup at the HotDocs library Preview tab</b>.</p> <p>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.)</p> <p>To assign font formatting and color to markup fields at the <b>Preview</b> tab, select <b>Apply attributes to answer fields when minimal markup is used</b>.</p>

# Define Colors for Text Template Fields and Instructions

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.

### Notes:

- 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 Used in a HotDocs Automator](#).

# Change the Units of Measurement

HotDocs Automator measures the fields you create to identify whether they should be treated as check-box 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 Automator

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.

**Note:** To change the colors used for text template fields, see [Define Colors for Fields and Instructions](#).

# 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 template, as described in the following table:

To	Do This
Display the colored fields	Select <b>Highlight Fields</b> .
Display the names of variables attached to each field	Select <b>Show Variables</b> .
Display thumbnail images of each page of the form	Select <b>Show Thumbnails</b> . The thumbnails appear in a separate pane in the window. You can click on the thumbnail images to move through the form.
Set the level of magnification	At the <b>Zoom</b> drop-down list, select one of the following options: <b>100%</b> (displays templates at actual size), <b>Fit to Width</b> (matches template width to the left and right edges of the window), or <b>Fit to Height</b> (matches template height to the top and bottom edges of the window).

### Notes:

- You can hide the fields, variables, and thumbnails by clearing these options.
- To change the colors used for the fields, see [Change Colors Used in a HotDocs Form](#).

# 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 Automator 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 Options** 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.

**Note:** 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. Optionally, print a form document, selecting the printer you want to associate with forms.

**Note:** 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 have HotDocs always print the assembled document using HotDocs' printing technology.
  - Select **Adobe Acrobat** to have HotDocs always print the document using Adobe's printing technology.

# 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 first create a new form template. 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.

The backup file is 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.

- 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.)

# 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 template or document. 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:

To	Do This
Set font properties for text in edit fields	Click the <b>Font</b> button, then make changes at the <b>Font</b> dialog box.
Set the horizontal alignment for text in edit fields	At the <b>Horizontal alignment</b> drop-down list, select <b>Left</b> (text is placed next to the left edge of the field), <b>Right</b> (text is placed next to the right edge of the field), <b>Center</b> (text is placed in the middle of the field), or <b>Justified</b> (text is spaced out to reach from the left edge to the right edge of the field).
Set the vertical alignment for text in edit fields	At the <b>Vertical alignment</b> drop-down list, select <b>Top</b> (text is placed next to the upper edge of the field), <b>Center</b> (text is placed halfway between the upper and lower edge of the field), <b>Bottom</b> (text is placed next to the bottom edge of the field), or <b>Justified</b> (text is spaced out with the first line next to the top edge, the last line next to the bottom edge, and any other lines spaced evenly across the middle).  Vertical alignment is only apparent in multiple-line fields.
Align the text in multi-line fields with the top border of the field	Select <b>Set to Top for multi-line fields</b> .
Change the distance between lines of text in a multiple-line field	At the <b>Lines per inch</b> box, 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 <b>Max lines 1</b> for single-line fields.
Change the distance between the text and the edges of the field	At the <b>Margins</b> group, type a number (or click the up or down arrows to select a number) for the left, right, top, and bottom margins.

**Warning:** 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.

**Note:** 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:

To	Do This
Set font properties for text in check-box fields	Click the <b>Font</b> button, then make changes at the <b>Font</b> dialog box.
Set the minimum size for edit fields so that all fields created smaller than this size will be created as check-box fields	Enter a number in the <b>Check-box detection threshold</b> box.  If you specify <i>0.00</i> 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 <b>Character</b> box.
Specify the smallest size that will be used for the check-box character	Enter a font size in the <b>Minimum size</b> box.
Set the horizontal alignment for text in check-box fields	At the <b>Horizontal alignment</b> drop-down list, select <b>Left</b> (text is placed next to the left edge of the field), <b>Right</b> (text is placed next to the right edge of the field), <b>Center</b> (text is placed in the middle of the field), or <b>Justified</b> (text is spaced out to reach from the left edge to the right edge of the field).
Set the vertical alignment for text in check-box fields	At the <b>Vertical alignment</b> drop-down list, select <b>Top</b> (text is placed next to the upper edge of the field), <b>Center</b> (text is placed halfway between the upper and lower edge of the field), <b>Bottom</b> (text is placed next to the

	<p>bottom edge of the field), or <b>Justified</b> (text is spaced out with the first line next to the top edge, the last line next to the bottom edge, and any other lines spaced evenly across the middle).</p> <p>Vertical alignment is only apparent in multiple-line fields.</p>
<p>Change the distance between the check-box character and the edges of the field</p>	<p>At the <b>Margins</b> group, type a number (or click the up or down arrows to select a number) for the left, right, top, and bottom margins.</p>

**Warning:** 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.

# 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** box. 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.

**Warning:** The changes you make are not applied to existing fields. The changes won't take effect until you create a new field.

### Notes:

- 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 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** box, 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** box, enter the text you want to appear as the label for the entry in the addendum.
6. At the **Table answer column indent** box, 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.

**Warning:** The changes you make are not applied to existing fields. The changes won't take effect until you create a new field.

**Note:** If you leave the **Cross-reference text** box empty, HotDocs uses a default reference: "*See # in Addendum.*"

# 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 create a default interview 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**.

**Warning:** The changes you make are not applied to existing templates. The changes won't take effect until you create a new template.

### Notes:

- You can generate default interviews on a template-by-template basis (see [Have HotDocs Create 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.

# Select Paper Size and Paper Source 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.

**Warning:** The changes you make are not applied to existing templates. The changes won't take effect until you create a new template.

**Note:** You can use command-line options to specify these printing properties. See the [Paper Tray](#) and [Paper Size](#) command-line options.

# Customize 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** box, type the text you want displayed in the addendum header or footer. (See [Use Answer Overflow and Addendum Text Codes](#).)
  - At the **Height** box, 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** (text is placed next to the left edge of the header), **Center** (text is placed in the middle of the header), **Right** (text is placed next to the right edge of the header), or **Justified** (text is spaced out to reach from the left edge to the right edge of the header).
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** box.
  - To control the distance between answers in the addendum, enter a number (or click the up or down arrows to select a number) in the **Spacing** box.
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.

**Warning:** The changes you make are not applied to existing templates. The changes won't take effect until you create a new template.

# Set Preferences for Saving Comparison Snapshots

**Note:** HotDocs Compare is available for Word users only.

Using HotDocs Compare, you can compare different versions of the document you are assembling. For example, if you want to change a single answer in a document and see how that change affects the content of the document, you can answer questions in the interview a certain way, save a snapshot of the document, change the answers, and then compare the current document against the snapshot. Differences in the document are marked using strike-through and underlining formats.

By default, HotDocs will always use a single snapshot. This means that each time you click the **Save Document Snapshot** button, the contents of the existing snapshot (if any) will be overwritten with the new snapshot. However, you can specify an option that lets you save multiple snapshots and have them available each time you want to compare documents. You can specify the default file location for saved snapshots.

## To control how document snapshots are saved

1. At the HotDocs library, click the  **Options** button. The **HotDocs Options** dialog box appears.
2. Click the **Document Comparison** folder. The window changes to show comparison options.
3. Select **Allow saving and comparing of multiple snapshots** to have HotDocs give you the option of having access to the different snapshots you've saved.
4. Click the  **Browse** button next to the **Default folder for saved snapshot files** box and specify the file location where HotDocs should save snapshot documents.

**Note:** Information in this topic applies to **HotDocs Compare** users only. For details on purchasing a license for HotDocs Compare, contact your HotDocs sales representative at (800) 500-3627.

# Save Edited Document Text in Comparison Snapshots

**Note:** HotDocs Compare is available for Word users only.

Some template developers design their templates so that you are able to edit the text of the document at the **Document Preview** tab of the assembly window. (See [Edit Document Text at the Document Preview Tab](#).) You can designate whether these changes are automatically represented in the **Saved snapshot** pane of the **Comparison** tab or whether HotDocs prompts you to include the edited text each time you create a snapshot.

## To always save edited document text in a saved snapshot

1. At the HotDocs library, click the  **Options** button. The **HotDocs Options** dialog box appears.
2. Click the **Document Comparison** folder. The window changes to show comparison options.
3. To have HotDocs ask each time you create a snapshot whether you want to include edited document text, select **Prompt when saving snapshot containing edited text**. (To always save edited text in the snapshot, clear this option.)

**Note:** Information in this topic applies to **HotDocs Compare** users only. For details on purchasing a license for HotDocs Compare, contact your HotDocs sales representative at (800) 500-3627.

# Specify Colors for Marking Changes in Comparisons

**Note:** HotDocs Compare is only available for Word users.

You can designate which colors HotDocs uses when marking changes in documents that are being compared. Additionally, you can designate that HotDocs simultaneously highlights both the insertion and deletion in both panes of the comparison as you tab through the changes in the document.

## To specify the colors used in document comparison

1. At the HotDocs library, click the  **Options** button. The **HotDocs Options** dialog box appears.
2. Click the **Document Comparison** folder. The window changes to show comparison options.
3. Optionally, select **Highlight paired insertion and deletion when moving to next/previous field**. When this option is selected, both the change in the snapshot and the change in the current document will be highlighted as you view each difference in the document.
4. Click the **Differences** drop-down button and choose an option: **Deleted text** (represents the original text from the document snapshot) or **Inserted text** (represents the changed text in the current document).
5. Click the **Color** drop-down button and choose a color.

### Notes:

To restore the default colors, click the **Color** drop-down button and choose **Default** from the top of the **Palette** tab.

Information in this topic applies to **HotDocs Compare** users only. For details on purchasing a license for HotDocs Compare, contact your HotDocs sales representative at (800) 500-3627.

# 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 make a question or answer summary appear in one column

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. At the **Format** drop-down list, select **One-Column**. Several one-column format options become available:
  - 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. Optionally, change the number in the **Maximum multiple-choice options to display** box to control the number of multiple-choice options that appear in a question or answer summary.

## To make a question or answer summary appear in two columns

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 several summary options.
3. At the **Format** drop-down list, select **Two-Column**. Several two-column format options become available:
  - 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** box. 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.
4. Optionally, change the number in the **Maximum multiple-choice options to display** box 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.

**Note:** To once again generate a question summary that shows *all* the document questions, clear **Honor IF instructions when answered**.

# Enter or Edit Your Personal Information

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.

There are two types of Personal Information variables—*built-in* and *custom*.

- **Built-in Personal Information variables** are those that are automatically available in every HotDocs template. If there are multiple users using one workstation, each Personal Information value is stored in the user's own specific section of the registry.
- **Custom Personal Information variables** are those that are created and used by individual users.

## Edit Answers for Built-In Personal Information Variables

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 your personal information

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** box and click **OK**. The value is now listed in the **Answer** column.

### To remove an answer for a built-in 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, select the variable that contains the answer you want removed.
4. Click **Delete**. HotDocs confirms the removal of the answer.
5. Click **Yes**. HotDocs removes the answer, but leaves the variable in the list for others to use.

## Create Custom Personal Information Variables

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 edit answers for custom Personal Information variables, refer to the first section in this topic.

### 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** box.
5. If you know the value you want assigned to the variable, type it in the **Answer** box and click **OK**.

#### **To remove a custom 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, select the variable you want removed.
4. Click **Delete**. HotDocs confirms the removal.
5. Click **Yes**. The variable is deleted.

# 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:

To	Do This
Have HotDocs ignore capitalized words, such as names, cities, titles, and so on	Select <b>Ignore capitalized words (e.g. Canada)</b> .
Have HotDocs ignore words that contain all capital letters, such as acronyms or abbreviations	Select <b>Ignore words in all capitals (e.g. ASAP)</b> .
Have HotDocs ignore words that contain both uppercase and lowercase letters	Select <b>Ignore words with mixed case (e.g. InLine)</b> .
Have HotDocs ignore words that also contain digits	Select <b>Ignore words with digits (e.g. Win95)</b> .
Have the replacement words HotDocs suggests match the case of the misspelled word.	Select <b>Match case when comparing</b> .
Have HotDocs match the case of misspelled words when replacing them.	Select <b>Match case when replacing</b> .
Have HotDocs provide a list of possible replacements for misspelled words	Select <b>Always suggest replacements</b> .
Have HotDocs provide a list of possible replacements based on words that sound like the misspelled word	Select <b>Make phonetic suggestions</b> .

Have HotDocs provide a list of possible replacements based on words that are typographically similar	Select <b>Make typographical suggestions</b> .
Have HotDocs provide a list of possible replacements that are made up of more than one word	Select <b>Make split word suggestions</b> .
Have HotDocs check the spelling only if the text is more than 1 line	Select <b>Check multiple-line text only</b> .
Have HotDocs use the British spelling dictionary rather than the English dictionary	Select <b>Use UK spelling dictionary</b> .

When a word is not in the spelling checker's dictionary, HotDocs checks for it in your personal dictionary, *IGNORE.TLX*. If it is not found there, you can add it by clicking **Add** at the **Check Spelling** dialog box.

**Note:** You can change these settings whenever you are spell checking. To do this, click **Options** at the **Check Spelling** dialog box.

# 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 Always 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 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 Whether 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.

## Example

<p><i>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.</i></p> <p>SHIPPED TO:</p> <p>«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»</p>	<p><i>In this example, returns are not included after each instruction. This makes it more difficult to see the conditions in relation to the text.</i></p> <p>SHIPPED TO:</p> <p>«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»</p>
---	--

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.

The following table demonstrates the differences in such a case:

<p><i>In this example, the returns after the instructions force text within a paragraph to a new line. This may be confusing to understand.</i></p> <p>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»</p>	<p><i>In this example, the returns after the instructions are omitted, allowing all of the text to remain in the same paragraph.</i></p> <p>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».</p>
--	--

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 Returns Are Inserted After Instructions](#) for a complete description.)
  - Choose **Never** to have HotDocs never insert a return character after an instruction.

**Warning: 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

**Note:** 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, either 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, either 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](#)).

**Note:** 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 latest version of HotDocs, select **Warn when automatically converting templates**. (See [Compatibility of HotDocs 2008 Files with Earlier Versions of HotDocs](#) and [Convert a 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.

# 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:

To	Do This
Have HotDocs start the next line of a script (after you enter a return) at the same level of indentation	Select <b>Automatic indentation</b> .
Include keywords for months, language codes, and so forth in the auto-complete list	Select <b>Include seldom-used keywords in auto-complete list</b> . (If you'd rather exclude these keywords from the list, clear this option.)
Allow the <b>Tab</b> key to insert a tab in a script, rather than move you to the next field in the component editor	Select <b>Tab key inserts a tab in scripts</b> . Now, whenever you press <b>Tab</b> or <b>Shift+Tab</b> , HotDocs will insert a tab in the script.  <b>Warning:</b> If you select this option, you can no longer use keyboard commands to move from the <b>Script</b> box to another field in the component editor—you must use your mouse to click out of the <b>Script</b> box.
Specify the number of characters HotDocs should use when you insert a tab in the script	Enter a number in the <b>Tab width</b> box.
Have HotDocs use colors to differentiate between instruction and expression keywords, operators, and component names	Select <b>Syntax coloring</b> . (You can then either accept the default colors HotDocs assigns to these, or you can designate your own colors at the <b>Colors and Attributes</b> list.)
Assign colors and other formatting attributes to the various parts of a script	Select the item, click the <b>Color</b> drop-down button, and choose a color. You can also select <b>Bold</b> to make the item bolded, and you

<p>These "parts" include instruction keywords, expression keywords, components (such as variables and dialogs), text values (literal text strings), other values (such as literal number and date values and language codes), placeholders, operators, commented scripts, and any syntax HotDocs doesn't recognize.</p>	<p>can select <b>Italic</b> to italicize the item.</p> <p>To restore the HotDocs defaults, select the script element, click the <b>Color</b> drop-down button, and choose <b>Default</b> from the top of the <b>Palette</b> tab.</p>
<p>Specify the font used for the script</p>	<p>Click <b>Select Font</b> and make your selection. You can also specify the size of the script text.</p>

# 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:

To	Do This
Have Template Manager expand the list of components to include example formats, patterns, and so forth	Select <b>Include supplemental components in component list</b> .
Include a column in the component list that shows component types	Select <b>Show component type column</b> .
Include a column in the component list that shows the status of the components. (The status tells you whether the component is <b>OK</b> (used in both the template and component file), <b>unused</b> (the component is not referenced in any templates), or <b>missing</b> (the component is referenced in a template, but not found in the component file).)	Select <b>Show component status column</b> .
Have Template Manager overlay a graphic reminder of what the component's status is on the component's icon. (The status tells you whether the component is <b>OK</b> (no reminder is used), <b>unused</b>  , or <b>missing</b>  .)	Select <b>Overlay status hint on component icons</b> .

**Note:** You can also access these options by clicking the  **Options** button in the Template Manager toolbar.

# Install Support for New Word Processors

**Warning:** 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 and Exit 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](#).)

## Notes:

- 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.)

# 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).

**Warning:** You cannot change the location of the startup/macro files.

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 the Local Disk or Network](#).

# 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 (ANS and 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.

**Warning:** 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 **^keywordGeneral\template.ext** for those templates in the *General* subfolder and **^keywordFamily 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

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** box.
5. Click the  **Browse** button next to the **Path** box, 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.)

## To assign a reference path to a specific template or library item

1. At the HotDocs library window, select the template for which you want to specify a reference path.
2. Click the  **Properties** button. The **Item Properties** dialog box appears.
3. If the **File name** box 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.
4. Place your cursor before the first letter in the **File name** box and type caret (^), the keyword, and a backslash (\).
5. Click **OK**.

**Notes:**

- 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](#).

# Manage Answer Files

To work with answer files, you can either use Answer File Manager, or you can use Windows Explorer. (See [Overview: 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. The **HotDocs Options** dialog box appears.
2. Click the **File Management** folder. The window changes to show file management options.
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.

**Warning:** 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.
- **Document Manager** uses a third-party document management program to manage the use of answer files. (To use this option, make sure you specify the ODMA DLL file name and path in the **Document manager ODMA DLL location** box.)

You can also specify an option that saves your answer files to a document management program. When you do this, you can specify in which file format the answer file should be saved.

## To specify the format document managers should use to save answer files

1. Complete the steps above, choosing **Document Manager**.
2. At the **Answer file save format** drop-down list, select one of the following options:
  - **Ask Before Saving** causes HotDocs to prompt for the file format each time an answer file is saved.
  - **HotDocs Answer File (.ans)** automatically saves the file in HotDocs native answer file format.
  - **HotDocs XML Answer File (.anx)** automatically saves the file in XML format. (See [Create an XML Answer File](#).)

# Manage Assembled Documents

You can use a document management program to control how assembled documents are saved. You can also specify the format HotDocs uses when it saves the document to the document manager.

## To manage assembled documents

1. At the HotDocs library, click the  **Options** button. The **HotDocs Options** dialog box appears.
2. Click the **File Management** folder. The window changes to show file management options.
3. Click the **Manage assembled documents using** drop-down button and select an option:
  - **Windows Explorer** uses Windows Explorer to save and access assembled document files.
  - **Document Manager** uses a third-party document management program to manage assembled document files. (To use this option, make sure you specify the ODMA DLL file name and path in the **Document manager ODMA DLL location** box.)

If you select **Document Manager**, you can specify the format of the assembled document by clicking the **Save format** drop-down button next to the types of documents you are saving and making a selection.

# 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. Select any of the following options:

To	Do This
Always check for updates when you start HotDocs	Select <b>Perform scheduled checks for updates when HotDocs is started</b> . 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 <b>Show progress window when checking for updates</b> .
Choose which template sets you want to check for updates	Select the set in the <b>Select the template sets you want to keep up-to-date</b> list.

# Control How PDF Documents Are Saved

The HotDocs PDF driver allows you to automatically name PDF documents, or save them to a specific location. Using these properties, you can ensure that new files don't overwrite existing files. Or, if your project requires it, you can ensure that new files automatically overwrite the outdated versions.

In addition, you can specify that new PDF documents are automatically sent to a specific e-mail address. Settings you choose are used each time you create a PDF document.

## To specify how new PDF files are stored

1. At any Windows program with printing capability, select the **Print** command (usually at the **File** menu).
2. At the printing dialog box, select the **HotDocs PDF Driver** from the list of available printers.
3. Click the **Properties** button.
4. Click the **PDF Settings** tab.
5. Select any options you need. (Click the **Help** button in the **HotDocs PDF Driver Properties** title bar for more information about a specific option.)

**Note:** Information in this topic applies to **HotDocs PDF Advantage** users only. For details on purchasing a license for PDF Advantage, contact your HotDocs sales representative at (800) 500-3627.

# Set Properties for the HotDocs PDF Driver

The HotDocs PDF Driver provides many controls for modifying the PDF files you create. When you change settings, the new values are saved and applied the next time you print a PDF document. (The functionality of PDF Driver properties may be limited by the program you use to print PDF files. When you work with PDF driver properties, use an application with full functionality.)

## To change the properties of a PDF document

1. Open a document in a Windows program with printing capability.
2. Select the **Print** command for that program.
3. At the printing dialog box, select **HotDocs PDF Driver** as the printer.
4. Click the **Properties** button. The **HotDocs PDF Driver Properties** dialog box appears.
5. Adjust the appearance and functionality of the PDF document you are creating, using the options described in this table:

To	Do This
Change the page size or orientation	At the <b>General</b> tab, change the <b>Paper Size</b> and <b>Orientation</b> settings.
Increase the quality of the PDF image	At the <b>General</b> tab, select a higher <b>Resolution</b> setting.  Higher resolution helps produce more accurate appearance, such as controlling exact character spacing and line breaks. However, higher resolution may also increase the file size. Lower resolutions work well for on-screen viewing, while higher resolutions (600 and greater) may be needed for printing to paper.
Enlarge or reduce the size of the text	At the <b>General</b> tab, change the <b>Scale</b> setting (from 25% reduction to 400% magnification).
Optimize for quicker viewing on the Internet	At the <b>PDF Settings</b> tab, select <b>Optimize for Web Viewing</b> .
Minimize file size by specifying which fonts are embedded	At the <b>PDF Settings</b> tab, click the <b>Font Embed</b> drop-down button and choose <b>Embed Non-Standard Fonts</b> , <b>Embed All Fonts</b> , or <b>Do Not Embed Fonts</b> .  Embedding fonts helps ensure accurate display of your document. If people accessing the document don't have the same fonts installed on their computers, a different font

	may be substituted. However, if you embed the font, the document will appear with the same fonts you used. The drawback is that embedding fonts often increases the file size.
Minimize file size by compressing all text and graphics	At the <b>PDF Settings</b> tab, click the <b>Compression</b> drop-down button and select a compression option.
Insert a watermark	At the <b>PDF Settings</b> tab, select a watermark. (See <a href="#">Create and Edit PDF Watermarks</a> .)

**Note:** Information in this topic applies to **HotDocs PDF Advantage** users only. For details on purchasing a license for PDF Advantage, contact your HotDocs sales representative at (800) 500-3627.

# Create and Edit PDF Watermarks

You can create your own watermarks to display on your PDF documents, or modify existing watermarks.

## To create or edit a watermark

1. Open the document in a Windows program with printing capability.
2. Select the **Print** command for that program.
3. At the printing dialog box, select **HotDocs PDF Driver** as the printer.
4. Click **Properties**. The **HotDocs PDF Driver Properties** dialog box appears.
5. Click the **PDF Settings** tab. The view changes to show properties for the PDF document.
6. In the **Watermark** group, click the **Watermark Name** drop-down button and select the type of watermark you want to use. (To change the properties of the watermark, including its placement on the page, click **Advanced** and make your changes there.)
7. In the **Opacity** box, enter a number to indicate the transparency you want to use for the image. (The smaller the number, the less visible the image will be.)
8. Indicate whether the watermark should appear on the **First Page Only** as well as if it should appear **As Background**.
9. Click **OK**. You are returned to the **Print** dialog box where you can specify your other printing options.

### Notes:

- If you want to use a combination of text and graphics for your watermark, click the **Advanced** button and choose **Composite** from the **Watermark Type** drop-down list. Then choose your options from the list of options in the **Composite Properties** group.
- Information in this topic applies to **HotDocs PDF Advantage** users only. For details on purchasing a license for PDF Advantage, contact your HotDocs sales representative at (800) 500-3627.

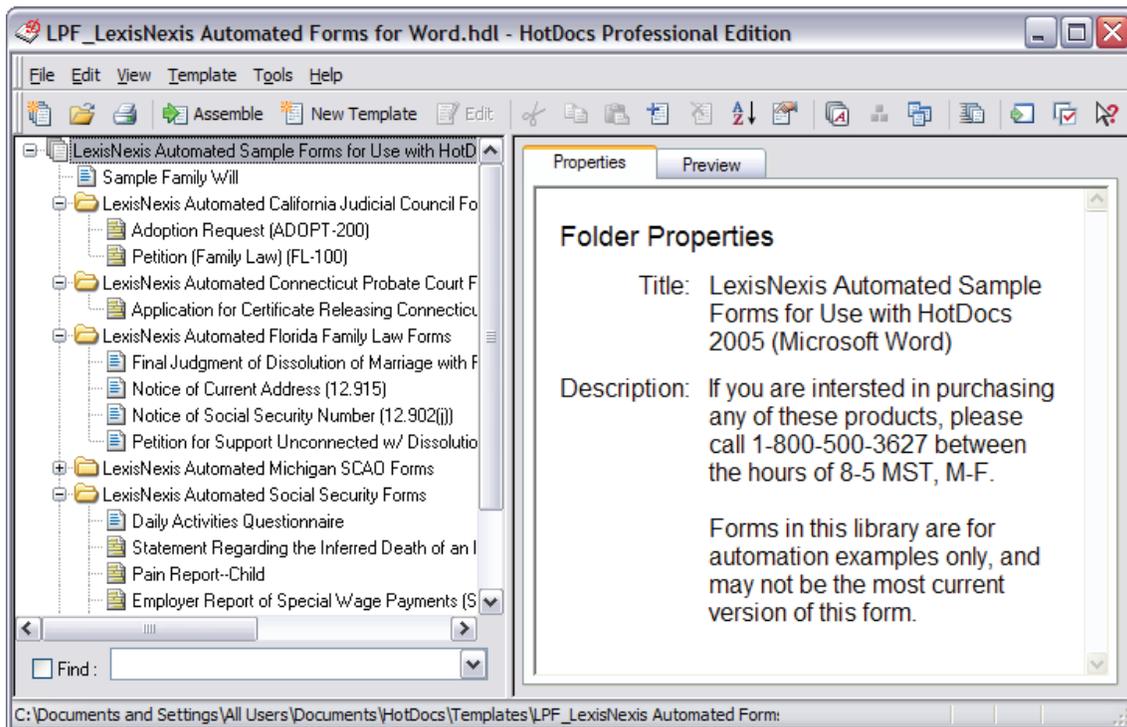
# Using HotDocs Libraries

## Overview: 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.

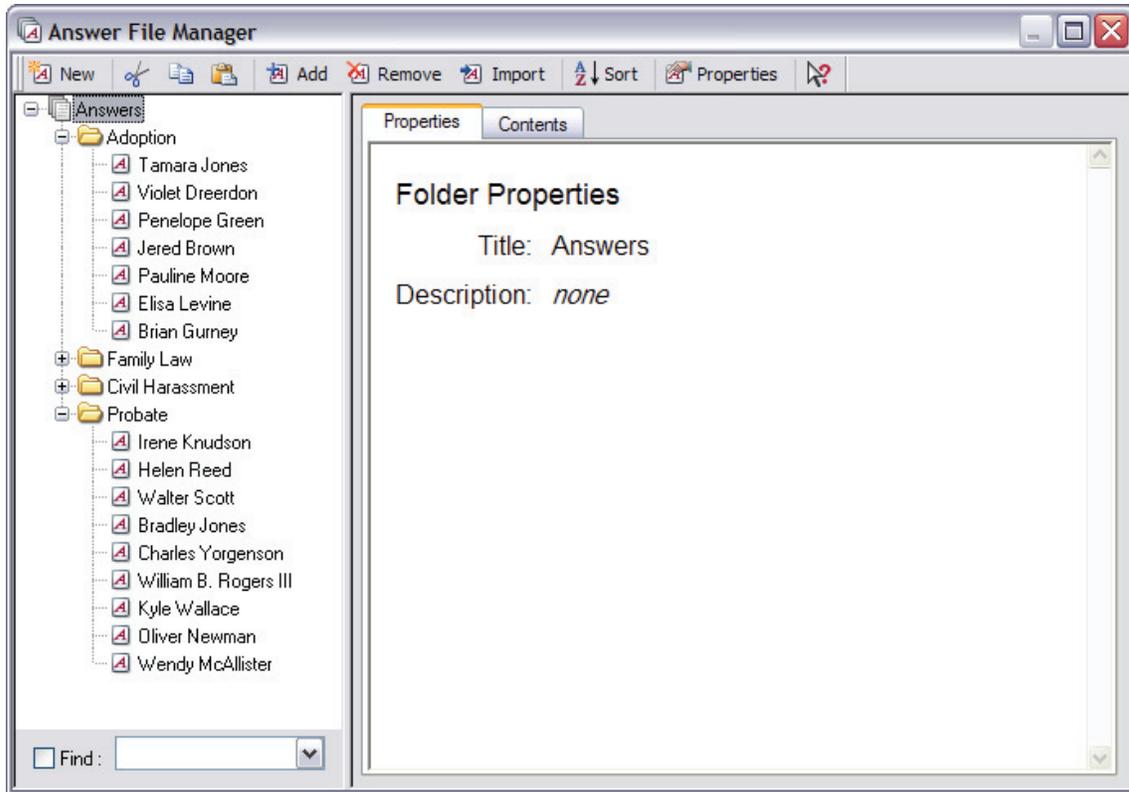
### Example of a template library

Template libraries help you organize template files and provide quick access to many HotDocs tools, such as the Assembly Queue, HotDocs Options, and Component Manager.



### Example of answer library

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.

**Warning:** 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 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**, **Component Manager**, **Template Manager**, and **HotDocs Options**.

# Create 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.

## To create a library

1. At the HotDocs library window, click the  **New Library** button. The **New Library** dialog box appears.
2. Click the  **Browse** button next to the **File name** box. 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** box. (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** box. 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** box. 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](#).

**Note:** 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 and Exit 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.

**Note:** See [Change HotDocs Program File Locations](#) for information on the default location HotDocs looks for and saves library files.

# Add Templates and Other Files to a Library

After you create a library, you can add files to it. This includes folders, template files, Web addresses, and so forth. You can also add these types of files to an existing library.

When you create a new text or interview template, the template is automatically added to the library. Clause libraries created at the template library are likewise added to the library automatically. Form templates, however, often must be added to the library manually.

## To add an item to a 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**, **Model Document**, **Clause Library**, **Auto-Assemble File**, **Web Address**, and **Folder**.)
4. Click the  **Browse** button next to the **File name** box 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** box, 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** box. The description appears in the **Properties** tab when the library item is selected.

### Notes:

- You can simultaneously add multiple files to a library. However, doing this doesn't allow you to assign properties (such as titles and descriptions) to each individual file. To add multiple files, select all the files you want to add in the folder you are browsing. Once they are added, you can modify the properties. (See [Change the Properties of a Library Item](#).)
- You can also add items by dragging files from a Windows Explorer folder to the library. This method allows you to add files like PDF files, word processor documents, and so forth to the library. Once you add the file, you may need to update the item properties (such as add a title or description). See [Change the Properties of a Library Item](#) for details.
- 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.

# 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, from the library, 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:

To	Do This
Assemble a document	Select a template or clause library and click  <b>Assemble</b> .
Create a new template or clause library and have it automatically added to the template library	Select the folder in which you want the item and click  <b>New Template</b> . (See <a href="#">Create a New Text Template</a> or <a href="#">Create a Form Template</a> .)
Create a new folder in the library	Click <b>Add Folder</b> ( <b>Edit</b> menu).
Edit an existing template or clause library	Select the item, then click  <b>Edit</b> . (See <a href="#">Edit a Template</a> or <a href="#">Customize a Clause Library</a> .)
Add an existing item to a library	Select the folder in which you want the item, then click  <b>Add Item</b> . (See <a href="#">Add Templates and Other Files to a Library</a> .)
Group templates you use most frequently in a <i>Favorites</i> folder	Select the template you want to add to the Favorites folder and choose <b>Add to Favorites</b> ( <b>Edit</b> menu). (See <a href="#">Create a Favorites Folder in the Template Library</a> .)
Remove an item from a library	Select the item, then click the  <b>Remove Item</b> button. (See <a href="#">Remove Items from a Library</a> .)

<p>Modify an item's file name, title, or description</p>	<p>Select an item, then click the  <b>Properties</b> button. (See <a href="#">Change the Properties of a Library Item.</a>)</p>
<p>Print a blank copy of the text or form template</p>	<p>Select the template and then choose <b>Print (Template menu)</b>. The document is opened so it can be printed.</p>
<p>View or hide all of the items in a library list</p>	<p>Choose <b>Expand All</b> from the library <b>View</b> menu. When you do this, HotDocs opens all of the subfolders in the library so you can view all of the contents of the library. To hide the list again, choose <b>Collapse All</b>.</p>
<p>Open Windows Explorer to view where the file is saved</p>	<p>Choose <b>Go To (Template menu)</b>.</p>

# 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.

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**.

**Note:** 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 to edit or assemble the other template.

# Change the Properties of a 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.

**Warning:** 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:

To	Do This
Change the file path	In the <b>File name</b> box, type the new file path, or click the  <b>Browse</b> button and specify a new path.
Change the item title	In the <b>Title</b> box, either type a title or click the  <b>Get Title</b> button to use the default title. (Titles can be 100 characters long.)
Change the command-line option(s)	In the <b>File name</b> box, type the new options. (For example, <i>Refund.wpt /af=bwhittington.ans</i> ). (See <a href="#">Overview: Command-Line Options</a> .)
Change the description	In the <b>Description</b> box, type the new description. (The description appears in the <b>Properties</b> tab when a library item (or folder) is selected.)

**Note:** You can change the properties of several other library items at once. See [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.

**Warning:** 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 the Local Disk or Network](#).)

## 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:

To	Do This
Change the file path for the selected items	<p>Click the File path  <b>Browse</b> button and browse to the new folder. Then click <b>OK</b>. The new folder path is inserted. (Click the <b>File path</b> drop-down button to view a list of all the various file paths used for the templates in your list.)</p> <p>To specify the default folder the different files, choose <b>&lt;default&gt;</b> from the <b>File path</b> drop-down list and click <b>OK</b>. (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 <a href="#">Change HotDocs Program File Locations</a>.)</p>
Change the reference path for the selected items	<p>Click the  <b>Reference path</b> button and choose a path from the <b>Select Reference Path</b> dialog box. When you click <b>OK</b>, HotDocs inserts the reference path keyword in the <b>File path</b> box.</p> <p>To remove all reference paths and specify the default folder for the different files, choose <b>&lt;default&gt;</b> from the <b>File path</b> drop-down list and click <b>OK</b>. (The default folder is the location HotDocs looks for the file when no file path is given. You can specify what this</p>

	<p>default location is at HotDocs Options. See <a href="#">Change HotDocs Program File Locations.</a>)</p> <p>The  <b>Reference path</b> button isn't available unless you have at least one reference path defined in the <b>File Locations</b> folder of HotDocs Options. (See <a href="#">Assign Reference Paths to HotDocs Files.</a>)</p>
<p>Change the file name extension for the selected items</p>	<p>Click the <b>File name extension</b> drop-down button and make a selection. If the desired extension is not listed, type it.</p> <p>HotDocs displays a list of the file name extensions for all the items you have selected in the <b>File path</b> drop-down list. If you have multiple file types selected, <b>&lt;various&gt;</b> appears in the <b>File name extension</b> box.</p>
<p>Change the command-line option or options for the selected items</p>	<p>Type the command-line option (for example <i>/af=path and file name</i>) in the <b>Command-line options</b> box and click <b>OK</b>. If assigning multiple options, include spaces between each option. (See <a href="#">Overview: Command Line Options.</a>)</p> <p>To remove existing command-line options, select <b>&lt;none&gt;</b> from the <b>Command-line options</b> drop-down list and click <b>OK</b>.</p>

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

# 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  **Sort** button. The **Folder Sort Options** dialog box appears.
3. Optionally, select **Sort Subfolders** to sort the folders inside the selected folder.
4. 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**.

The template list changes to show the rearranged library items.

**Note:** 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).

# 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. Perform tasks as explained in the following table:

To	Do This
Add a folder to the library so you can group library items	Choose <b>Add Folder</b> ( <b>Edit</b> menu). The <b>Add Item</b> dialog box appears where you can specify the folder name and a description. (Once you create a folder, you can move items to it. See <a href="#">Move Items Within a Library</a> for details.)
Print a list of the titles, file names, and descriptions of templates and folders in a library	Click the  <b>Print Library</b> button. (See <a href="#">Print a List of Items in a Library</a> .)
Remove an item from the library	Select the item and click the  <b>Remove Item</b> button. (This command has different effects for different items. See <a href="#">Remove Items from a Library</a> .)
Remove an item from the library and place a copy of it on the Clipboard	Select the item and click the  <b>Cut</b> button. The item is saved to the Clipboard so you can paste it somewhere else.
Copy an item in the library	Select the item and click the  <b>Copy</b> button. A copy of the item is saved to the Clipboard so you can paste it somewhere else.
Paste an item to a new location in the library	Cut or copy an item, specify the new location, then click the  <b>Paste</b> button.
Sort items in a library in ascending or descending order based on their titles	Select a folder, then click the  <b>Sort</b> button. (See <a href="#">Sort Items in a Library</a> .)

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

# 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).
3. 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.

**Warning:** 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.)
2. Choose **File > Export Library To > HotDocs Library File**. The **Export HotDocs Library File** dialog box appears.
3. At the **File name** box, click  **Browse**, select a location for the new library, and enter a library name in the **File name** box. (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** box, or accept the suggestion HotDocs makes. This title will appear in the **Properties** tab of the HotDocs library window.
5. Optionally, in the **Description** box, 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](#).)

# 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.

**Warning:** 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 you want to print. (See [Open a Library](#).)
2. Click the  **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.

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

## 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.

**Warning:** 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 the Local Disk or Network](#) 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.

## To cut and paste several library items

1. At the HotDocs library window, select the items you want to move.
2. Click the  **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 the Local Disk or Network

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. See [Change the Properties of a Library Item](#) or [Change the Properties for Multiple Items in a Library](#).)

## To move template and clause library files

1. At the HotDocs library window, select the templates and clause libraries you want to move.
2. Select  **Move** (**Template** menu). The **Browse for Folder** dialog box appears.
3. Select the folder where you want to move the files.
4. Click **OK**. HotDocs automatically removes all associated files (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.rtf* 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.

**Note:** You can also copy templates to new locations on disk and have the library reference updated. See [Copy Templates to New Locations](#) for details.

# Copy Templates to New Locations

Often you need to make a copy of a template and save it to a new location. 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.

## To copy the template to a different location

1. At the HotDocs library window, select the templates and clause libraries you want to copy. (Press **Ctrl** or **Shift** to select multiple items.)
2. Choose **Copy (Template menu)**. The **Folder for Copied Templates** dialog box appears.
3. Browse to the folder where you want to copy the template or templates and click **OK**. The **Copy Templates** dialog box appears, asking you to set some properties for the copied templates.
4. 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.)
5. 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.

# 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:

To	Do This
Install specific update items	Select <b>Install</b> for each item you want to install.
Discard an update item to prevent HotDocs from prompting you to install the update	Select <b>Discard</b> for the item. If this box is disabled, the template provider has marked it as a <i>required</i> 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 <b>Discarded updates</b> . The list changes to include discarded items.
View a list of all installed updates	Select <b>Installed updates</b> . The list changes to include installed updates.
Reinstall an update	Select <b>Installed updates</b> , then select the update you want to reinstall from the list.

4. Click **Continue**. HotDocs installs the selected updates.

**Note:** 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](#).)

# 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.
2. Click the  **Add** button. The **Add Item** dialog box appears.
3. At the **Type** drop-down list, select **Web Address**.
4. In the **URL** box, enter the URL. Or, click the  **Browse** button to locate the address. Then close the browser window to save the URL to the **Add Item** dialog box.

**Note:** 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.

5. In the **Title** box, enter a title for the URL. The title identifies the item in the template library.
6. Optionally, type a description in the **Description** box. The description appears at the **Properties** tab.

# Automating Text Templates

---

## Overview: 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.

- A text template is created and automated 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. (See [Create a New Text Template File](#).)

**Note:** HotDocs assembles all Word templates using an RTF-based assembly process. This means that if your template is in DOT format, HotDocs must open it, convert it to RTF, and then assemble it. By developing the templates in RTF, you can reduce the number of steps in this process, thus decreasing the amount of time it takes to assemble a document.

- A form template uses 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 PDF Advantage to create a PDF version of the form. Finally, you use HotDocs Automator to automate the template. (See [Create a Form Template](#).)
- An interview template gathers common information (such as court, attorney, or client information) and saves 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](#).)

### 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 example shows a fully automated text template with its component file.

IN THE CIRCUIT COURT OF THE «Court judicial circuit TE:LIKE THIS»  
 IN AND FOR «Court county MC::LIKE THIS» COUNTY  
 «IF Print family law division TF»FAMILY LAW DIVISION

«END IF»  
 «IF Bold caption heading TF»  
 IN THE CIRCUIT COURT OF THE «Court judicial circuit TE:LIKE THIS»  
 CIRCUIT,  
 IN AND FOR «Court county MC::LIKE THIS» COUNTY  
 «IF Print family law division TF»FAMILY LAW DIVISION

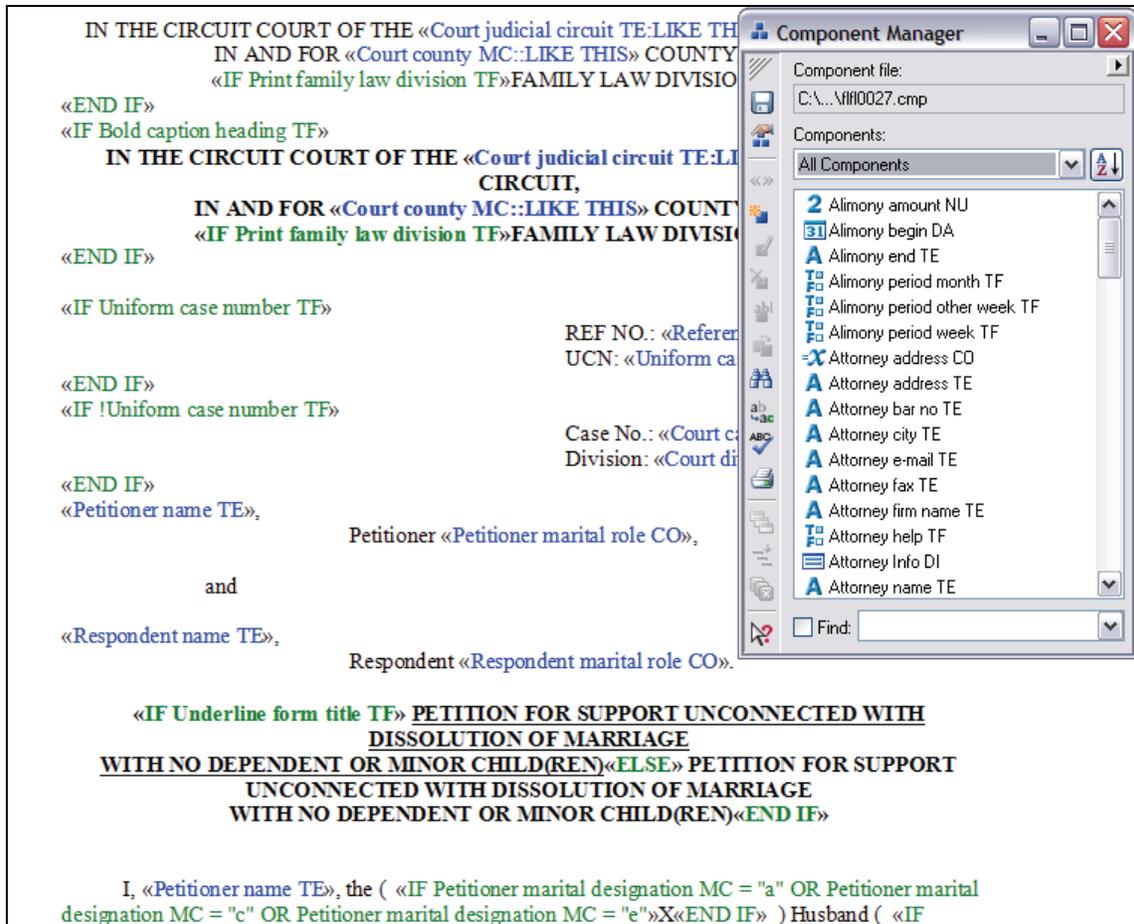
«END IF»  
 «IF Uniform case number TF»  
 REF NO.: «Referen  
 UCN: «Uniform ca

«END IF»  
 «IF !Uniform case number TF»  
 Case No.: «Court c  
 Division: «Court di

«END IF»  
 «Petitioner name TE»,  
 Petitioner «Petitioner marital role CO»,  
 and  
 «Respondent name TE»,  
 Respondent «Respondent marital role CO».

«IF Underline form title TF» **PETITION FOR SUPPORT UNCONNECTED WITH  
 DISSOLUTION OF MARRIAGE**  
**WITH NO DEPENDENT OR MINOR CHILD(REN)«ELSE» PETITION FOR SUPPORT  
 UNCONNECTED WITH DISSOLUTION OF MARRIAGE  
 WITH NO DEPENDENT OR MINOR CHILD(REN)«END IF»**

I, «Petitioner name TE», the ( «IF Petitioner marital designation MC = "a" OR Petitioner marital  
 designation MC = "c" OR Petitioner marital designation MC = "e"»X«END IF» ) Husband ( «IF



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 Editing and Navigation Toolbar Buttons

When you create a text template, HotDocs adds an editing toolbar as well as a navigation toolbar to the word processor window. This toolbar lets you perform basic HotDocs tasks, such as inserting variables, opening Component Manager, test assembling the template, or matching REPEAT and IF instructions with their END instructions.

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. The following is a list of these buttons, followed by a description of what each button does:

**Note:** To access these buttons in Word 2007, click the **HotDocs** tab on the Word ribbon.

Button Name	What It Does
<b>HotDocs</b> 	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.)
<b>Variable Field</b> 	Opens the <b>Variable Field</b> 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 <b>Variable Field</b> dialog box with that information already appearing.  <b>Note:</b> If you're using Word 2000 or higher, you can double-click in a variable field and the <b>Variable Editor</b> will appear.
<b>IF Field</b> 	Opens the <b>IF Field</b> dialog box where you can create a true/false test for conditional text. If you need a more complex test, click <b>IF Expression</b> at the <b>IF Field</b> dialog box.
<b>ASK Field</b> 	Opens the <b>ASK Field</b> 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.
<b>INSERT Field</b> 	Opens the <b>INSERT Field</b> 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.
<b>REPEAT Field</b> 	Opens the <b>REPEAT Field</b> 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.
<b>SPAN Field</b> 	Opens the <b>SPAN Field</b> dialog box where you can enter a name for Span component you are creating. SPAN fields represent the section of text you

<p><b>Note:</b> SPAN fields are supported in Microsoft Word only.</p>	<p>want to allow the user to edit while viewing the <b>Document Preview</b> tab. (Giving users access to the assembled document from the assembly window lets users save document text changes to an answer file so they can reassemble the document later and still maintain those changes.)</p>
<p><b>Edit Component</b>  </p>	<p>Opens the <b>Component Editor</b> 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.)</p>
<p><b>Test Assemble</b>  </p>	<p>Test assembles the document or a selected portion of the document.</p>
<p><b>Component Manager</b>  </p>	<p>Opens the <b>Component Manager</b> window where you can create, edit, delete, rename, and copy components.</p>
<p><b>Clause Library</b>  </p>	<p>Opens a clause library where you can create new clauses or add existing clauses.</p>
<p><b>Save</b>  </p>	<p>Saves the template and component file you are currently editing.</p>
<p><b>Save and Close</b>  </p>	<p>Saves and closes the template and component file you are currently editing.</p>
<p><b>Help</b>  </p>	<p>Displays the HotDocs Help viewer, which contains information on automating templates and assembling documents. You can search the help file using the <b>Contents</b> tab, the <b>Index</b> tab, or the full-text <b>Search</b> tab. (See <a href="#">Get Help While Using HotDocs</a>.)</p>
<p><b>Apply Colors</b>  </p>	<p>Assigns custom colors to the different field types in your templates. This can help you more quickly identify sections or parts of your template as you are automating. You can customize the colors at <b>HotDocs Options</b>. (See <a href="#">Define Colors for Text Template Fields and Instructions</a>.)</p>
<p><b>Label Fields</b>  </p>	<p>Opens the <b>Label Fields</b> 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.</p>

<p><b>Match Fields</b></p> 	<p>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.</p>
<p><b>Go to Field</b></p> 	<p>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.</p>
<p><b>Previous and Next Fields</b></p> 	<p>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.</p>
<p><b>Markup View</b></p>  <p><b>Developer View</b></p> 	<p>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). Developer View is the mode in which you automate the template. (See <a href="#">View the Template in Markup View.</a>)</p>
<p><b>HotDocs Outliner</b></p> 	<p>Opens the HotDocs Outliner so you can review an outline of scripting (and variables) used in the template. See <a href="#">View an Outline of Scripting in the Template.</a></p>

**Note:** You should always use the HotDocs  **Save** and  **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.

# Create 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.

**Warning:** 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. (See [Create a Library](#).)

## 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**. The **New Template** dialog box appears.
3. Select the type of text template you want to create from the **Type** drop-down list. Depending on which word processor(s) you have installed, your options include **Word RTF Template (.rtf)**, **Word Document Template (.dot)**, and **WordPerfect Template (.wpt)**.
4. Type a file name in the **File name** box. HotDocs automatically adds the correct extension to the file name based on the type you have selected. (To save the template in a location other than the default *Templates* folder, include the folder path with the file name in the **File name** box. You can check the default *Templates* location at the **HotDocs Options** dialog box. See [Change Word Processor File Locations](#).)
5. Type a title for the template in the **Title** box (or accept the suggestion HotDocs makes). The title is what appears in the library.
6. Optionally, enter a description in the **Description** box. Descriptions appear in the **Properties** tab of the library when the template is selected.
7. Select the contents for the new template:
  - To create an empty template file, select **Empty**. (You can type or copy and paste text into your template later.)
  - To copy the contents of the currently open word processor file into the new template, select **Current open word processor document**.
  - To copy the contents of some other file (for example, a word processor file or another template file) into the new template, click the  **Browse** button next to **Other file** box and locate the file.
8. Click **OK** at the **New Template** dialog box. HotDocs opens your selected word processor and displays the template. You can use the HotDocs editing toolbar to begin automating the template. (See [HotDocs Editing and Navigation Toolbar Buttons](#).)

## Notes:

- If you are using Microsoft Word to develop your templates, it is recommended that you create RTF (Rich Text Format) templates. RTF templates assemble much faster than .DOT templates. (See [Overview: Template and Component Files](#) for more details.)
- When you create a new, empty Word template, HotDocs applies whatever styles are in *HotDocs6.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** box) that allow you to specialize the assembly process for a given template. See [Overview: Command-Line Options](#).
- If you have multiple word processors installed, you can have HotDocs always suggest a specific

template type when you create a new template. See [Change Your Default Word Processor](#).

# Create 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 applies the styles in that file to the new template. It also copies the template text (including components) into the new template file.

**Warning:** 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. (See [Create a Library](#).)

## To create a new template based on an existing one

1. At the HotDocs library, select the existing template from the template list.
2. Click  **New Template**. The **New Template** dialog box appears.
3. Change the file name and title for the new template (at the **File name** and **Title** boxes).
4. Do *not* change the existing template's file path and name in the **Other file** box.
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.

**Note:** Another way to create a new template based on an existing template is to use the **Copy Templates** command. (See [Copy Templates to New Locations](#).) Once you have copied the template and component file, simply rename the files using Template Manager. See [Rename Templates Using Template Manager](#).

# Convert a Single Template to a New File Format

**Note:** To convert multiple files at once or to convert all templates to RTF, use Template Manager. (See [Convert Multiple Templates to Work with HotDocs 2008](#) and [Convert Templates and Clauses to Microsoft RTF](#).)

HotDocs 2007 uses a file format for component files that is different from earlier versions of HotDocs, including HotDocs 5, HotDocs 6, and HotDocs 2005. (HotDocs 2006, HotDocs 2007, and HotDocs 2008 use the same file format.) This means that all component files created in these earlier versions must be converted to HotDocs 2008 format in order for them to work. Failure to convert the files may result in errors when you attempt to assemble documents using them.

Additionally, perhaps you've changed word processors and need to convert all your templates from one file type to another.

**Warning:** It is a good idea to back up templates, component files, or other HotDocs files before making major changes to them. This is especially true when converting HotDocs 5 component files to work with HotDocs 2008—once a version 5 component file has been converted, it can no longer be used with HotDocs 5.

## To convert an older HotDocs template to HotDocs 2008 format

1. At the template library, select the template you want to convert and click  **Edit Template**. HotDocs warns you it is about to convert the component file to a new format.
2. Click **Yes** to confirm this. The component file is converted and the template opens.
3. 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 2008](#).

## 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** box 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** box.
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 [Delete a Library Item](#).)

**Note:** You can designate that component files always be compatible with a specific version of HotDocs 6 / 200x. See [Change Component File Properties](#) for details. (When doing this, however, you must be careful not to use features in the template that are not backwards compatible. For example, if you are using HotDocs 2008 to edit a template that you want to be compatible with HotDocs 2005, you should not use any features that were introduced in HotDocs 2006, HotDocs 2007, or HotDocs 2008 because they may not work with HotDocs 2005.)

# 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.

# Assign Colors to Fields and Instructions in Templates

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 Colors for Text Template Fields and Instructions](#).)

## To apply custom colors to template fields and instructions

1. Open the template for editing. (See [Edit a Template](#).)
2. Click the  **Apply 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.)

### Notes:

- If you change your default colors, or if you receive templates from other developers whose colors are different from yours, you can click the  **Apply 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** drop-down menu in the HotDocs toolbar and choosing **Apply Colors**.

# Move Between Fields in a 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](#).)
2. 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 double-clicking in the field.)

**Note:** 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**.

# Use Labels to Identify Instructions

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 *L1*, 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.)

## To label instructions

1. At the text template, click the  **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.

**Notes:**

- SPAN fields are supported in Microsoft Word only.
- To clear labels from fields or show comments once again, click the  **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**.

# 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

1. At the template, click the  **Go to Field** button. The **Go To Field** dialog box appears.
2. Enter a number in the **Field number** box.
3. Click **OK**. Your cursor is taken to that field.

# View the Template in Markup View

**Note:** Markup View is supported in Microsoft Word templates only.

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 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  **Markup View** button. The template changes to show the marked up template.

### Notes:

- To view a developer version of the template again, click the  **Developer View** button.
- 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.

# View an Outline of Scripting in the Template

**Note:** 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  **HotDocs Outliner** button. The **HotDocs Outliner** window appears.
3. Perform any tasks, as described in the following table:

To	Do This
Arrange the Outliner with the Microsoft Word window	Adjust the Outliner to the width you want and click the  <b>Arrange Windows</b> button.
View all instructions in the template—including all <i>nested</i> instructions	Click the  <b>Expand All</b> button. The outline expands to show all levels of instructions.  To collapse the interview outline, click the  <b>Collapse All</b> button.
Update the Outliner after making changes in the template	Click the  <b>Refresh Outline</b> 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  <b>Synch Template</b> 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  <b>Show Variables</b> 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	<p>Click the  <b>Apply Colors</b> button. The <b>Apply Colors</b> dialog box appears, where you can choose the color scheme you want to use in the outline:</p> <ul style="list-style-type: none"> <li>■ Choose <b>None</b> to make all variable and instruction entries black.</li> <li>■ Choose <b>Regular</b> 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.</li> <li>■ Choose <b>Nested</b> 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.</li> <li>■ Choose <b>Sequential</b> 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.</li> </ul>
Copy the contents of the script outline to the Windows Clipboard so you can paste it into another application	Click the  <b>Copy to Clipboard</b> 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 <b>Limit to</b> 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 variable.
View the conditions or other instructions surrounding the cursor position in the template	Select <b>Limit to</b> and then choose <b>&lt;cursor position&gt;</b> 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 <b>Edit</b> or <b>Test</b> .
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 <b>Find Matching END</b> . In the template, the instruction is highlighted.

# 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.

**Note:** Contact a HotDocs sales representative for information about obtaining these free DLLs. Once you have the DLL, it must be saved to the HotDocs program folder.

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.

## 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** box. 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.
5. 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: If you're automating a WordPerfect template, insert a LANGUAGE instruction before the non-English text by creating a variable (or by copying and pasting an existing variable) and then replacing the text between the chevrons (« ») with the language instruction. For example: «LANGUAGE FRA».)
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

8. Optionally, enter the punctuation character you want to use for the **Thousands separator** and the **Decimal separator** in the appropriate boxes.
9. Click **OK**. The instruction is merged in the template.

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.

**Notes:**

- 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** box.

# 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.
2. 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.

**Note:** 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 File](#)), as well as edit clause components in a clause library.

**Warning:** 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:

To	Do This
Sort components in Component Manager either by alphabetical name or by component type	Click the  <b>Sort Components</b> button and then choose either <b>Sort by Name</b> or <b>Sort by Type</b> .  To sort component lists that appear in the <b>Dialog Editor</b> and <b>Computation Editor</b> , 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  <b>Arrange Windows</b> 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  <b>Save Components</b> button.
Specify certain assembly and interview preferences for the template and component file	Click the  <b>Component File Properties</b> button. (See <a href="#">Change Component File Properties</a> .)
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 <b>Components</b> list, and click the  <b>Insert Variable</b> button. (See <a href="#">Create and Insert a Variable Using Component Manager</a> .)

<p>Create a new component, including a variable, dialog, or clause</p>	<p>Click the <b>Components</b> drop-down button, select the type of component you want to create, and click the  <b>New Component</b> button.</p>
<p>Make changes to an existing component</p>	<p>Select the component from the component list and click the  <b>Edit Component</b> button. (You can also double-click the component.)</p>
<p>Remove a component from the component file</p>	<p>Select the component in the component list and click the  <b>Delete Components</b> button.</p> <p><b>Warning:</b> 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.</p>
<p>Assign a new name to a component</p>	<p>Select the component(s) and click the  <b>Rename Component</b> button. (See <a href="#">Rename Components in a Single Template.</a>)</p>
<p>Make a duplicate copy of a variable or group of variables</p>	<p>Select the variable(s) and click the  <b>Duplicate Variables</b> button. (See <a href="#">Duplicate a Variable.</a>)</p>
<p>Bring all open Component Editors to the front</p>	<p>Click the  <b>Restore All</b> button. HotDocs brings all open Component Editors to the front so you can view and edit them.</p>
<p>Minimize all open Component Editors to the Windows taskbar</p>	<p>Click the  <b>Minimize All</b> button. All open Component Editors are then minimized. However, you can still access them by clicking their icon in the Windows taskbar.</p>
<p>Close all open Component Editors</p>	<p>Click the  <b>Save and Close All</b> button. All changes made to components are saved and all open Component Editors are closed.</p>

**Notes:**

- 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, Print, 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—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  **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  **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.

### Notes:

- 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 HotDocs 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 [Overview: Template Manager](#).)

# 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.
3. Change any of the following options:

### General Properties

To	Do This
Specify a default title for the template	At the <b>General</b> tab, enter the template title in the <b>Template title</b> box. This title is used to identify the template in the template library file list.
Include a short description of the template, such as how the template should be used or when it was created	At the <b>General</b> tab, enter a description in the <b>Template description</b> box. The description appears when the user views its properties at the library <b>Properties</b> tab.
Specify with which version of HotDocs the template will be used	At the <b>General</b> tab, click the <b>Component file format</b> drop-down button and choose the version of HotDocs you need: <ul style="list-style-type: none"> <li>■ Choose <b>Current version</b> to save it in the latest version of HotDocs—in this case, HotDocs 2006-2008. When you upgrade to a future version of HotDocs, HotDocs automatically upgrades the component file version so you don't have to (as you do if you choose one of the other options).</li> <li>■ Choose <b>HotDocs 6 / HotDocs 2005 / HotDocs 2006-2008</b> to save the</li> </ul>

	<p>component file in that specific version. To upgrade or downgrade the version, you must manually choose the new version here.</p> <p><b>Warning:</b> Be careful not to use new features that aren't supported in the older versions of HotDocs or you may experience errors during assembly.</p>
<p>Specify a shared component file for the template</p> <p>To 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.</p>	<p>At the <b>General</b> tab, click the <b>Shared component file</b> drop-down button and choose the component file from the list. (Shared and pointed component files must be stored in the same folder.) See <a href="#">Use One Component File for Multiple Templates and Specify Whether Component File Properties are Shared</a>.</p>

### Interview Properties

To	Do This
<p>Choose whether to use your own interview script or have HotDocs generate a default interview</p>	<p>At the <b>Interview</b> tab, if you choose to create and use your own interview component, select <b>Use custom interview</b> and enter the name of the computation in the <b>Interview component</b> box. (In HotDocs 2005 and earlier, this component's name was INTERVIEW. You can continue to use this name, or you can use any name—as long as it's specified here.)</p> <p>If you choose to let HotDocs generate a default interview, select <b>Generate default interview</b>. HotDocs then asks the variables and processes the instructions in the template in the order it "reads" them. If a variable is linked to a dialog, the dialog is displayed instead.</p>
<p>Automatically disable variables in dialogs that aren't referred to in the document</p>	<p>Select <b>Automatically disable irrelevant variables and dialogs</b>. (See <a href="#">Automatically Disable Unused Variables in Interviews</a>.)</p>
<p>Display just answer-gathering dialogs without showing the interview outline</p>	<p>At the <b>Interview</b> tab, select <b>Hide interview outline</b>. The only way the user can navigate the interview is by using the  <b>Previous</b> and  <b>Next</b> commands.</p>

<p>Keep the <i>End of Interview</i> dialog from appearing</p>	<p>At the <b>Interview</b> tab, select <b>Hide End of Interview</b> dialog. When the user clicks  <b>Next</b> at the last dialog in the interview, HotDocs will either send the assembled document to the word processor or it will display the <b>Document</b> tab, depending on which <b>End of Interview</b> action the user has defined. (See <a href="#">Control Whether Users See at the End of Interview Dialog</a> and <a href="#">Control What Happens When You Finish an Interview.</a>)</p>
<p>Have HotDocs automatically generate titles for dialogs based on the contents of the dialog</p> <p>The way HotDocs generates default dialog titles is by seeing if the first item in a dialog is dialog element text. If so, it displays that text as the dialog title in the interview outline. If not, it uses the prompt from the first variable and appends the word <i>"etc."</i> For example, if you have a dialog gathering personal information about an employee and you have not used any additional text to introduce the dialog, HotDocs looks at the prompt of the first variable in the dialog, <i>Employee name</i>, and creates a dialog title called <i>Employee name, etc.</i></p>	<p>At the <b>Interview</b> tab, select <b>Generate default titles for dialogs</b>.</p> <p>If you want to create your own custom dialog titles, assign them at each dialog's Dialog Editor. (See <a href="#">Change a Dialog's Options.</a>) Once you specify a title for a dialog, HotDocs will not generate them.</p>
<p>Specify the text that will be used for the resource button ToolTip</p>	<p>At the <b>Interview</b> tab, enter the text in the <b>Resource button label</b> box that you want users to see when they hover their mouse cursor over the resource button during the interview.</p>

### Assembly Properties

To	Do This
<p>Specify a product title or template set name that appears in the assembly window title bar</p>	<p>At the <b>Assembly</b> tab, enter a title in the <b>Product title</b> box. (If you leave this field empty, HotDocs will use the text <i>HotDocs 2008.</i>)</p>
<p>Keep users from saving their answers or using an existing answer file for this particular template</p>	<p>At the <b>Assembly</b> tab, select <b>Do not use answer files</b>.</p> <p>Once selected, this option causes the first information-gathering dialog to appear immediately after the template is selected for assembly. At the end of assembly, users</p>

	cannot save their answers.
Use variable names instead of prompts when generating Question and Answer Summaries	At the <b>Assembly</b> tab, select <b>Use variable names in summaries</b> .
Position the cursor at a certain place in the document after it has been assembled	Create a word processor bookmark in the template and then, at the <b>Assembly</b> tab, select <b>Move to "TypeHere" bookmark</b> . (See <a href="#">Position the Cursor in the Assembled Document</a> .)
Automatically update the document's table of contents and/or index once the document has been sent to the word processor	At the <b>Assembly</b> tab, select <b>Update table of contents, references, fields, etc.</b>  Once the document is sent to the word processor, these references will be updated based on any changes made during assembly of the document.
Let users see their answers highlighted in the assembled Microsoft Word document	At the <b>Assembly</b> tab, select <b>Mark answers in assembled documents</b> .  When you do this, HotDocs marks merge fields in the <b>Document Preview</b> tab. Users can then click the  <b>Highlight Answers</b> button in the assembly window and answers will be highlighted with a special color.
Let users edit answers while viewing the assembled document in the <b>Document Preview</b> tab (Microsoft Word users only)	At the <b>Assembly</b> tab, select <b>Enable Edit Answer at the Document Preview</b> tab.  As users change answers, answers throughout the document (and interview) will be updated to reflect the change.
Designate what text will be merged in the document if the user chooses to not answer a question	At the <b>Assembly</b> tab, click the <b>Unanswered variable placeholder</b> drop-down button and choose an option: <b>Default</b> , <b>[Variable]</b> , <b>***Variable***</b> , <b>Underscores</b> , <b>Asterisks</b> , or <b>Nothing</b> . See <a href="#">Tips on Using Unanswered Variable Placeholders</a> for an explanation.
Specify a Word document template where you want to store your post-assembly macros (Word RTF users)	At the <b>Assembly</b> tab, click the  <b>Browse</b> button next to the <b>Post-assembly macro file (Word only)</b> box and locate the template that contains these macros.  See <a href="#">PLAY "MACRO"</a> and <a href="#">Specify a Template for Storing Post-Assembly Macros</a> for more

	information.
Prevent HotDocs from infinitely processing a WHILE expression, which will cause HotDocs to stop responding.	At the <b>Assembly</b> tab, type a number in the <b>Maximum WHILE iterations</b> box. 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.
Prevent HotDocs from infinitely recursing (or processing) a computation, which will cause HotDocs to stop responding  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 <i>processing stack</i> . If too much information gets added to this stack, HotDocs may stop responding.	At the <b>Assembly</b> tab, type a number in the <b>Maximum processing stack depth</b> box. This number represents how many instructions you want HotDocs to allow in the processing stack (see note in previous column). When HotDocs reaches this limit, the recursion will stop.

### Other Properties

To	Do This
Have HotDocs automatically remove hidden data from the assembled document	At the <b>Hidden Data</b> tab, select which types of data you want to remove from the assembled document. See <a href="#">Remove Hidden Data from Assembled Documents</a> for more details.
Enable your templates for server-based assembly	At the <b>HotDocs Server</b> tab, select <b>Enable template for use with HotDocs Server</b> . See <a href="#">Enable Templates for Use with HotDocs Server</a> for a description of all of these options.  If you do not select this option, you will not be able to browser test the template or publish it for use with HotDocs Server.
Have HotDocs upload answer files to a Web server	At the <b>Answer Upload</b> tab, fill in the information about answer uploading in the corresponding fields. (See <a href="#">Specify Options to Upload Answers</a> .)

# 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.

**Warning:** Deleting a component removes it 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. If you are sharing components across multiple files, this must be done in each template that uses the component file.

## 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.

### Notes:

- 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.

**Note:** 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

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 in 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** box, 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 HotDocs 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  **Rename Component** button. The **Rename Component** dialog box appears.
3. Type the new component name in the **New name** box 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 HotDocs Template](#).)

## To rename multiple components using Component Manager

1. Open Component Manager. (See [Open and Close Component Manager](#).) The **Component Manager** window appears.
2. Select the components from the **Components** list and click the  **Rename Components** 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** box, and then type the new word or phrase (i.e. *Client*) in the **with** box, 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 HotDocs Template](#).)

HotDocs lists all the components that were renamed in the **Renamed Components** list of Component Manager. This list will retain this information until you either rename other components or make changes to the component file that require similar lists to be displayed (for example, performing a *Find and Replace*). To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

**Note:** 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](#).)

# 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 Manager** 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** box).
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

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** box, and then type the new word or phrase (i.e. *Plaintiff*) in the **with** box, 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. This list will retain this information until you either duplicate other variables or make changes to the component file that require similar lists to be displayed (for example, performing a *Find and Replace*). To view the full list of components again, click the **Components** drop-down button and choose **All Components**.

# Edit Formats, Merge Text, Dialog Elements, and Patterns

You can edit example formats, dialog elements, merge text, and patterns using Component Manager.

## To edit variable formats using Component Manager

1. Open Component Manager. (See [Open and Close Component Manager](#).) The **Component Manager** window appears.
2. Click the **Components** drop-down button and select the variable format you want to edit (for example, **Text Formats**). HotDocs displays the list of available formats.
3. Select a specific format and click the  **Edit Component** button. The **Format** dialog box appears.
4. Make any changes you need.

## To edit merge text using Component Manager

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**.
3. Select a specific merge text component and click the  **Edit Component** button. The **Merge Text** dialog box appears.
4. Make any changes you need.

## To edit dialog elements using Component Manager

1. Open Component Manager. (See [Open and Close Component Manager](#).) The **Component Manager** window appears.
2. Click the **Components** drop-down button and select **Dialog Elements**.
3. Select a dialog element component and click the  **Edit Component** button. The **Dialog Element Editor** appears.
4. Make any changes you need.

## To edit text patterns using Component Manager

1. Open Component Manager. (See [Open and Close Component Manager](#).) The **Component Manager** window appears.
2. Click the **Components** drop-down button and select **Text Pattern**.
3. Select a specific pattern and click the  **Edit Component** button. The **Text Pattern** dialog box appears.
4. Make any changes you need.

### Notes:

- 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** box. This name will be displayed in format lists instead of the format itself.

# Search, Print, and Spell Check Components

Using Component Manager, you can search for literal text strings in component prompts, titles, dialog element text, scripts, patterns, formats, and merge text. Likewise, you can print lists of properties for selected components, and you can spell check text-based properties of components, such as prompts, titles, and dialog element 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  **Find in Components** button. The **Find in Components** dialog box appears.
3. 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 of Component Manager), **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.
2. Click the  **Find and Replace in Components** button. The **Find and Replace in Components** dialog box appears.
3. 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 of Component Manager), **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.

- Specify which properties of a component you want to search by selecting an option. (See [Component Properties](#) for an explanation of each property.)
- 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 print a list of properties for selected components

- Open Component Manager. (See [Open and Close Component Manager](#).) The **Component Manager** window appears.
- Click the  **Print Components** button. The **Print Components** dialog box appears.
- From the **Include** list, select one of the following options: **All components in list** (prints all of the components currently displayed in the **Components** list of Component Manager) , **Selected components in list** (prints only the components that are currently selected (highlighted) in the **Components** list) , **All components in component file** (prints all of the components in Component Manager, regardless of what is currently displayed (and selected) in the **Components** list) .
- Select the properties you want to print. (See [Component Properties](#) for an explanation of each property.)
- Click **OK** and specify any printer options.

#### To spell check components

- Open Component Manager. (See [Open and Close Component Manager](#).) The **Component Manager** window appears.
- Click the  **Spell Check Components** button. The **Spell Check Components** dialog box appears.
- From the **Include** list, select one of the following options: **All components in list** (spell checks all of the components currently displayed in the **Components** list of Component Manager), **Selected components in list** (spell checks only the components that are currently selected (highlighted) in the **Components** list), **All components in component file** (spell checks all of the components in Component Manager, regardless of what is currently displayed (and selected) in the **Components** list).
- Select the component properties you want to spell check. (See [Component Properties](#).)
- Click **OK**. If HotDocs finds any unrecognized words, it displays the **Check Spelling** dialog box.
- Make your selection, based on information in the following table:

To	Do This
Ignore the current instance of the word and continue spell checking	Click <b>Ignore</b> .
Ignore all instances of the word and continue spell checking	Click <b>Ignore All</b> .

Correct only the current instance of the word and continue spell checking	Select an existing replacement from the <b>Change to</b> list (or type the replacement in the <b>Change to</b> box) and click <b>Change</b> .
Correct all instances of the misspelled word and continue spell checking	Select an existing replacement from the <b>Change to</b> list (or type the replacement in the <b>Change to</b> box) and click <b>Change All</b> .
Add the word to your personal dictionary so that the spelling checker will not question the word again	Click <b>Add</b> .
Display additional spelling alternatives for the unrecognized word	Click <b>Suggest</b> . <b>Note:</b> 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 <b>Options</b> . (You can also change your options at the <b>HotDocs Options</b> dialog box. See <a href="#">Change Your Spell Checking Options</a> .)

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**.

**Note:** 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.

# Use One Component File for Multiple Templates

**Warning:** It's a good idea 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 will appear in all templates that use that 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. *All templates, pointed component files, and shared component files must be saved in the same folder.*

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. Otherwise, you must copy those components to the shared file so HotDocs can find them. (See [Copy Components From One File to Another](#).) (Pointing the current component file to the shared file does not automatically copy the components there.)

When specifying component file properties, properties from the shared component file are used 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 some of the properties you want to use are specified in the shared component file, you can copy them to the pointed component file. See [Specify Whether Component File Properties are Shared](#) for details.

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** box.

To unpoint a component file, see [Make Templates Stop Sharing Component Files](#).

## To point the current component file to a shared component file

1. Make sure all templates and component editors are closed. (The only window that should be open is the template library.)
2. At the template library, select the template whose component file you want to point.
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 select the shared component file. (This drop-down list shows all of the component files listed in the same folder as the actual component file. Remember, pointed and shared component files must be saved 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**. (Any properties you change after choosing to use the pointed component file will not be updated in the shared file.)

7. 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 pointed.)

**Warning:** If you created variables before you pointed the component file to the shared file, you must copy those variables to the shared file. See [Copy Components From One File to Another](#) for details.

**Note:** 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. See [Display Certain HotDocs Warnings](#) for details.

# 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.

**Warning:** 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 shared) component file.

### Notes:

- 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** box.
- 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.

**Warning:** 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 users to 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.
2. 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.
4. 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** tab. (See [Control What Happens When You Finish an Interview](#).)

**Note:** 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 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 an 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 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** box, 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.)

**Note:** 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

**Note:** 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.

**Note:** In addition to removing hidden data from assembled documents, you can also remove hidden data from Word RTF templates you're automating. See [Remove Hidden Data from Word RTF 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 your 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.

**Warning:** For complete explanations of each individual property, view the **What's This** help for

each option on the tab, or see the Microsoft Word documentation.

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.

# Remove Hidden Data from Word RTF Templates

**Note:** Removing hidden data is supported in Microsoft Word RTF templates only.

As you work in Microsoft Word RTF 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.

**Note:** 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 RTF templates

1. Open a template library. (See [Open a Library](#).)
2. Select the RTF 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 your 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.

**Warning:** 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.

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.

Once you've selected which data you want to remove, click **Remove Data**. HotDocs removes the data from the selected templates.

# HotDocs Interviews

---

## Overview: HotDocs Interviews

Once you automate a template, the template can be used to generate an interview, which displays the questions the user must answer in order to produce a complete document.

How HotDocs generates this interview is important to understand. To start, HotDocs reads through the template text and extracts all variables and instructions from it. This produces what is called an extracted script.

This example shows a short template, followed by its 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 this script, it then processes the script and asks any variables referenced in it, displaying those questions in the interview. (When a variable is linked to a dialog, HotDocs displays the dialog instead.) Additionally, HotDocs processes other instructions (such as ASK, REPEAT, or IF instructions) included in the extracted script. While the extracted script may refer to a variable or instruction multiple times, once the variable or instruction has been answered, it will not be asked again.

For most templates, allowing HotDocs to generate the interview this way is sufficient. In fact, all templates are set, by default, to use this method.

For complicated templates that contain a lot of scripting, however, this default interview generation may produce an interview that is sluggish for the user to navigate. This is because HotDocs must repeatedly process every variable and instruction in the extracted script—even if the user has already answered the

question or the instruction has already been executed. Depending on the complexity of the script, processing this script may cause a delay when moving between answers in the dialog or between dialogs in the interview.

If this happens, you can create a custom interview that contains only the required scripting and variables. You create this script using a Computation variable. Once you write the custom interview script, you must select a component file property that tells HotDocs to use this script instead of the default extracted script. The following example shows a custom interview script that might be created for the template shown above:

(Custom interview script)

ASK Client Information

IF Client Marital Status = "Married"

    ASK Spouse Name

END IF

# Create an Interview Template

You can create an interview template, which can gather common information (such as court, attorney, or client information) and save the answers 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, users cannot generate documents from an assembled interview template—they only generate answer files.

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. (See [Suggest an Answer Source for Dialogs](#) and [Share Answers Between Two Dialogs](#).)

**Note:** 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 automatically lock answer file usage and save the answers to a specified answer file. See [Overview: Command-Line Options](#).

## 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**. The **New Template** dialog box appears.
3. Click the **Type** drop-down button and choose **HotDocs Interview Template**.
4. In the **File name** box, enter a file name. HotDocs automatically adds the correct extension (.CMP) to the file name. To save the template in a location other than the default *Templates* folder, include the folder path with the file name in the **File name** box. (You can check the default *Templates* location at the **HotDocs Options** dialog box. See [Change HotDocs Program File Locations](#).)
5. Enter a title for the interview template in the **Title** box (or accept the suggestion HotDocs makes). The title is what identifies the file in the library.
6. Optionally, enter a description in the **Description** box. Descriptions appear in the **Properties** tab of the library when the template is selected.
7. Click **OK**. The **Component Manager** window appears, with a single computation variable in it named **INTERVIEW**. (Your interview templates are not required to use this specifically named Computation variable. If you prefer, you can delete this component and create a new one. Simply specify the name of the new interview component at the **Interview** tab of **Component File Properties**.) The interview component must contain the script that asks the questions in the interview. Variables and dialogs used in the script will be used to create the interview.
8. Select the interview component and click the  **Edit Component** button. The **Computation Editor** appears.
9. Create the variables you want answered, link them to dialogs, and then create a script that asks the dialogs you want users to see when they assemble this template. (See [Put ASK Instructions in a Computation Variable](#).)

## To edit an existing interview template

1. At the HotDocs library window, select the interview template and click  **Edit Template**. The **Component Manager** window appears.

2. Select the interview component and click the  **Edit Component** button. The **Computation Editor** appears.
3. Make any necessary changes.

**Notes:**

- When converting interview templates from previous versions of HotDocs to HotDocs 2008 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. See [Define a Custom Interview](#) for details.
- 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** box as is). See [Create a Text Template Based on an Existing Template](#).

# Have HotDocs Generate a Default Interview

When you assemble a template, HotDocs displays an interview, based on variables and instructions it finds in the template. An *outline* of the interview is displayed in the left pane of the assembly window.

This interview is comprised of dialogs that contain one or more questions, which the user must answer to assemble a complete document. As users answer questions, the outline updates to show which questions are still unanswered.

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, if you need control over the order questions are asked in the interview, you can create a custom interview component. See [Define a Custom Interview](#) for details.)

## To generate a default interview for a template

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 **Interview** tab. The window changes to show interview options.
4. Select **Generate default interview**.

Now, whenever this template is assembled, the questions in the interview will be asked based on the order they are asked in the template.

If you want to control the appearance of just one or two dialogs in the interview, you can insert ASK instructions at the place in the template where you want your questions asked. See [Control When Your Dialogs Appear](#) for details. If you want to use a custom interview script, see [Define a Custom Interview](#).

# Define a Custom Interview

When you assemble a template, HotDocs displays an interview, based on variables and instructions it finds in the template. An *outline* of the interview is displayed in the left pane of the assembly window.

This interview is comprised of dialogs that contain one or more questions, which the user must answer to assemble a complete document. As users answer questions, the outline updates to show which questions are still unanswered.

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, 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 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.

One reason why it may be beneficial to create a custom interview is to speed up the process with which HotDocs displays and updates the interview. To explain, during assembly, the interview that is presented to the user is dynamic. This means that each time a user enters an answer, the entire interview is updated to reflect any changes caused by that answer. Depending on the complexity of the template and the frequency with which variables and instructions are used in the template, this updating may take longer than expected because HotDocs must process each field. By creating a custom interview, you can create a script that asks these components and processes these instructions just once, thereby reducing the number of times HotDocs has to process each field. This can improve assembly speed considerably. (For a detailed description of how a complex template can slow interview speed, see [Interviews and Complex Text Templates](#) as well as [Overview: HotDocs Interviews](#).)

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.

## Part 1: To create a computation that contains your scripting

1. At the template (which should be completely automated), open Component Manager. (See [Open and Close Component Manager](#).)
2. Select **Computation Variables** from the **Components** drop-down list and click the  **New Component** button. HotDocs opens the **Computation Editor**.
3. Enter a name in the **Variable name** box. (The component can use any name, including INTERVIEW.)
4. Using a series of instructions, specify how you want variables in the template to be asked, 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 are 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.

## Part 2: To specify that the interview component be used to generate the interview

1. With Component Manager still open, click The  **Component File Properties** button. The **Component File Properties** dialog box appears.
2. Click the **Interview** tab and select **Use custom interview**.
3. Specify the name of the Computation variable in the **Interview component** box.

Now, HotDocs will use this computation to generate the interview.

**Notes:**

- 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** box.
- You can test your custom interview computation to make sure each variable is asked correctly. To do this, clear **Use custom interview** at the **Component File Properties** dialog box, insert the interview component at the beginning of the template, and test assemble it. (See [Test Assemble a Text Document](#).) If HotDocs displays any dialogs after the expected final dialog, that means some variables were not included in the script and you must go back and modify the script. (Make sure you remove the computation from the template once you are finished testing, and remember to select **Use custom interview** again.)
- When you publish a template for use with HotDocs Server that has the **Use custom interview** property set, HotDocs uses the computation to build the interview definition (or .JS) file. (See [Publish Templates for Use with HotDocs Server](#).)

# Automatically Disable Irrelevant Variables in Interviews

When creating dialogs, 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. To accomplish this, you can either write a script (see [Use Scripts to Add Power to Your Dialogs](#)) 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 pre-existing 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.

When you start an assembly, HotDocs determines which variables are referred to—for example, referred to in instructions, merged in the document, and so forth—based on the current answers. If a variable is unused, for example, if you answer a conditional True/False variable such that it doesn't merge the associated Text variable, the Text variable will be grayed or hidden in the dialog. (A third option lets you always show the variable, even if it's unused.)

Finally, if a dialog contains all inactive variables, you can keep the dialog from being asked in the interview.

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 referred to 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 variables should be treated if they're not referred to in the document

1. Edit the variable. (See [Edit a Variable](#).)
2. At the **Advanced** tab of the Variable Editor, click the **When irrelevant** drop-down button and choose an option:
  - **Gray** causes the variable to appear grayed out if it is unused in the document. (This is the default option for all variables.)
  - **Hide** causes the variable to be hidden if the variable is unused in the document. If the answer file changes and the variable is needed, the variable will be shown in the dialog.
  - **Show** causes the variable to always appear in the dialog, regardless of whether it's used or not.

## To automatically exclude dialogs that contain only irrelevant variables

1. Edit the dialog. (See [Edit a Custom Dialog](#).)
2. Click the **Options** tab. The view changes to the different options for displaying a dialog.
3. Select **Hide this dialog when all variables are inactive**.

## To automatically disable unused variables in the interview

1. Open **Component Manager** for the template. (See [Open and Close 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 options.
4. Select **Automatically disable irrelevant variables and dialogs**.

**Note:** Model documents assembled from markup are set to automatically disable unused variables. For information on marking up model documents, see [Overview: Create a Model Document](#).

# Inserting Variables

## Overview: 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, Number, Date, 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.

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.

# 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**, **Number**, **Date**, **True/False**, **Multiple Choice**, **Computation**, and **Personal Information**.
3. Type a variable name in the **Variable** box. (If the variable already exists, select it from the drop-down 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. 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](#).)

**Warning: 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**, **Number**, **Date**, **True/False**, **Multiple Choice**, **Computation**, and **Personal Information**.
4. Type a name in the **Variable** box. (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.
7. When finished making changes, click **OK** at both the **Variable Editor** and **Variable Field** dialog boxes. HotDocs inserts the variable.

**Notes:**

- 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.

# 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

1. Select the template text and create the variable. (See [Insert a Variable in a HotDocs 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).
2. If you clicked **Replace Multiple**, the **Find and Replace** dialog box appears. The **Find what** box shows what text HotDocs is searching for. The **Replace with** box shows the name of the variable HotDocs is inserting.
3. Make a selection, based on the following information:

To	Do This
Replace this instance of text and move to the next instance	Click <b>Replace</b> .
Have HotDocs replace all instances of text from the selected instance to the end of the template (without confirmation)	Click <b>Replace All</b> .
Have HotDocs skip this instance and search for the next	Click <b>Find Next</b> .
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 <b>Find whole words only</b> .
Replace only those instances of text that have the same capitalization as the selected instance	Select <b>Match case</b> .
Change the properties of the variable	Click <b>Modify</b> 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**.
5. 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.)
3. 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.

# 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 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 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.

## To add a comment to a variable or instruction field

1. Insert a new variable or edit an existing variable. (See [Insert a Variable in a HotDocs 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** box.
4. 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» .

**Note:** 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 |».

# 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

1. In the text template, click in the variable field and click the  **Edit Component** button. Or, in a form template, select the field and click the  **Edit Component** button. The **Variable Editor** appears.
2. Make any changes.

## 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  **Edit Component** button. The **Variable Editor** appears.
3. Make any changes.

### Notes:

- 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.](#)

# Customize a Text Variable

Text variables 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.

## 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 HotDocs 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:

To	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 <b>Default format</b> drop-down list. (See <a href="#">Format the Variable</a> .)  Optionally, to keep the answer from breaking across lines in the document, select <b>Non-breaking</b> .
Specify an alternate name for the variable  Sometimes you use variable names that are useful for you during template development, but are not useful for users during an interview. For example, maybe you name your variables with some type of variable notation, like <i>Client name TE</i> . To a user, this name, if seen in an interview, may not make much sense. Here, a title (like <i>Client's Name</i> ) can be used in place of the variable name.	Enter a title in the <b>Title</b> box.  See <a href="#">Understand How Component Titles and Prompts Are Used</a> for a description of how and when variable titles are used.
Provide users with additional information about the variable	Enter a prompt in the <b>Prompt</b> box. This information replaces the <b>Variable name</b> and <b>Variable title</b> when the variable is presented during the interview. (See <a href="#">Create a Prompt for a Variable</a> .)
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 <b>Field height</b> box, or click the up or down arrows to select a number.  Changing the answer field height affects only

	<p>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 <b>Maximum characters</b> box.</p>
<p>Control how an answer is merged in a Word document when the user presses <b>Enter</b> in a multi-line answer field</p> <p>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 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 (↵).</p>	<p>Select <b>Enter key in multi-line answers inserts new paragraph mark (¶)</b>.</p> <p>To control whether line or paragraph breaks are inserted in literal text strings used in computation scripts, see <a href="#">Use Line Breaks and Tabs in Computation Scripts</a>.</p>
<p>Control the number of characters allowed in the answer</p>	<p>Either type a value in the <b>Maximum characters</b> box, or click the up or down arrows to select a value (up to 15,000 characters).</p>
<p>Control what type of information a user types and format it as they type</p>	<p>Select a pattern from the <b>Pattern</b> 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 <a href="#">Create a Pattern for a Text Variable</a>.)</p>
<p>Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document</p>	<p>Click the <b>Advanced</b> tab and select any of the options. (See <a href="#">Control How HotDocs Processes a Variable</a>, <a href="#">Specify the Width of Answer Fields in the Interview</a>, and <a href="#">Control How Answers Appear in the Assembled Document</a>, respectively.)</p>
<p>Provide users with helpful information that can assist them in providing the correct answer</p>	<p>Click the <b>Resource</b> tab and provide a resource option. (See <a href="#">Add Resource Information to a Variable or Dialog</a>.)</p>
<p>View a list of all components that refer to this variable</p>	<p>Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components</a>.)</p>

Once a variable is created, you can edit it at any time. See [Edit a Variable](#).

**Note:** To save your changes to a variable without closing the variable editor, click **Save**.

# 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.

## 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 HotDocs Template](#).)
2. At the **Variable Field** dialog box, click the  **Edit Component** button to open the **Number Variable Editor**.
3. Make any changes, as described in the following table:

To	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 <b>Default format</b> drop-down list. (See <a href="#">Format the Variable as well as Specify Number Variable Example Formats and Fractions</a> .)  Optionally, to keep the answer from breaking across lines in the document, select <b>Non-breaking</b> .
Specify an alternate name for the variable  Sometimes you use variable names that are useful for you during template development, but not are useful for users during an interview. For example, maybe you name your variables with some type of variable notation, like <i>Client Monthly Salary NU</i> . To a user, this name, if seen in an interview, may not make much sense. Here, a title (like <i>Client's Monthly Salary</i> ) can be used in place of the variable name.	Enter a title in the <b>Title</b> box.  See <a href="#">Understand How Component Titles and Prompts Are Used</a> for a description of how and when variable titles are used.
Provide users with additional information about the variable	Enter a prompt in the <b>Prompt</b> box. This information replaces the <b>Variable name</b> and <b>Variable title</b> when the variable is presented during the interview. (See <a href="#">Create a Prompt for a Variable</a> .)

Force users to enter a number that falls within a certain numeric range	Type numbers in the <b>Minimum</b> and <b>Maximum</b> boxes.
Specify a certain number of decimal places allowed in the users' answer	Either type a number (0-7) in the <b>Decimal places</b> box, 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 <b>Currency</b> 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 <b>Advanced</b> tab and select any of the options. (See <a href="#">Control How HotDocs Processes a Variable</a> , <a href="#">Specify the Width of Answer Fields in the Interview</a> , and <a href="#">Control How Answers Appear in the Assembled Document</a> , respectively.)
Provide users with helpful information that can assist them in providing the correct answer	Click the <b>Resource</b> tab and provide a resource option. (See <a href="#">Add Resource Information to a Variable or Dialog</a> .)
View a list of all components that refer to this variable	Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components</a> .)

Once a variable is created, you can edit it at any time. (See [Edit a Variable](#).)

**Notes:**

- 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](#).
- To save your changes to a variable without closing the variable editor, click **Save**.

# 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.

## 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 HotDocs Template](#).)
2. At the **Variable Field** dialog box, click the  **Edit Component** button to open the **Date Variable Editor**.
3. Make any changes, as described in the following table:

To	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 <b>Default format</b> drop-down list. (See <a href="#">Format the Variable</a> .)  Optionally, to keep the answer from breaking across lines in the document, select <b>Non-breaking</b> .
Specify an alternate name for the variable  Sometimes you use variable names that are useful for you during template development, but are not useful for users during an interview. For example, maybe you name your variables with some type of variable notation, like <i>Client Birth Date DA</i> . To a user, this name, if seen in an interview, may not make much sense. Here, a title (like <i>Client's Birth Date</i> ) can be used in place of the variable name.	Enter a title in the <b>Title</b> box.  See <a href="#">Understand How Component Titles and Prompts Are Used</a> for a description of how and when variable titles are used.
Provide users with additional information about the variable	Type a prompt in the <b>Prompt</b> box. This information replaces the <b>Variable name</b> when the variable is presented during the interview. (See <a href="#">Create a Prompt for a Variable</a> .)
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 <b>Advanced</b> tab and select any of the options. (See <a href="#">Control How HotDocs Processes a Variable</a> , <a href="#">Specify the Width of Answer Fields in the Interview</a> , and <a href="#">Control How Answers Appear in the Assembled Document</a> , respectively.)

Provide users with helpful information that can assist them in providing the correct answer	Click the <b>Resource</b> tab and provide a resource option. (See <a href="#">Add Resource Information to a Variable or Dialog</a> .)
View a list of all components that refer to this variable	Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components</a> .)

Once a variable is created, you can edit it at any time. See [Edit a Variable](#).

**Note:** To save your changes to a variable without closing the variable editor, click **Save**.

# 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.

## 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 HotDocs Template](#) as well as [Tips on Naming True/False Variables](#)).
2. At the **Variable Field** dialog box, click the  **Edit Component** button to open the True/False Variable Editor.
3. Make any changes, as described in the following table:

To	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)	<p>Select a format from the <b>Default format</b> drop-down list.</p> <p>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 <a href="#">Format the Variable</a>.)</p> <p>Optionally, to keep the answer from breaking across lines in the document, select <b>Non-breaking</b>.</p> <p><b>Note:</b> 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.</p>
Specify an alternate name for the variable	Enter a title in the <b>Title</b> box.
Sometimes you use variable names that are useful for you during template development, but are not useful for users during an interview. For example, maybe you name your variables with some type of variable notation, like <i>Client Is Married TF</i> . To a user, this name, if seen in an interview, may not make much sense. Here, a title (like <i>Is Client Married?</i> ) can be used in place of the variable	See <a href="#">Understand How Component Titles and Prompts Are Used</a> for a description of how and when variable titles are used.

name.	
Provide users with additional information about the variable	Enter a prompt in the <b>Prompt</b> box. This information replaces the <b>Variable name</b> or <b>Variable title</b> when the variable is presented during the interview. (See <a href="#">Create a Prompt for a Variable</a> .)
Have <i>Yes</i> and <i>No</i> appear on the same line in the interview	Select <b>Yes/No on same line</b> .
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 <b>Advanced</b> tab and select any of the options. (See <a href="#">Control How HotDocs Processes a Variable</a> , <a href="#">Specify the Width of Answer Fields in the Interview</a> , and <a href="#">Control How Answers Appear in the Assembled Document</a> , respectively.)
Provide users with helpful information that can assist them in providing the correct answer	Click the <b>Resource</b> tab and provide a resource option. (See <a href="#">Add Resource Information to a Variable or Dialog</a> .)
View a list of all components that refer to this variable	Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components</a> .)

Once a variable is created, you can edit it at any time. See [Edit a Variable](#).

**Notes:**

- 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. 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 [Condition Text Using True/False Variables](#).
- To save your changes to a variable without closing the variable editor, click **Save**.

# Customize a Multiple Choice Variable

You can let users select one or more answers from a list of possible options by using a Multiple Choice variable. 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. See [Specify Merge Text Options as Default or Field-Specific Properties](#).)

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 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.

## To edit the properties of a Multiple Choice variable

1. Follow the instructions for inserting a variable, selecting **Multiple Choice** as the variable type. (See [Insert a Variable Field in a HotDocs Template](#).)
2. At the **Variable Field** dialog box, click the  **Edit Component** button to open the **Multiple Choice Variable Editor**.
3. Type the options (at least two) you want users to choose from in the **Option** column. (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. See [Work with Multiple Choice Options](#).)
5. 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. See [Work with Multiple Choice Options](#).)
6. From the **Select** drop-down list, select either **One Only** (presents the options so that only one may be selected), or **All That Apply** (presents the options so that all that are applicable may be selected).
7. Depending on your selection in Step 6, you can select **Other** (allows users to type an answer not in the list), or **None of the Above** (lets the user 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 other changes, as described in the following table:

To	Do This
Add longer prompts and merge text for each option, as well as suggest an answer for the user	Click the <b>Options</b> tab and make your changes. (See <a href="#">Work with Multiple Choice Options</a> .)
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 <b>Default format</b> drop-down list. (See <a href="#">Format the Variable</a> .)  The example formats that are available depend on whether you choose <b>One Only</b> or

	<p><b>All That Apply.</b></p> <p>Optionally, to keep the answer(s) from breaking across lines in the document, select <b>Non-breaking</b>.</p>
<p>Specify an alternate name for the variable</p> <p>Sometimes you use variable names that are useful for you during template development, but are not useful for users during an interview. For example, maybe you name your variables with some type of variable notation, like <i>Client Gender MC</i>. To a user, this name, if seen in an interview, may not make much sense. Here, a title (like <i>Client's Gender</i>) can be used in place of the variable name.</p>	<p>Enter a title in the <b>Title</b> box.</p> <p>See <a href="#">Understand How Component Titles and Prompts Are Used</a> for a description of how and when variable titles are used.</p>
<p>Provide users with additional information about the variable</p>	<p>Type a prompt in the <b>Prompt</b> box. This information replaces the <b>Variable name</b> when the variable is presented during the interview. (See <a href="#">Create a Prompt for a Variable</a>.)</p>
<p>Control how the variable is processed during assembly, as well as how its answer appears in the interview and in the assembled document</p>	<p>Click the <b>Advanced</b> tab and select any of the options. (See <a href="#">Control How HotDocs Processes a Variable</a>, <a href="#">Specify the Width of Answer Fields in the Interview</a>, and <a href="#">Control How Answers Appear in the Assembled Document</a>, respectively.)</p>
<p>Provide users with helpful information that can assist them in providing the correct answer</p>	<p>Click the <b>Resource</b> tab and provide a resource option. (See <a href="#">Add Resource Information to a Variable or Dialog</a>.)</p>
<p>View a list of all components that refer to this variable</p>	<p>Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components</a>.)</p>

Once a variable is created, you can edit it at any time. See [Edit a Variable](#) for details.

**Notes:**

- 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. See [Specify Merge Text Options as Default or Field-Specific Properties](#) for details.
- You can 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. See [Copy and Paste Columns in a Multiple Choice Variable Spreadsheet](#).
- 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.)
- 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. See [Use Variables and Scripts in Prompts, Dialog Element Text, and Plain Text Resources](#) for details.
- To save your changes to a variable without closing the variable editor, click **Save**.

# 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** box and allow HotDocs to guide you in filling in the required information. With experience, you can type the script directly in the box. (See [Understand the HotDocs Scripting Language](#) and [Use the Script Editor](#).)

## To edit the properties of a Computation variable

1. Follow the instructions for inserting a variable, selecting **Computation** as the variable type. (See [Insert a Variable Field in a HotDocs Template](#).)
2. At the **Variable Field** dialog box, click the  **Edit Component** button to open the **Computation Editor**.
3. In the **Script** box, enter a computation script. (See [Use the Script Editor](#) for specific instructions.)
4. Make any other changes, as described in the following table:

To	Do This
Indent matching pairs of IF and REPEAT instructions	Click the  <b>Auto Format</b> 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)  It is best to wait until you are finished writing the script before you assign a format, since the list of available formats (text-based or number-based) may change, depending on the type of value that is returned by the computation.	Select a format from the <b>Default format</b> drop-down list. (See <a href="#">Format the Variable</a> .)  Optionally, to keep the answer from breaking across lines in the document, select <b>Non-breaking</b> .
Change the properties of default merge fields	Click the <b>Advanced</b> tab and select any of the options. (See <a href="#">Control How Answers Appear in the Assembled Document</a> .)
Add a title to the Computation variable  Sometimes you use variable names that are useful for you during template development, but are not useful for those viewing a reviewable version of the template. For example, maybe you name your variables	Click the <b>Advanced</b> tab and enter the title in the <b>Title</b> box.

<p>with some type of variable notation, like TaxCalc1 CO. To a reviewer, this name, if seen, may not make much sense. Here, a title (like Federal Tax Calculation) can be use in place of a variable name.</p>	
<p>Provide users with helpful information that can assist them when directly filling a form document</p>	<p>Click the <b>Resource</b> tab and provide a resource option. (See <a href="#">Add Resource Information to a Variable or Dialog.</a>)</p> <p>Since Computation variables generally use other variables to produce an answer or value, resources assigned to them will not appearing during the interview. The only exception is when users directly fill a form document and they select the field to which the Computation variable is attached.</p>
<p>View a list of all components that refer to this variable</p>	<p>Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components.</a>)</p>

Once a variable is created, you can edit it at any time. See [Edit a Variable](#) for details.

**Notes:**

- 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. See [Resolve Syntax Errors in a Template or Script](#) for troubleshooting tips. 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. See [Overview: Debugging Templates](#) for information.
- To save your changes to a variable without closing the variable editor, click **Save**.

# 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** box, 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 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 [Enter or Edit Your Personal Information](#).

# 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 HotDocs 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.

**Warning:** 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 in a HotDocs Template](#).)
3. With the **Variable Field** dialog box displayed, type **PN#** in the **Variable** box. (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.

**Note:** 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.

# 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.

## To determine the number of entries in a list

1. Create a Computation variable containing the expression **COUNT( DIALOG )**. (See [Customize a Computation Variable](#).)
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 simply 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.)

## 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. (See [Insert a Variable Field in a HotDocs Template](#) and [Customize a Number Variable](#).) Instead of typing a variable name, however, click the **Variable** drop-down 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»

# 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 HotDocs 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

1. 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**.
6. 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  **Edit 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** box, 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.

### Notes:

- 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** box. 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 allows 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** box, 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 [Overview: Dot Codes](#).)
4. If the prompt still does not convey all the information you want the user to know, you can specify a resource. See [Add Resource Information for a Variable or Dialog](#).

### Notes:

- You can include a variable in the **Prompt** box, which will merge the user's answer into the prompt during the interview. See [Use Variables and Scripts in Prompts, Dialog Element Text, and Plain Text 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** box.
- To make a character in your prompt an accelerator key, type an ampersand (&) character immediately before the letter you want as an accelerator. Then, during the interview, users can press the **Alt** key while also pressing the underlined character in the prompt, which moves the cursor to that answer field.

# 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.

When **Ask automatically** is selected, 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. By clearing **Ask automatically**, you cause a variable to be asked only when it is in a dialog that is specifically asked (or one asked using an ASK instruction). 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 [Control Whether Dialogs are Asked Automatically](#).)

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.

## 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:

To	Do This
Prevent the variable question from appearing in an interview unless the variable is used in a dialog that is specifically asked	Clear <b>Ask automatically</b> .
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 <b>Warn when unanswered</b> .
Prevent the variable's answer from being saved in an answer file	Clear <b>Save in answer file</b> .

**Note:** These options are not available for Computation variables.

# 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.

### Notes:

- 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** box. 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:

To	Do This
Specify the minimum number of character spaces you want the merge field to use	Type a number in the <b>Field width</b> box, 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 column of dollar amounts, you can right-align the numbers.)	Choose <b>Right</b> or <b>Left</b> from the <b>Align answer</b> drop-down list.  When aligning <i>Number</i> 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 <b>Fill character</b> box.
Merge specific text when the variable is	Type the text in the <b>Unanswered text</b> box.

unanswered	(See <a href="#">Tips on Using Unanswered Variable Placeholders.</a> )
Format answers using symbolic character fonts, such as bar code fonts	<p>Click the <b>Font</b> drop-down button and choose a font from the list.</p> <p>This is a field-specific property available only in text templates. Additionally, you can assign this property only at the <b>Variable Field</b> dialog box.</p> <p><b>Warning:</b> Make sure your users have the same fonts installed on their computers.</p>

**Notes:**

- 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.

# Automatically Disable Irrelevant Variables in Interviews

When creating dialogs, 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. To accomplish this, you can either write a script (see [Use Scripts to Add Power to Your Dialogs](#)) 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 pre-existing 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.

When you start an assembly, HotDocs determines which variables are referred to—for example, referred to in instructions, merged in the document, and so forth—based on the current answers. If a variable is unused, for example, if you answer a conditional True/False variable such that it doesn't merge the associated Text variable, the Text variable will be grayed or hidden in the dialog. (A third option lets you always show the variable, even if it's unused.)

Finally, if a dialog contains all inactive variables, you can keep the dialog from being asked in the interview.

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 referred to 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 variables should be treated if they're not referred to in the document

1. Edit the variable. (See [Edit a Variable](#).)
2. At the **Advanced** tab of the Variable Editor, click the **When irrelevant** drop-down button and choose an option:
  - **Gray** causes the variable to appear grayed out if it is unused in the document. (This is the default option for all variables.)
  - **Hide** causes the variable to be hidden if the variable is unused in the document. If the answer file changes and the variable is needed, the variable will be shown in the dialog.
  - **Show** causes the variable to always appear in the dialog, regardless of whether it's used or not.

## To automatically exclude dialogs that contain only irrelevant variables

1. Edit the dialog. (See [Edit a Custom Dialog](#).)
2. Click the **Options** tab. The view changes to the different options for displaying a dialog.
3. Select **Hide this dialog when all variables are inactive**.

## To automatically disable unused variables in the interview

1. Open **Component Manager** for the template. (See [Open and Close 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 options.
4. Select **Automatically disable irrelevant variables and dialogs**.

**Note:** Model documents assembled from markup are set to automatically disable unused variables. For information on marking up model documents, see [Overview: Create a Model Document](#).

# 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 formats or programs to create and display resources: WinHelp, HTML Help, Folio infobase, a custom program, and a URL.

## To assign plain text as a resource

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** box.
5. Optionally, to assign font properties to the prompt text (such as bold or underline), enter a formatting dot code. (See [Overview: Dot Codes](#).)
6. 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 Plain Text Resources](#).)
7. 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.)

**Warning:** You cannot use angle brackets with text between them in a plain-text resource (for example, <Your Name>). Instead, you must use the codes **&lt;** and **&gt;**; to insert the brackets (for example, **&lt;Your Name&gt;**). (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 >**.)

## To assign HTML Help as a resource

This format uses text from an HTML Help (.CHM) file. The information can appear either in a help viewer or as pop-up text. You may need to review your HTML Help program's documentation for additional information on concepts and terms described below.

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 **HTML Help** from the **Resource type** drop-down list.
4. Click the  **Open** button next to the **HTML Help file** box to locate the file.
5. In the **Topic** box, type a topic name (for example, *Employee\_Name.htm*) or an alias map number (for example, *1*).

6. Select **Resource pane** to display the HTML text directly in the Resource pane.
7. Optionally, if you have a specific window style you want your help to appear in, include the window name after a right-angle bracket at the end of the topic name, for example, *topic\_title.htm>secondary window*. (To reference a bookmark, include the pound sign between the topic title and the bookmark name—for example, *topic\_title.htm#bookmark\_name*.)

### To display HTML Help text as a pop-up window

1. Select **HTML Help** from the **Resource type** drop-down menu.
2. Click the  **Open** button next to the **HTML Help file** box to locate the file.
3. In the **Topic** field, enter a context ID map number for the pop-up help. If your context-sensitive help is saved as something other than *popup.txt* (the default file name for which HotDocs searches), you must include the file name after the map number (for example, *1.context.txt*).
4. Select **Pop-up**.
5. When the user views the resource pane, HotDocs displays a link which the user must click. At that point, What's This? help-style window appears.

### To assign Windows Help as a resource

**Warning:** Windows Help (or HLP files) are not supported on Windows Vista.

1. This format uses text from a Windows Help (.HLP) file. The file must be in the same folder as the template.
2. Open the variable or dialog to which you want to assign resources. (See [Edit a Variable](#) or [Edit a Custom Dialog](#).)
3. Click the **Resource** tab.
4. Click the **Resource type** drop-down button and select **Windows Help** from the drop-down list.
5. Click the  **Open** button next to the **Windows Help file** box and locate the file.
6. In the **Topic** field, type the context ID number of the corresponding help topic. (Sometimes this is called a hash number. You may need to review your Windows Help program's documentation for additional information.)
7. Optionally, select **Pop-up** to display the information in a What's This? help-style window. (The window only stays open until the user clicks somewhere else.)

During assembly, HotDocs displays a link in the resource pane, which the user can click to open the WinHelp viewer. The information from the help file appears.

### To assign Folio Help as a resource

1. This format uses text from a Folio infobase, which appears in Folio Views. The Folio infobase must be in the same folder as the template.
2. Open the variable or dialog to which you want to assign resources. (See [Edit a Variable](#) or [Edit a Custom Dialog](#).)
3. Click the **Resource** tab.
4. Click the **Resource type** drop-down button and select **Folio Infobase** from the drop-down list.
5. Click the  **Open** button next to **Infobase file** box and locate the infobase (.NFO) file.
6. In the **Jump link** box, type the name of the jump link.

During assembly, HotDocs displays a link in the resource pane, which the user can click to open Folio

Views. The information from the infobase appears.

See [Additional Support Information for Folio Views 4.2](#) for more information on using Folio help.

### To assign a custom program as a resource

This format uses text from custom programs, which displays the resource according to the program's design. Make sure the custom program is in either the same folder as the template or the Windows\System folder.

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 **Custom Program** from the drop-down list.
4. Click the  **Open** button next to the **Program file** box and locate the file.
5. Type any necessary command-line options in the **Command-line parameters** box.
6. During assembly, HotDocs displays a link in the resource pane, which the user can click to open the custom program.

### To assign a URL as a resource

This format uses text from a Web page, displayed in a Web browser.

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 **URL** from the drop-down list.
4. Type the full URL path in the **URL** field.
5. Optionally, type the anchor tag of the specific text in the Web page you want to display in the **Anchor tag** box.
6. Optionally, select **Resource pane** to display the Web page directly in the resource pane.

During assembly, HotDocs displays the Web page either in its own browser window, or in the resource pane, depending on whether you selected **Resource pane**.

#### Notes:

- Resources assigned to a Computation variable can only be accessed by users if the variable is attached to a form field and the user is directly filling the form. (This may be useful if you want to explain how the field is computed or tell users why they may want to override a computed value.) If you want to provide a resource for a computation in a text template, assign it to the variables that are used in the computation.
- 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.

# 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 this 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.
3. 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**.

### Notes:

- 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.

# 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 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 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.

## To add a comment to a variable or instruction field

1. Insert a new variable or edit an existing variable. (See [Insert a Variable in a HotDocs 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** box.
4. 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» .

**Note:** 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 |».

## Create a Pattern for a Text Variable

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** box (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
X	any character
9	any numeric character
A	any alphabetic character
U	uppercase alphabetic character
L	lowercase alphabetic character
N	alphanumeric character (9 and A)
H	hexadecimal character (0-9, A-F, a-f)

**Note:** 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.)

# Specify 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** box 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

Merge 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** box 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	2.5	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 require more space, you can 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. (See [Customize a Multiple Choice Variable](#).)
2. After specifying all 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. Perform any of the following optional tasks:
  - Type a new prompt in the **Prompt** box or edit the existing prompt. (500-character limit)
  - Type a new merge text option in the **Default merge text** box 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.

**Warning:** Do not use the **Automatically select this option if variable is unanswered when displayed** if you are using the Multiple Choice variable in a repeated dialog. Because of the way HotDocs processes dialogs, whenever you default a 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, you should use a computation script that conditions the default value. See [DEFAULT VAR TO VALUE](#) for more details.

### Notes:

- 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 Multiple Choice Variable](#) or [Edit a 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 default**. HotDocs ungrays the **Merge Text** column.
3. Enter the merge text that corresponds with the **Options** you have specified.

### Notes:

- 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.

# 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.)

**Note:** 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** box.
5. Click **Rename**. The component is renamed.

**Note:** To edit an existing merge text group, see [Edit Formats, Merge Text, Dialog Elements, and Patterns.](#)

# 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.

## Warnings:

- 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 2008 automatically assembles these, so the option has been removed from the **Component File Properties** dialog box.

**Note:** If a Word RTF 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 Variables and Scripts in Prompts, Dialog Element Text, and Plain Text 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 [Edit a Custom Dialog](#).)
2. Place your cursor where you want the variable reference (for example, a prompt, title, or dialog element text box), right-click, and choose **Variable Field**. The **Variable Field** dialog box appears.

**Note:** 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.
4. Optionally, to enter a script in a plain-text resource, right-click in the **Text** box and choose **IF Field** or **REPEAT Field** from the shortcut menu, depending on the type of instruction you are inserting.

### Notes:

- 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.

# 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** box.
3. Drag a variable you want to receive the new answer from the **Variables** list onto the **VAR** placeholder.
4. Replace the **EXPRESSION** 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** box, 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.

### Notes:

- 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"*).

# 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" (Tip: In this instance, the Multiple Choice variable must have the Select All That Apply property set.)
```

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
```

# 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.

**Warning:** 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 be used as a keyword, which will cause HotDocs to read your variable name incorrectly. (See [Overview: Instruction and Expression Models](#).)

Finally, there are certain characters you cannot use when naming your components. They are:

. (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)

## 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 the Answer Will Be Formatted in the 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 the Answer Will Be Formatted in the Document
09	4 78	04 78
9	7,898	7,898
9 1/8 (variable must have decimals specified)	2.3	2 ¼
9,999.00	9214 9214.36	9,214.00 9,214.36

9,999.99999	87.5984	87.5984
9.9	87.6	87.6
9.9.a (this format is best used with PN variables)	287.3	287.3.a
9999	12587	12587
9th	23	23 <sup>rd</sup>
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 Alpha ALPHA	12	twelve Twelve TWELVE
ordinal Ordinal ORDINAL	25	twenty-fifth Twenty-Fifth TWENTY-FIFTH

**Warning:** 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 the Answer Will be Formatted in the Document
03 JUN 90	November 6, 2000	06 NOV 00
06/03/90	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 6 <sup>th</sup>
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

You can also create your own set of date variables using a different set of format codes:

Property Name	Description	Example Usage	Example Formatted Answer
d	Numeric day	d	1

m	Numeric month	m	1
y	Numeric year (four digits)	y	2007
dd	Two-digit numeric day	dd	01
mm	Two-digit numeric month	mm	01
yy	Two-digit numeric year	yy	07
yyyy	Four-digit numeric year	yyyy	2007
dth	Case-sensitive numeric ordinal day	dth dTH	1st 1 <sup>ST</sup>
dy	Case-sensitive, spelled-out day	dy Dy DY	first First FIRST
mn	Case-sensitive, spelled-out month	mn Mn MN	january January JANUARY
yr	Case-sensitive, spelled-out year	yr Yr YR	two thousand seven Two Thousand Seven TWO THOUSAND SEVEN
wd	Case-sensitive, spelled-out weekday	wd Wd WD	monday Monday MONDAY
mnt	Case-sensitive month abbreviation	mnt	jan

		Mnt MNT	Jan JAN
wdy	Case-sensitive weekday abbreviation	wdy Wdy WDY	mon Mon MON

### Examples

Following are some examples of how to use these types of date formats:

How It's Used	How it Formats Answer
d/m/yy	1/1/07
dd/mm/yy	01/01/07
dth day of Mnt	1st day of Jan
dy day of the month	first day of the month
d/m/yy	1/1/07
dd/mm/yy	01/01/07
Mn, yyyy	January, 2007
Mnt yyyy	Jan 2007
d/m/yyyy	1/1/2007
yr	two thousand seven
Wdy, dd/mm/yy	Mon, 01/01/07

Wd, Mnt dth	Monday, Jan 1 <sup>st</sup>
-------------	-----------------------------

## True/False Variables

Example Format	User's Answer	How the Answer Will Be Formatted in the Document
/x	Yes No	(nothing) X
true/false	Yes No	true false
x/	Yes No	X (nothing)
yes/no	Yes No	yes no

## Multiple Choice Variables

Example Format	User's Answer	How the Answer Will Be Formatted in the Document
<b>Select One Only</b>		
like this	Option One	option one
Like this	Option One	Option one
Like This	Option One	Option One
LIKE THIS	Option One	OPTION ONE

<b>Select All That Apply</b>		
a, and b	Options 1 and 2	Option 1, and Option 2
a, b	Options 1, 2, and 3	Option 1, Option 2, Option 3
a, b and c	Options 1 and 2	Option 1 and Option 2
A, b and c	Options 1, 2, and 3	Option 1, Option 2 and Option 3
A, B and C	Options 1, 2, and 3	OPTION 1, OPTION 2 and OPTION 3
A, B AND C	Options 1, 2, and 3	OPTION 1, OPTION 2 AND OPTION 3
A, b or c	Options 1, 2, and 3	Option 1, Option 2 or Option 3
a, b, and c	Options 1, 2, and 3	Option 1, Option 2, and Option 3
a; b; and c	Options 1, 2, and 3	Option 1; Option 2; and Option 3

## 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.

## REPEAT Dialogs

<b>Example Format</b>	<b>User's Answer</b>	<b>How the Answer Will Be Formatted in the Document</b>
a, and b	apples and oranges	apples, and oranges

a, b	apples, oranges, and cherries	apples, oranges, cherries
a, b and c	apples, oranges, and cherries	apples, oranges and cherries
A, b and c	apples, oranges, and cherries	apples, oranges and cherries
A, B and C	apples, oranges, and cherries	apples, oranges and cherries
A, B AND C	apples, oranges, and cherries	apples, oranges AND cherries
A, b or c	apples, oranges, and cherries	apples, oranges or cherries
a, b, and c	apples, oranges, and cherries	apples, oranges, and cherries
a; b; and c	apples, oranges, and cherries	apples; oranges; and cherries

# 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 It's Used During the Interview	Order in Which Properties Are Used
Variable (used or not used in a custom dialog)	Answer field in the dialog pane This includes answer fields in pop-up interviews.	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	Title / Prompt / Variable name
	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	Dialog title in interview outline	Title / Prompt / Clause name

(referenced using an IF instruction in a template; is not used in a custom dialog)	Dialog title in the dialog pane title bar	Title / Prompt / Clause name
	Answer field in the dialog pane	Prompt / Title / Clause name
Database component	Dialog title in interview outline	Title / Component name
	Dialog title in dialog pane title bar	Title / Component name
	Table in dialog pane	Prompt / Title / Component name
Dialog (nonrepeated)	Dialog title in interview outline	Dialog title / Dialog name
	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 title in dialog pane title bar	Dialog title / Dialog name
Dialog (repeated as spreadsheet)	Dialog title in interview outline	Dialog title / Dialog name
	Dialog title in dialog pane title bar	Dialog title / Dialog name

**Notes:**

- 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 [Overview: Test HotDocs Templates](#).

# Overview: 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 do this, you insert dot code fields in the your prompts, text, or scripts. During assembly, these dot codes will be processed and will apply whatever formatting you specify to the text with which you are working.

## 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 used for the list. This is useful when you have separate answers that are being merged conditionally in list format.

## How Dot Codes Work

When you view the document during assembly, HotDocs actually processes the assembled document twice. The first time, the plain text (including the dot codes) is merged in the document. Then, the second time the document is assembled, the codes are processed and the formatting is applied or the characters are inserted.

A similar process is used to process dot codes used to format prompts, dialog element text, and resources in the interview.

# 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 to text

1. Either edit a Computation variable (see [Customize a Computation Variable](#)), or insert your cursor in the template text where you want the punctuation mark to appear.
2. 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):

Dot Code	Name	What It Does
«.»	Period	Inserts a period 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.
«:»	Colon	Inserts a colon if no other punctuation precedes it.
«!»	Exclamation point	Inserts an exclamation point if no other punctuation precedes it.
«?»	Question mark	Inserts a question mark if no other punctuation precedes it.

## To capitalize text

1. Either edit a Computation variable (see [Customize a Computation Variable](#)), or insert your cursor in the template text where you want the capitalization to take effect. (If you want to capitalize a section of text, highlight the text.)
2. 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	Name	What It Does
«.a» «.ae»	All Caps	Capitalizes every letter in the selected text.
«.c»	Capitalize Letter	Capitalizes the word immediately following the code.
«.l» «.le»	Leading Caps	Capitalizes the first letter of each word in the selected text.
«.s» «.se»	Small Caps	Capitalizes each letter in the selected text and displays the text at a smaller font size.

**Note:** 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.

# Change Font Properties of Text

**Warning:** 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.

## 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.
3. Choose **Character Format > Format** (where *Format* represents one of the dot codes described in the following table):

Dot Code	Name	What It Does
«.b» «.be»	Bold	Applies bold formatting to a word or group of words.
«.h» «.he»	Hidden	Applies the hidden text property to a word or group of words.
«.i» «.ie»	Italics	Applies italic formatting to a word or group of words.
«.u» «.ue»	Underline	Applies underline formatting to a word or group of words.
«.x» «.xe»	Strike-through	Crosses out a word or group of words.

<p>«.z +size» «.z -size»  «.ze»</p> <p><b>Warning:</b> The font size options are not supported in WordPerfect templates.</p>	<p>Font size</p>	<p>Changes the font size for a word or group of words: «.z +Size» is a percent increase in the current size. The dot code, «.z -Size» is a percent decrease in the current size.</p>
--	------------------	--

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».
```

**Note:** 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):

Dot Code	Name	What It Does
«.an»	An	Inserts “a” or “an”, 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» «.ra»	Left Apostrophe ( <code>'</code> ) Right Apostrophe ( <code>'</code> )	Inserts a left or right apostrophe.
«.lq» «.rq»	Left Quote ( <code>"</code> ) Right Quote ( <code>"</code> )	Inserts a left or right quotation mark.
«.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.

**Note:** 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](#).)
2. 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 punctuation dot codes, manually type the codes as you see them in the table below.)

Dot Code	Name	What It Does
«.p "style"»	Punctuation Format	Identifies the beginning of a punctuated list and assigns the punctuation format, based on this list:  a, and b a, b a, b and c A, b and c A, B, and C A, B, AND C a, b or c a, b, and c a; b; and c  See <a href="#">How Example Formats are Interpreted</a> for a more detailed explanation of list formats.
«.p»	Punctuation	Identifies the spot where punctuation characters should be inserted.
«.pe»	End Punctuation	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».
```

# Grouping Variables in Dialogs

## Overview: Create Your Own Dialogs

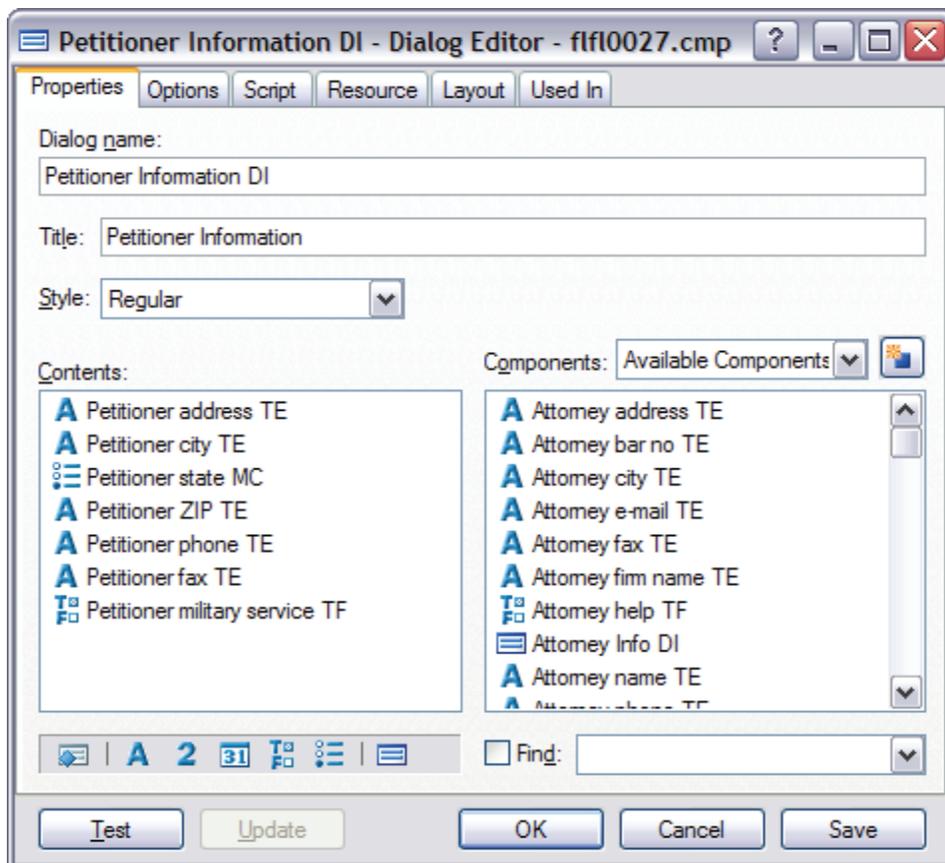
When you insert variables in your templates, HotDocs automatically creates a *default dialog* (or a question window) for each variable. When the user assembles the document, HotDocs presents each of these individual dialogs in the order the variables are used in the template.

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.

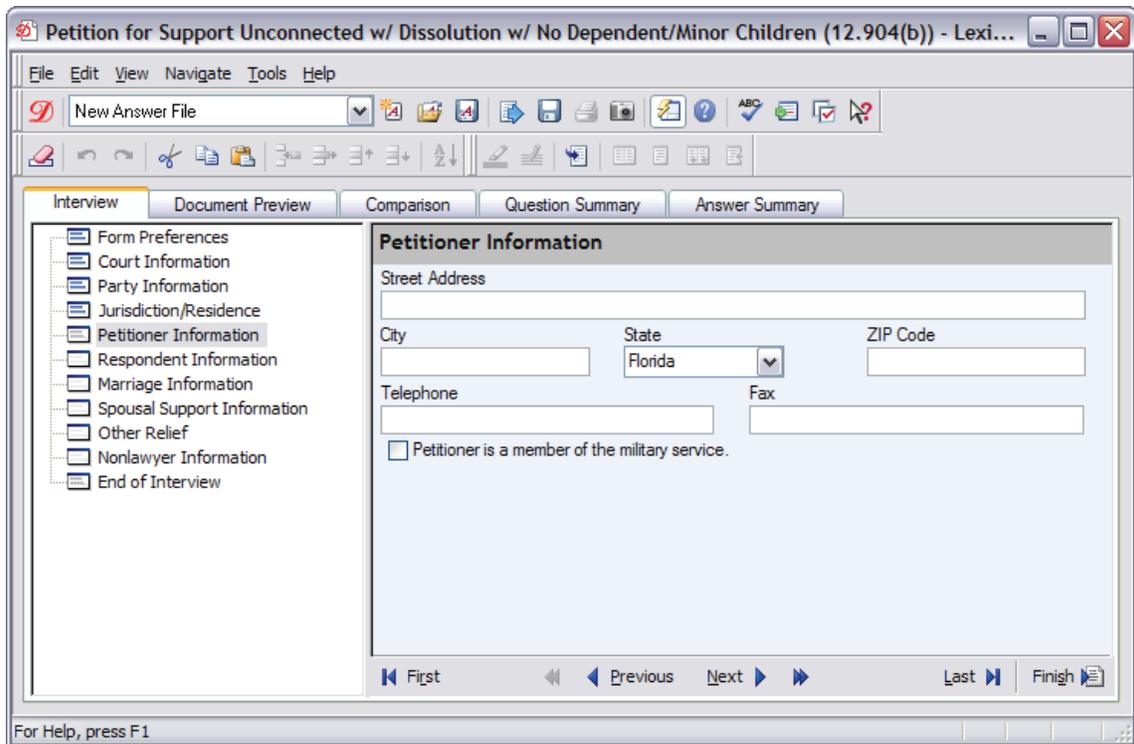
When you add a variable to a dialog, HotDocs creates a link between the variable and the dialog. When an answer is needed for the variable during the interview, HotDocs displays the linked dialog instead of displaying the variable by itself.

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.



During the interview, HotDocs displays the dialog and allows the user to enter answers.



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.

# Gather Questions into a Custom Dialog

By default, when HotDocs displays variables in a HotDocs interview, each variable is shown in a dialog by itself. This dialog is called a *default* dialog. While this approach may be sufficient for some, you may want to group related variables and present them in a *custom* dialog. Grouping questions together makes it easier for users to answer questions in the interview 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** box.
5. Select the components you want from the **Components** list and drag them to the **Contents** box.

**Note:** Click the **Components** drop-down button and select a component type to limit the list of components you are viewing. To 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:

To	Do This
Specify a prompt for the dialog that is different from the dialog name	Type the desired prompt in the <b>Title</b> box. 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 <i>name</i> 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. (See <a href="#">Change Font Properties of Text</a> .)
Add other items to a dialog, such as text, graphics, links, or design elements	Click on the <b>Create New Dialog Element</b> icon and drag it to the <b>Contents</b> list. The <b>Dialog Element Editor</b> 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.

<p>Choose how you want to display a dialog if it's repeated</p>	<p>Click the <b>Style</b> drop-down button and choose a repeated display option: <b>Repeated Series</b> (repeats the same dialog, one dialog at a time), or <b>Spreadsheet</b> (displays the dialog in spreadsheet format, allowing the user to type all their answers in one dialog). (See <a href="#">Create a REPEAT Instruction to Gather a List of Answers.</a>)</p> <p>To assign a label to the entire repeated series of dialogs, enter the label in the <b>Label</b> box. (See <a href="#">Customize Repeat Titles.</a>)</p>
<p>Customize how the dialog displays variable questions and control how the dialog functions</p>	<p>Click the <b>Options</b> tab and make your selections. (See <a href="#">Change a Dialog's Options.</a>)</p>
<p>Make the contents of your dialog dynamic based on answers a user enters</p>	<p>Click the <b>Script</b> tab and create a script. (See <a href="#">Use Scripts to Add Power to Your Dialogs.</a>)</p>
<p>Provide helpful information about the dialog for the user</p>	<p>Click the <b>Resource</b> tab and enter the resource text. The information you enter here appears in the resource pane during the interview. (See <a href="#">Add Resource Information to a Variable or Dialog.</a>)</p>
<p>Have two or more variables appear side by side in the dialog</p>	<p>Click the <b>Layout</b> tab and drag variables to the line you want. (See <a href="#">Change the Layout of Variables in a Dialog.</a>)</p>
<p>View a list of all components that use this dialog</p>	<p>Click the <b>Used In</b> tab. (See <a href="#">View Relationship Between the Current Component and Other Components.</a>)</p>
<p>See how your dialog looks and works without closing the <b>Dialog Editor</b></p>	<p>Click <b>Test</b>. A test assembly window appears, showing the current dialog. (See <a href="#">Test a Custom Dialog.</a>)</p> <p>If you want to make changes to the dialog, make your changes in the <b>Dialog Editor</b> (without closing the assembly window) and click <b>Update</b>. When you are satisfied with your changes, close the test assembly window and click <b>OK</b>.</p>

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 all components by dragging them back to the **Components** list. To edit a component, double-click it (at either the **Contents** box or the **Components** list).

By default, HotDocs asks variables and dialogs in the order it "reads" them in the template (top-to-bottom,

left-to-right). If you want a dialog to be asked at a specific place in the template, you must use an ASK instruction. See [Control When Your Dialogs Appear](#) for details.

**Warning:** When you add variables to a dialog, the variables and the dialog become linked and used variables appear grayed in the **Components** list of the **Dialog Editor**. Be careful when adding an already-linked variable to a different dialog. If you don't first clear the **Link variables to this dialog** option (**Options** tab), you will remove the variable from the first dialog. (See [Use the Same Variable or Clause in Two or More Dialogs](#).)

**Notes:**

- 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. See [Specify the Width of Answer Fields in the Interview](#).
- 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  **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.)

### Notes:

- To edit a dialog without using Component Manager, make sure your cursor isn't in a variable field. Then, click the  **Edit 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.
- You can specify the width of answer fields in a dialog. See [Specify the Width of Answer Fields in the Interview](#).

# Change a Dialog's Options

You can make several changes to further customize your dialog, such as:

- 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.

You can specify all of these options at the **Options** tab of a custom dialog.

## To change dialog options

1. Open a dialog for editing. (See [Edit a Custom Dialog](#).) HotDocs displays the **Dialog Editor**.
2. Click the **Options** tab. The window changes to show several custom options.
3. Perform any of the following tasks:

To	Do This
Place groups of True/False variables together, clause components together, or child dialogs together so users can choose only one option	<p>Choose <b>Select One</b> from the <b>Selection grouping</b> drop-down list.</p> <p>When HotDocs processes a dialog that has a selection grouping, it automatically 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. Because of this, you should always select <b>None of the Above</b>, which adds a <i>None of the Above</i> option to the list. This way, the dialog can appear answered, even if the user doesn't select an option.</p>
Place groups of True/False variables together, clause components together, or child dialogs together so users can choose from several options	<p>Choose <b>Select All That Apply</b> from the <b>Selection grouping</b> drop-down list.</p> <p>When HotDocs processes a dialog that has a selection grouping, it automatically 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. Because of this, you should always select <b>None of the Above</b>, which adds a <i>None of the Above</i></p>

	option to the list. This way, the dialog can appear answered, even if the user doesn't select an option.
Add a <i>None of the Above</i> option to a single-selection or multiple-selection group	<p>Select <b>None of the Above</b>.</p> <p><b>Note:</b> Providing a <b>None of the Above</b> option for your users gives them the chance to answer a question without selecting any of the options you have presented.</p>
Choose the visual relationship you want between variable prompts and answer fields	<p>Select an option from the <b>Prompt position</b> drop-down list:</p> <ul style="list-style-type: none"> <li>■ <b>Above</b> places the prompt above the answer field.</li> <li>■ <b>Left</b> places the prompt to the left of the answer field. The length of the prompt text can be from 12 to 25 units, as specified in the <b>Maximum units</b> box. If the prompt length exceeds this, the prompt will wrap to the next line.</li> <li>■ <b>Automatic</b> places the prompt according to the values you specify in the <b>Maximum units</b> box. Prompts that are shorter than this number will be placed to the left of the answer field. Prompts that are longer are placed above the answer field.</li> </ul> <p>To align the left edges of the answer fields, select <b>Align left edges</b>.</p> <p><b>Note:</b> One <i>unit</i> is about equal to the width of the character <i>5</i>.</p> <p><b>Note:</b> When variables are placed side-by-side in a dialog, prompts will always appear above the answer field.</p>
<p>Enter a prompt for a child dialog button on the parent dialog</p> <p>(By default, child dialogs appear on parent dialogs as links with the child dialog's name as the prompt. Specifying a prompt lets you provide more instructive text for accessing and then answering questions in the child dialog.)</p>	<p>Enter the new prompt in the <b>Prompt to use when displayed as child dialog</b> box.</p> <p>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.</p>
Keep child dialogs from appearing as buttons on the parent dialog	<p>Clear <b>Show buttons for child dialogs</b>. 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</p>

	dialog to answer questions within it.
Keep the dialog from being asked if none of the variables in the dialog are active (or used by the document)	Select <b>Hide this dialog when all variables are inactive</b> . (See <a href="#">Automatically Disable Unused Variables in Interviews</a> for details.)
Make sure the dialog is asked—even if it has already been answered—when a user clicks the  <b>Next Unanswered</b> button	Select <b>Always stop at this dialog when moving to next unanswered</b> .  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  <b>Next Unanswered</b> button will still see this dialog. (See <a href="#">Tips on Using a Dialog to Display Dialog Elements Only</a> .)
Control whether dialogs are asked automatically	Clear <b>Ask automatically</b> . (See <a href="#">Control Whether Dialogs are Asked Automatically</a> .)
Cause variables in the dialog to be asked separately, rather than together in a dialog	Clear <b>Link variables to this dialog</b> . If this option is cleared and the dialog is not specified in an ASK instruction, the variable questions are asked individually.
Use one variable in two or more dialogs	Clear <b>Link variables to this dialog</b> . (See <a href="#">Use the Same Variable or Clause in Two or More Dialogs</a> .)
Identify an answer source file from which the user can choose answers	Either type the name of an answer source file in the <b>Answer source</b> box, or click the drop-down button and select an existing answer source. (See <a href="#">Suggest an Answer Source for Dialogs</a> .)  Click the <b>Map Variables</b> button to link an existing answer source to a dialog where the variable names are not the same. (See <a href="#">Share Answers Between Two Dialogs</a> .)  You can also link an answer source to an Outlook Contacts list or to a Time Matters Client or Matter Record.

**Note:** You can control the width of answer fields in a dialog. See [Specify the Width of Answer Fields in the Interview](#).

# Change the Layout of Variables in a Dialog

You can change the appearance of dialogs, 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, which tells you where the variable will be moved 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.

4. 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.)

### Notes:

- If you place only two components on the same line, you can make one of the components wider than the other by right-clicking on 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. (See [Specify the Width of Answer Fields in the Interview](#) for further instructions.)
- To edit a component or remove it from the dialog, right-click on the component and select **Edit** or **Remove** from the shortcut menu.
- When two or more components appear on the same line, the prompts will automatically appear above the answer field—you cannot specify a different prompt alignment.

# 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 present a correct dialog 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

1. 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**.

**Warning:** 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 When Your Dialogs Appear](#).)

# 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**.

By default, a dialog is asked: 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.

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](#)). 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**.

**Note:** This option replaces the need for using ASK NONE / ASK ALL instructions. However, instructions for using them are still available here.

# 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  **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** box, or select **Generate name automatically**, which allows HotDocs to name the component.
4. In the **Style** group, make sure **Text** is selected.
5. In the **Display text** box, 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. (See [Overview: Dot Codes](#).)
7. Optionally, if the dialog is repeated as a spreadsheet, click the **Options** tab and select one of the following options:
  - **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.

### Notes:

- To edit the dialog element, double-click on 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. See [Using a Dialog to Display Only Dialog Element Text](#).
- You can customize dialog element text based on users' answers by including variables and other scripting instructions in the element text. See [Use Variables and Scripts in Prompts, Dialog Element Text, and Plain Text Resources](#) for details.

# 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  **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** box, 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** box, click the  **Browse** button and locate the application's .EXE file.
6. In the **Link text** box, enter the text that will appear on 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 box if you select Image.)
  - 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 link placed on the dialog.

### Notes:

- 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  **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** box, or select **Generate name automatically**, which allows HotDocs to name the component.
4. In the **Style** group, select **Script link**.
5. In the **Computation** box, 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** box, enter the text that will appear on 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** box if you select **Image**.)
  - 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 link placed on the dialog.

# Add a Hyperlink to a Dialog

You can include a hyperlink on a dialog, which can point to any URL. When the user clicks the hyperlink, the Web page will open. This may be useful if you need users to refer to a specific Web page when answering questions in a dialog.

One reason you may want to add a hyperlink to a dialog is to let HotDocs Server users retrieve answers from a database. For example, the site administrator can create a Web page at the specified URL that presents information from a database and allows users to select the appropriate data. Once the data is selected, the Web page is closed and the answer fields in the dialog are filled with answers from the database. *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  **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** box, or select **Generate name automatically**, which allows HotDocs to name the component.
4. In the **Style** group, select **Web link**.
5. In the **URL** box, 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** box, enter the text you want to use as the hyperlink.
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** box if you select **Image**.)
  - In the **Display in** group, select whether the link should appear when the dialog is displayed in **Desktop interviews**, **Server interviews**, or both.

**Warning:** If your hyperlink is designed to allow HotDocs Server users to access a database via a Web page, you must select **Server interviews**. Note that the hyperlink will not work in a desktop interview.

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)** box.

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.

**Note: HotDocs Server Users:** If using a Web link to access database answers, any image files you use with the link must be located in the same folder as the rest of the HotDocs Server images. When testing the

interview in a browser, the image file must be in the *JavaScript Files* folder (for example, *C:\Program Files\HotDocs 6\JavaScript*).

# Add Spacing and Separator Lines to a Dialog

Sometimes you need to add white (or empty) space or separator lines to a dialog. This can help you better organize the dialog.

When adding empty space to a dialog, you can control how much space will be used.

When adding a separator, you can include a caption with the separator. For example, if you have two groups of questions in a dialog—one about the plaintiffs and the other about the defendants in a case—you can separate each group with a line and include the text *Plaintiff Information* or *Defendant Information* on each separator line.

## 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  **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** box, or select **Generate name automatically**, which allows HotDocs to name the component.
4. In the **Style** group, select **Vertical spacing**.
5. In the **Spacing** box, 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** box. If you want it to be larger or smaller, enter the appropriate number in the box.)
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** box, 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** box, and then enter any of the following options:
  - In the **Font size** box, enter a font size for the caption, based on the percentage of the rest of the dialog text's 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 box.)
  - Click the **Justification** drop-down button and select which side of the dialog the caption will be aligned with.
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 divider placed in the dialog.

# Add a Graphic File to a Dialog

Sometimes you may want to include images in your dialogs. Images may help users better understand questions in the dialog, or they may allow you to display your company logo or other icon. The types of files you can add include .BMP, .JPG, .GIF, and .PNG.

## 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** box, or select **Generate name automatically**, which allows HotDocs to name the component.
4. In the **Style** group, select **Image**.
5. In the **Image file** box, click the  **Browse** button and locate the image file.
6. Optionally, select an alignment option: **Left** (aligns the image with the left edge of the dialog pane), **Center** (aligns it in the center of the dialog), and **Right** (aligns the image with the right edge of the dialog pane).
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.

**Note:** Bitmap files are not supported in HotDocs Server.

# 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.
3. Select a dialog from the **Dialog** drop-down list, or click the  **Edit Component** button to create a new dialog. (See [Gather Questions into a Custom Dialog](#).)
4. Click **OK**. The ASK instruction is inserted in the template.

### Notes:

- 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** drop-down menu in the HotDocs toolbar and choosing **ASK Field**, or by right-clicking in the template and choosing **ASK Field** from the shortcut menu.

## 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** box.
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

## Warnings:

- If you are using HotDocs Professional Edition, you should specify this property for the dialog itself. See [Control Whether Dialogs are Asked Automatically](#).
- **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 2008. 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. A *variable* is asked automatically when all of the following conditions are true:

- The answer for the variable is used (for example, tested in an IF instruction or merged into text), and
- The variable hasn't been asked before in the interview, either by itself or as part of a dialog, and
- The variable is marked **Ask automatically**.

A *dialog* is asked automatically when all of the following conditions are true:

- The dialog has not been asked before in the interview, and
- A variable to which the dialog is linked is asked automatically.

Additionally, a dialog will be asked automatically when it is used in a REPEAT instruction.

It is particularly useful if you need to repeat a dialog but you don't want the dialog to be asked during the interview.

```
«ASK NONE»
```

```
«REPEAT Signer Information» (Tip: Signer Information is a dialog that merges answers from two other REPEAT instructions. (See WHILE EXPRESSION for an explanation on how to do this.) )
```

```
«ASK ALL»
```

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** (tells HotDocs to display every variable after the instruction, even if the variable has already been assigned an answer), or **NONE** (tells HotDocs to display none of the variables following the instruction).

## To change the ASK mode using a Computation variable

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** box.
3. Replace **DIALOG** with one of the following keywords: **ALL** (tells HotDocs to display every variable after the instruction, even if the variable has already been assigned an answer), or **NONE** (tells

HotDocs to display none of the variables following the instruction).

**Warning:** If you change the ASK mode in a computation, you should always change it back to what it was originally (ALL or NONE).

# 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](#).)

**Note:** 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** box. (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, or they can click the child dialog icon, which automatically selects the check box or option and then displays the child dialog.

**Warning:** 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** (groups child dialogs so users can choose only one dialog for answering) or **Select All That Apply** (groups child dialogs so users can choose any number of dialogs for answering).

**Note:** 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.

# 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.

**Warning:** 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](#).)

**Note:** 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

**Warning:** Dialog scripts should only change the appearance of variables in a dialog—they should not be used to compute answers.

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:

- **LIMIT NUM** limits the number of answers that can be entered when a dialog is repeated.
- **GRAY VAR** grays (or dims) a variable so the user can see it, but can't answer it.
- **GRAY ALL** grays all the variables in the dialog.
- **UNGRAY VAR** makes a grayed variable in the dialog active.
- **UNGRAY ALL** makes all grayed variables in the dialog active.
- **HIDE VAR** hides a variable so the user can't see it in the dialog.
- **HIDE ALL** hides all the variables in a dialog.
- **SHOW VAR** restores a hidden variable to view.
- **SHOW ALL** restores all hidden variables to view.
- **REQUIRE VAR** requires users to answer a certain variable before they can proceed to the next dialog.
- **REQUIRE ALL** requires users to answer all of the questions in a dialog before they can proceed to the next dialog.
- **CONCEAL VAR** keeps variables from appearing in an answer source selection dialog box. (See [Suggest an Answer Source for Dialogs](#).)
- **OMIT VAR** operates much like the CONCEAL instruction, except when used, it not only keeps the variable from appearing in the answer source selection dialog box, it treats the variable as if it doesn't belong to the answer source at all. (See [Suggest an Answer Source for Dialogs](#).)
- **SET VAR TO VALUE** lets you enter an answer for a variable that can be seen—but not changed—by the user. (See [Create a SET Instruction](#).)
- **DEFAULT VAR TO VALUE** lets you suggest an answer for an unanswered variable. Users can either accept the suggestion or enter a different answer.

# Suggest an Answer Source for Dialogs

You can suggest an answer source (a list of answers from which a user can select one) for a dialog. Users open the list when answering questions in a dialog and pick an existing set of answers, instead of entering one manually. For example, perhaps you have a large number of attorneys in your firm. Rather than enter information for a specific attorney 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.)

See [Tips for Working With Answer Sources](#) for more information on answer source files.

## To create an answer source

1. Edit or create a dialog that contains the variables you want to associate with an answer source file. The **Dialog Editor** appears.

**Warning:** 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. Select an existing answer source file by clicking the **Answer source** drop-down button, or create an answer source by typing a file name with a three-letter extension, such as .ANS or .HPL, in the **Answer source** box. (You can use an existing answer source by clicking the  **Open** button and then searching for the file.)
4. If you are linking an existing answer source to a dialog, and the variable names used in the answer source don't match the variable names in the dialog, you must associate the variables so they can share information. Click **Map Variables** to do this. (See [Share Answers Between Two Dialogs](#).)
5. Click **OK** to close the **Dialog Editor**.

Once your answer source is created, a **Select** button appears on the dialog during the interview. The user can click this button, select an answer or enter new answers, and the answers are saved to the answer source file you specified.

### Warnings:

■ Be careful about 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 use 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.

# Share Answers Between Two Dialogs

You can use one answer source to provide answers either in the same template or in multiple templates—even if the variables that must share information have different names.

## 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** box, 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.
5. 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.

## 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.)

**Note:** If you don't want all the variables from the answer source to appear in the dialog, you can HIDE them. If you don't want them to appear in the **Select From Answer Source** dialog, you can CONCEAL them. If you don't want the variables to appear in the answer source at all, you can OMIT them.

# Let Users Retrieve Answers from a Time Matters Record

**Note:** 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 file.

# Let Users Retrieve Answers from an Outlook Contacts List

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.

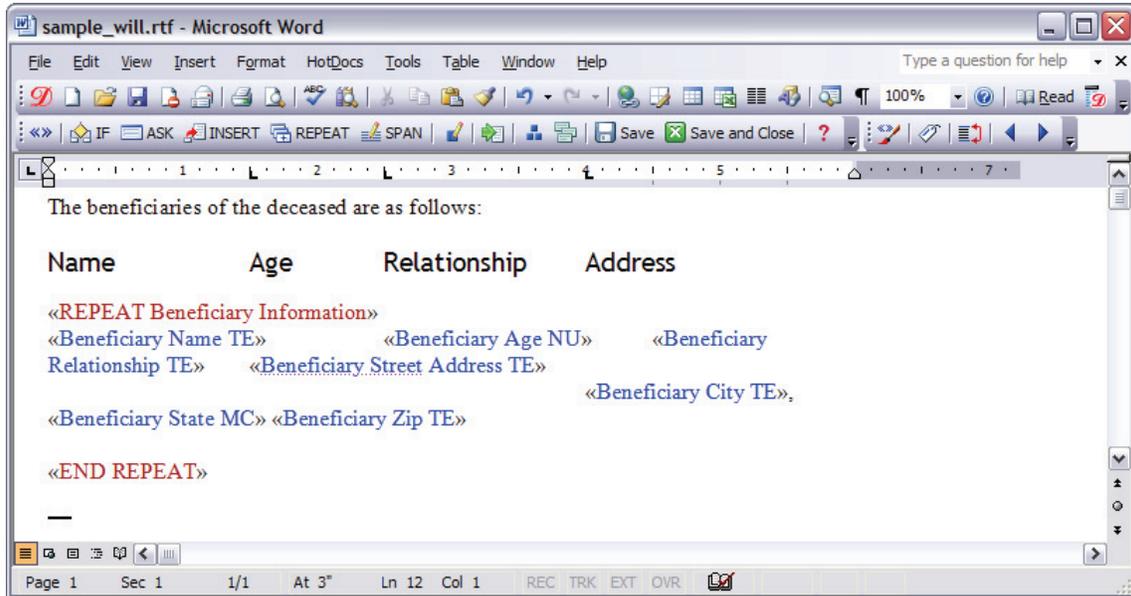
**Note:** This feature is supported in Microsoft Outlook 2000 and later.

# Using Repeat Instructions

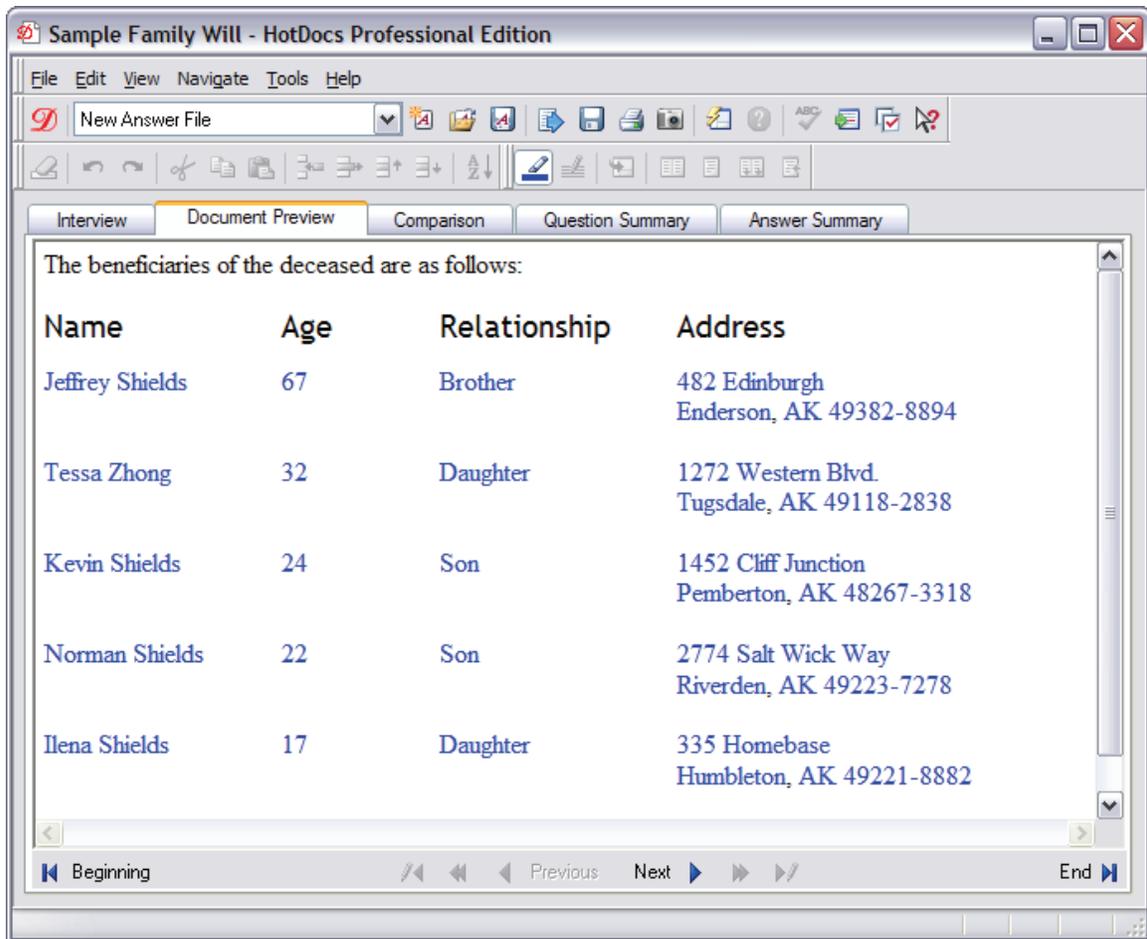
## Overview: Include Lists in Your Documents

You can include lists of answers in your documents by repeating variables and allowing the user to enter 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.

The following shows a REPEAT instruction inserted into a template:



This shows the list of answers the REPEAT instruction merged into the template:



When you create a REPEAT instruction, you will be asked to create or select a custom dialog to control the instruction. During the interview, users can answer the dialog over and over again, until they have entered all the answers for the list. Or, you can limit the number of answers a user can enter for a repeated variable.

Additionally, you can repeat any text, instructions, or other parts of a template. Everything inside a REPEAT instruction is repeated along with the variables.

You can make lists in a document appear the way you like—either in sentence style with punctuation, or in column style. You can also have HotDocs sort, number, and filter lists automatically.

# Create 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. For example, perhaps you must insert a list of personal property items with their associated values. You can create the two variables and then repeat them. When users complete the interview, they can enter as many property items as they have. Those answers are merged into the finished document.

By default, when HotDocs inserts a REPEAT instruction 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, or if you want to merge a return character only under certain situations, you can specify an option that controls this. See [Control Whether Returns Are Inserted After Instructions in Text Templates](#) and [Understand How Returns Are Inserted After Instructions](#).

**Warning:** Often, you must repeat information within a word processor table. To repeat a row in the table, see [Use a Word Processor Table to Display a List](#). To repeat the contents of a single cell, see [Create a REPEAT Instruction Using a Computation Variable](#).

## To create a REPEAT instruction

1. Select the template text, including variables you want to repeat, and click the  **REPEAT Field** button. The **REPEAT Field** dialog box appears.
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  **Edit Component** button. The **Dialog Editor** appears.
4. Click the **Style** drop-down button and select the format for the repeated dialog. (See [Choose a Presentation Style for the Repeated Dialog](#).)
5. If you are creating a new dialog, add the variables you want repeated to the dialog. (See [Gather Questions into a Custom Dialog](#).)
6. Click **OK** at the **Dialog Editor**. HotDocs displays the **REPEAT Field** dialog box again.
7. 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*.) (See [Punctuate a Sentence-Style List](#).)
  - Click **Show Advanced** to select sorting and filtering options for the repeated dialog. (See [Sort a List of Answers and Filter a List of Answers](#).)
8. Click **OK**. The REPEAT instruction is inserted in the template.

You can always change punctuation, sorting, or other options by editing the instruction. See [Edit a REPEAT Instruction](#). (To edit the *contents* of the repeated dialog, see [Edit a Custom Dialog](#).)

### Notes:

- You can find the total sum of a series of repeated Number variables, such as a list of dollar amounts. See [Get Sum Totals for Repeated Number Variables](#) for details.
- In Microsoft Word, you can also insert a REPEAT instruction either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **REPEAT Field**, or by right-clicking in the template and choosing **REPEAT Field** from the shortcut menu.

# Edit a REPEAT Instruction

You can make changes to REPEAT instructions, including editing the repeated dialog's contents, adding a filter, choosing an answer-sorting order, and adding punctuation to the list of answers.

## To edit a REPEAT instruction

1. Position the cursor in the opening REPEAT instruction.
2. Click the  **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  **Edit Component** button to edit the dialog (including adding or removing variables from the dialog and selecting presentation options for the dialog.) (See [Change a Dialog's Options](#).)
  - Click the **Format** drop-down button and select a punctuation format. (See [Punctuate a Sentence-Style List](#).)
  - Click Show **Advanced** to specify a sorting order and add a filter. (See [Sort a List of Answers](#) and [Filter a List of Answers](#).)

**Note:** If you're using a complicated series of REPEAT instructions, you can match REPEAT instructions with END REPEAT instructions. See [Match REPEATs with END REPEATs](#) for details. You can also assign color coding to REPEAT instructions to help you view relationships between them and other instructions in the template. See [Assign Colors to Fields and Instructions in Templates](#).

# Choose a Presentation Style for the Repeated Dialog

You can choose how a repeated dialog appears during the interview, either by presenting the list as a series of dialogs or as a single spreadsheet with multiple entry points.

## To choose a presentation style for the list

1. Edit the REPEAT instruction. (See [Edit a REPEAT Instruction](#).) The **REPEAT Field** dialog box appears.
2. Click the  **Edit Component** button to open the **Dialog Editor**.
3. Click the **Style** drop-down button and make a selection, based on the following descriptions:
  - **Repeated Series** causes a regular dialog to appear with an answer field for each variable. When the user answers the questions and advances to the next dialog, an unanswered version of the same dialog appears again until the user clicks **Next** at an empty dialog. A number next to the dialog icon in the interview outline tells which repetition the dialog is on. (Displaying the dialog over and over can be a good approach if the dialog includes several variables.)
  - **Spreadsheet** causes a single dialog to appear with a column for each variable and an unlimited number of rows. Users fill in as many rows as they need. (The spreadsheet format lets users 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.)

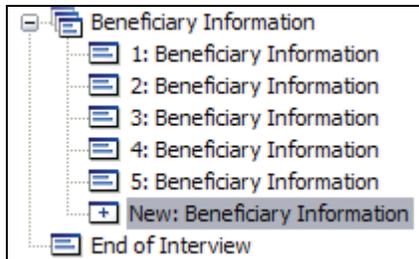
**Warning:** If the repeated dialog contains a multiple-selection Multiple Choice variable, the dialog must use the **Repeated Series** style—it cannot use the **Spreadsheet** style. (See [Customize a Multiple Choice Variable](#).)

### Notes:

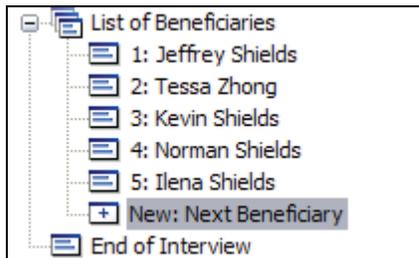
- For information on using the **Spreadsheet on Parent** option, see [Display a Child Dialog Directly on the Parent Dialog](#).
- You can control the width HotDocs uses for the columns in a spreadsheet. See [Specify the Width of Answer Fields in the Interview](#).
- When working with a repeated series, you can assign a **Repeated Series Label** for the entire series. This label is different from the dialog title, which can be customized for each individual repetition. See [Customize Repeat Titles](#) for more information.

# 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:



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:



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 no answer has been given, the script will merge other text that will help the user know the dialog is unanswered. Otherwise, HotDocs will merge an unanswered variable placeholder in the title, like this:



## Part 1: To assign a repeated series label

1. Edit the repeated dialog and select **Repeated Series** at the **Style** drop-down list. (See [Edit a Custom Dialog](#) and [Choose a Presentation Style for the Repeated Dialog](#).)
2. Enter a generic title in the **Label** box.
3. In the next part, you will create a computation script that merges the user's answer in the dialog title.

## Part 2: To create a Computation variable

1. While still at the **Dialog Editor**, right-click in the **Title** box and choose **Variable Field** from the shortcut menu.
2. Choose **Computation** as the **Variable type** and enter a name in the **Name** box.
3. Click the  **Edit Component** button. The **Computation Editor** appears.
4. In the **Script** box, enter the script that tests whether the variable you want to use in the title is answered. The script should also merge the text you want to be used in the title. An example script would be:

```
VALUE(Beneficiary Name, "Next Beneficiary")
```

5. Click **OK** at both the **Computation Editor** and at the **Variable Field** dialog box. The variable is merged in the title field.
6. Click **OK** to close the Dialog Editor.

# Punctuate a Sentence-Style List

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*.

## To punctuate a list

1. At the template, edit the REPEAT instruction you want to punctuate. (See [Edit a REPEAT Instruction](#).) The **REPEAT Field** dialog box appears.
2. Click the **Format** drop-down button.
3. Select a punctuation style, as explained in the following table:

To	Do This
Insert a comma even when there are only two items	Choose <b>a, and b</b> .
Punctuate a list, but keep <i>and</i> from being used	Choose <b>a, b</b>
Keep the last comma before <i>and</i> from being inserted	Choose <b>a, b and c</b> .
Use <i>or</i> instead of <i>and</i>	Choose <b>a, b, or c</b> .
Use semicolons instead of commas between list items	Choose <b>a; b; and c</b> .
Enter your own punctuation style	Type your style in the <b>Format</b> box. Base it on the other formats in the list. For example, you could use <i>or</i> or some other text instead of <i>and</i> , or you could use a semicolon instead of a comma.

### Notes:

- If you want to create a list in a column, see [Make a List Appear as One or More Columns](#).
- 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. See [How Example Formats are Interpreted](#) for a list of formats for repeated dialogs.
- You can punctuate the same list differently at another place in the document. Choose a punctuation style separately at each REPEAT instruction field.

## Sort a List of Answers

You can have HotDocs alphabetize your list automatically, 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. (See [Edit a REPEAT Instruction](#).) The **REPEAT Field** dialog box appears.
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** (A to Z, 1 to 9), or **Descending** (Z to A, 9 to 1).
5. Optionally, to sort on a second level, select a variable at the **Then By** box and choose either **Ascending** (A to Z, 1 to 9), or **Descending** (Z to A, 9 to 1).

When you close the **REPEAT Field** dialog box, you will see the names of the variable(s) on which you are sorting added to the REPEAT instruction.

**Note:** 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.

## Filter a List of Answers

You can 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.

### To filter a list

1. At the template, edit the REPEAT instruction. (See [Edit a REPEAT Instruction](#).) The **REPEAT Field** dialog box appears.
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** box.
5. In the **Script** box, 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:  

```
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.

**Note:** 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.

# 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 enter—it only controls the number of answers that can be viewed at any given time.)

## To change the number of viewable rows in a spreadsheet

1. Edit the REPEAT instruction. (See [Edit a REPEAT Instruction](#).) The **REPEAT Field** dialog box appears.
2. Click the  **Edit Component** button to open the **Dialog Editor**.
3. Click the **Style** drop-down button and choose **Spreadsheet**.
4. In the **Rows to display** box, either type a number, or click the up or down arrows to select a number.

# Limit 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 a dialog is repeated. Once the limit is reached, users can no longer enter anymore answers.

## To limit the number of answers that can be entered

1. Open the repeated dialog for editing. (See [Edit a Custom Dialog](#).) The **Dialog Editor** appears.
2. Click the **Script** tab. The window changes to show scripting options.
3. Drag **LIMIT NUM** from the **Instruction models** list into the **Script** box.
4. Replace the **NUM** placeholder with the maximum number of answers you want the user to enter.

### Notes:

- You can let the user change the limit each time a document is assembled. To do this, replace **NUM** 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** box. (Make sure **Spreadsheet** is selected as the **Style**.)

# Get 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. Create a computation that contains the expression **SUM( NUM\_VAR )** if the repeated variable you are totaling is a list of Number variables, or **SUM( COMPUTATION\_VAR )** if you are totaling a list of Computation variables. (See [Customize a Computation Variable](#).)
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.

# Use 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 automatically adds rows to the table.

When you insert a REPEAT instruction in a table, HotDocs does not include an END REPEAT instruction—the end of the row signifies the end of the repeat. Additionally, you can repeat only a single row of the table, and each row must have the same number of columns. Finally, to REPEAT the contents of a *single cell* of a table, you must use a Computation variable. See [Create a REPEAT Instruction Using a Computation Variable](#).

## To insert a list into a word processor table

1. 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.)
2. Insert the variables in the table, with one variable per column. (See [Insert a Variable Field in a HotDocs Template](#).)
3. Once your variables are created, position the cursor in the first table cell—before the first variable.
4. Click the  **REPEAT Field** button. The **REPEAT Field** dialog box appears.
5. Click the **Dialog** drop-down button and select the dialog you want to repeat. (To create a new dialog, click the  **Edit Component** button. See [Gather Questions into a Custom Dialog](#).)
6. Select a repeat style for the dialog. (See [Choose a Presentation Style for the Repeated Dialog](#).)
7. When you are finished, click **OK** at the **REPEAT Field** dialog box. The REPEAT instruction is inserted in the table. For example:

Quantity	Item Number	Title	Unit Price	Item Amount
«REPEAT Product Information»«Quantity»	«Item Number»	«Title»	\$«Unit Price»	\$«Item Amount»

## 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.

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  
Maddie  
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  
Maddie     7  
Adam       2
```

**Note:** By default, when HotDocs inserts a REPEAT instruction 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, or if you want to merge a return character only when repeating an entire paragraph, you can specify an option that controls this. See [Control Whether Returns Are Inserted After Instructions in Text Templates](#) and [Understand How Returns Are Inserted After Instructions](#).

# 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.

## To determine the number of entries in a list

1. Create a Computation variable containing the expression `COUNT( DIALOG )`. (See [Customize a Computation Variable](#).)
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 simply 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.)

## 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. (See [Insert a Variable Field in a HotDocs Template](#) and [Customize a Number Variable](#).) Instead of typing a variable name, however, click the **Variable** drop-down 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»

## 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 enter the index number of the answer you need between brackets, just after the variable name.

### To use explicit indexing

1. Insert the variable whose value you want to retrieve in the template text. (The variable should be repeated using a REPEAT instruction earlier in the template.)
2. Once the variable is inserted, place your cursor immediately after the variable name and enter—between brackets—the index number of the repetition you want to retrieve. This index should be before any additional field formatting notations, for example:

```
The second alternate will be «Alternate Board Member[2]:LIKE  
THIS».
```

#### Notes:

- 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** box, immediately after the variable name (like in the example above).
- Normally, a repeated variable must be inside a REPEAT instruction, but when you 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. See [Change Component File Properties](#) for details.)

## Retrieve Information Other Than Answers from a REPEAT Instruction

You can use a computation to retrieve information (other 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
```

# 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 Whether 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 third option, using smart returns, depends on the text surrounding the instruction. The following examples show how smart return insertion works. In these examples, bracketed text represents the area of text selected when you click the  **IF Field** button. When necessary, HotDocs inserts a **Keep Return** code (or vertical bar (|)) at the end of an instruction to indicate not to delete the return immediately following the instruction. (You will see this explained in the examples below.)

## Example 1

If text is selected when you click the  **IF Field** button, and the selected text represents a complete paragraph, HotDocs will insert returns following the instructions. For example:

```
Here is a paragraph.  
[Here is another paragraph.  
]Here is a third paragraph.
```

In this case, the outcome will be:

```
Here is a paragraph.  
«IF Instruction»  
Here is another paragraph.  
«END IF»  
Here is a third paragraph.
```

**Warning:** 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.

## Example 2

If text is selected when you click the  **IF Field** button, and the selected text is anything other than one or more complete paragraphs, HotDocs does not insert returns. For example:

```
Here is a paragraph.
```

[Here is another paragraph.]

Here is a third paragraph.

Note that the entire second paragraph is NOT selected because the paragraph marker isn't included. The outcome of smart insertion will be:

Here is a paragraph.

«IF Instruction»Here is another paragraph.«END IF|»

Here is a third paragraph.

Since HotDocs didn't insert a return following the END IF instruction, and since there is a return that "belongs" to the user right after the END IF instruction, HotDocs includes a Keep Return code in the END IF instruction so the return will *not* be deleted.

The outcome of assembly will be:

Here is a paragraph.

Here is another paragraph.

Here is a third paragraph.

Without the return code, the result would be:

Here is a paragraph.

Here is another paragraph. Here is a third paragraph.

Another example is:

Dr. «Name of Doctor» is my [favorite ]doctor.

which will yield:

Dr. «Name of Doctor» is my «IF Instruction»favorite «END IF»doctor.

### Example 3

If text is NOT selected when you click the  **IF Field** button, and if the cursor is between paragraphs, HotDocs will insert returns. For example:

First paragraph.

[ ]Second paragraph.

will yield

First paragraph.

«IF Instruction»«END IF»

Second paragraph.

### Example 4

If text is NOT selected when you click the  **IF Field** button, and if the cursor is within a paragraph, HotDocs does *not* insert returns. For example:

First paragraph.[ ]

Second Paragraph.

will yield:

First paragraph.«IF Instruction»«END IF|»

Second Paragraph.

# 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. One reason for this is if your list of answers must appear in a single table cell. Also, using a computation allows you to quickly insert a list in more than one location in the template. (One drawback of this approach is that punctuating, sorting, and filtering will be applied wherever the computation is inserted—the list won't appear differently in different parts of your assembled document.)

## To use a REPEAT instruction in a computation

1. At the template, create a Computation variable. (See [Customize a Computation Variable](#).)
2. In the **Script** box, type either empty quotation marks ( "" ) or a zero (0) as the starting value for the computation. (These set the repeated value to “nothing.”)
3. Enter a REPEAT instruction, followed by the dialog name.
4. Enter the RESULT instruction, followed by the names of the variables whose answers you want to merge. (Use the plus ( + ) operator to add variables, text, space characters (including a hard return), and so forth, to what's being repeated.)
5. Enter 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](#).)
2. Drag **FORMAT "LIST\_FORMAT"** from the **Instruction models** list into the **Script** box on a line of its own, after the REPEAT instruction but before the RESULT instruction.
3. 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](#).)
2. Drag **ASCEND VAR** or **DESCEND VAR** from the **Instruction models** list into the **Script** box on a line of its own, after the REPEAT instruction but before the RESULT instruction.
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
```

**Note:** 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](#).)
2. Once you have added the REPEAT instruction to the script, drag **FILTER COMPUTATION\_VAR** from the **Instruction models** list into the **Script** box. (Place this model on a line of its own, after the **REPEAT** instruction and any punctuation or sorting instructions but before the **RESULT** instruction.)
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** box, 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
```

## 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** box, 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.

## Overview: 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.

**Warning:** Nested repeats cannot be used if you are retrieving answers from a database.

# 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.

## Part 1: To insert a REPEAT instruction for each list

1. Create the variables you want repeated and insert them into the template. (See [Insert a Variable Field in a HotDocs 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  **REPEAT 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  **REPEAT 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.

## Part 2: To insert each repeated dialog into the preceding dialog

1. Edit the dialog in the first (or top) level of the repeat. (See [Edit a Custom Dialog](#).) The **Dialog Editor** appears.
2. Drag the dialog for the second repeat level from the **Available Components** list into the **Contents** box and click **OK**.
3. Repeat this process until each dialog in the nested repeat (except for the first) is inserted into the preceding level's dialog.

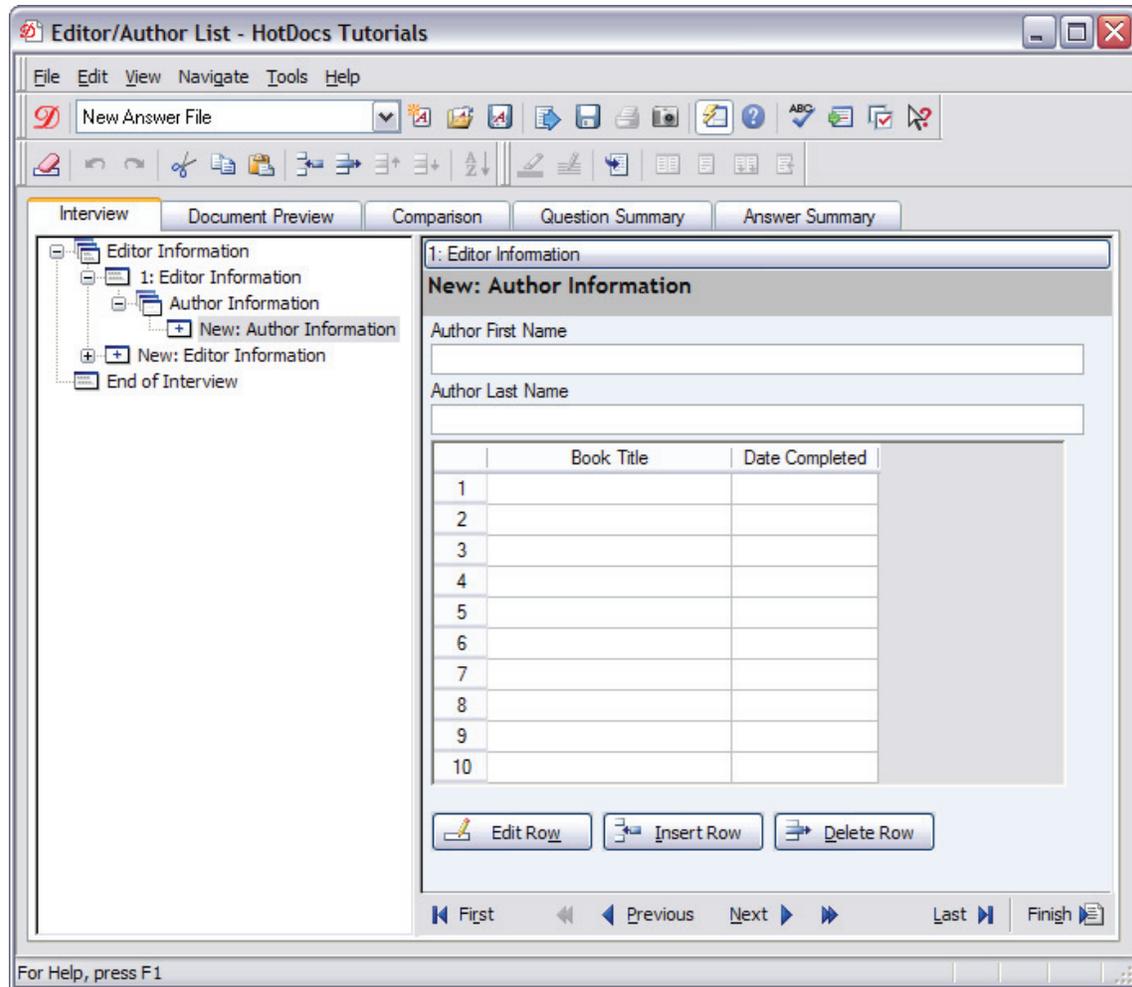
### Notes:

- 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 an Inserted 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 in 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.

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.



## To display a spreadsheet dialog on its parent

1. 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** box 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** box. (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](#).)

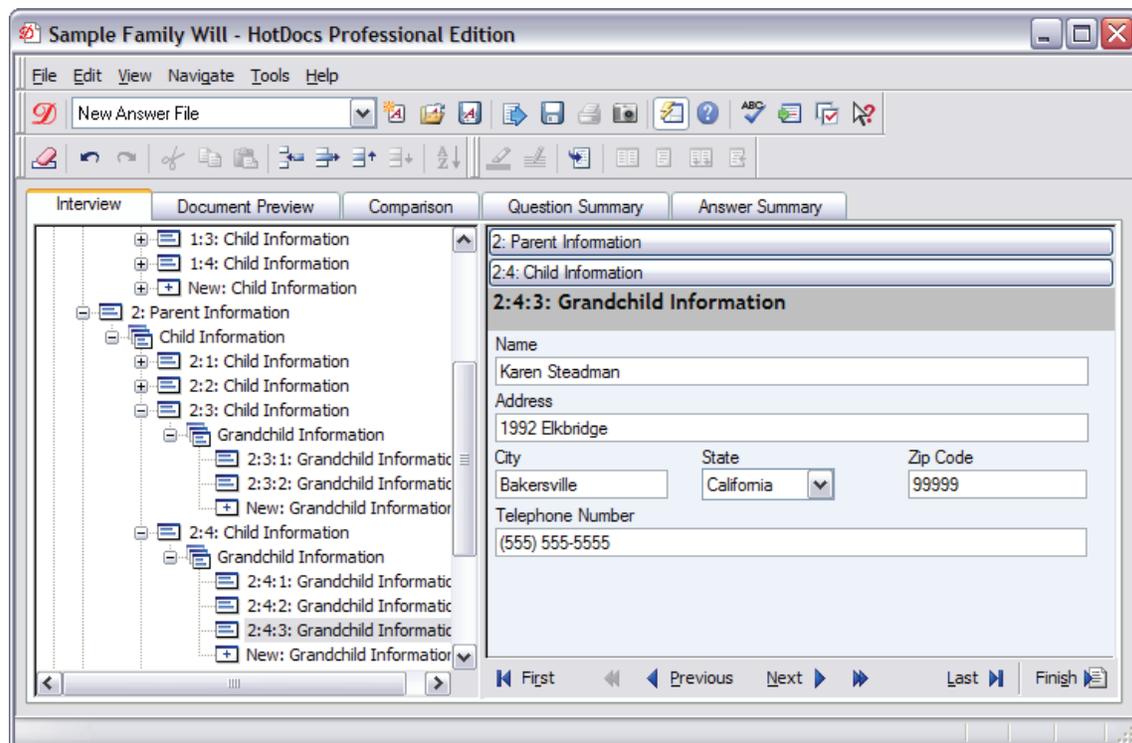
**Note:** It may be useful to review the topic [Create a List Within a List](#), which covers basic information about nested repeats.

## 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 name of the fifth grandchild born to the fourth child of the second parent, you would specify the following in the variable name field: «*Grandchild Name [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 answer entered for the variable *Parent Name*. The **4** in the second title bar directs HotDocs to the fourth answer entered for the variable *Child Name* for the second parent. The **3** in the third title bar directs HotDocs to the third answer entered for the variable *Grandchild Name* 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.

**Note:** When automating a form template, you must enter the number path in the **Variable** box at the **Field Properties** dialog box. (See [Retrieve a Specific Answer From a List](#).)

# Using IF Instructions and Expressions

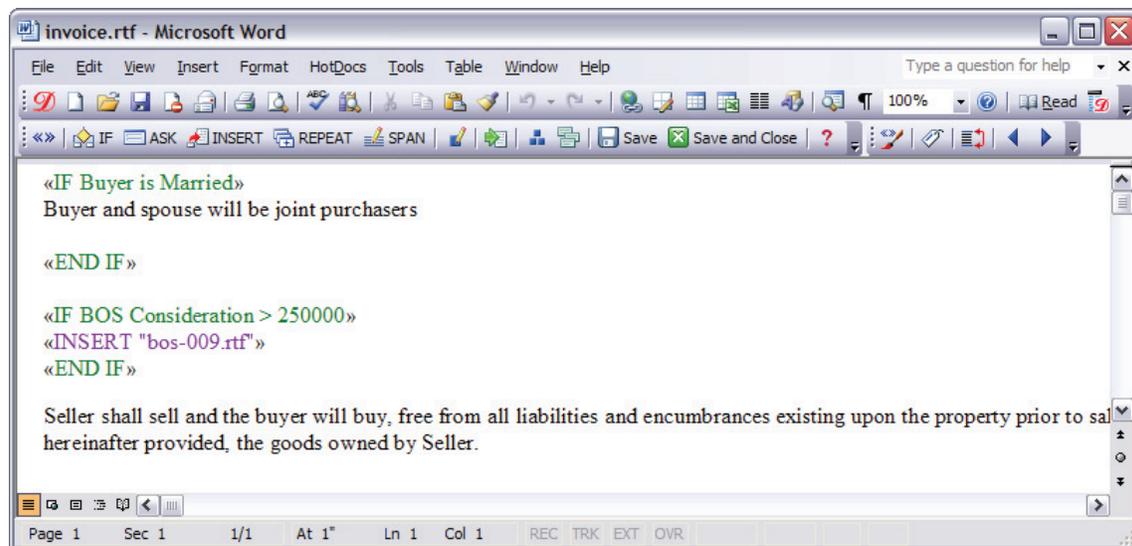
## Overview: Make Parts of Templates Conditional

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 *Buyer and Spouse will be joint purchasers* will be merged into the document only when the answer to the True/False variable, *Buyer is Married*, is *true*.

Also, the sub-template, *bos-009.rtf* (which contains special insurance provisions) will be inserted only if the answer to the variable *BOS Consideration* is greater than *\$250,000*.



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.

# Condition Text Using True/False Variables

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.

In some cases, 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 (tells HotDocs that if the preceding IF instruction is *false*, HotDocs must test this instruction and insert text into the assembled document if it's true; otherwise move to the next instruction) and ELSE (tells HotDocs that if all the preceding conditions (IF and ELSE IF instructions) are false, the part of the template that follows ELSE should be included) 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.

**Note:** 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.

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 someplace 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»
```

By default, when HotDocs inserts an IF instruction in the template, it inserts 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 Whether Returns Are Inserted After Instructions in Text Templates](#) and [Understand How Returns Are Inserted After Instructions](#).

## To add or exclude text based on a single 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  **IF Field** button. The **IF Field** dialog box appears.
3. Select **IF True/False Variable**.

4. Type a name in the **True/False variable** box, or select an existing variable from the drop-down list.
5. Optionally, click the  **Edit 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.

**Note:** When the True/False variable appears during document assembly, the variable name is used as the prompt. You can enter a **Prompt** to make it more specific. (See [Tips on Naming True/False Variables and Create a Prompt for a Variable](#).) Additionally, you can include a resource (see [Add Resource Information to a Variable](#).)

### To add or exclude multiple versions of text using several True/False variables

1. Highlight the first conditional part of the template and create an IF instruction.
2. Select the IF instruction's **END IF** keywords together with the next section of conditional text and click the  **IF Field** button. The **ELSE Field** dialog box appears.
3. Select **ELSE IF True/False Variable**. The dialog box changes to show the **True/False variable** box.
4. Type a name in the **True/False variable** box, or select an existing variable from the drop-down list.
5. Optionally, click the  **Edit Component** button to make changes to the variable's properties. (See [Customize a True/False Variable](#).)
6. Click **OK** to complete the instruction.
7. Repeat this procedure for each section of conditional text, making sure you select **END IF** along with the section of text.

### To add or exclude text based on all preceding variables being false

1. 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.
2. Click the  **IF Field** button. The **ELSE Field** dialog box opens.
3. 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*.)

#### Notes:

- 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](#).)
- In Microsoft Word, you can also access the **IF Field** command either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **IF Field**, or by right-clicking in the template and choosing **IF Field** from the shortcut menu.

# Condition Text Using a True/False 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 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.

To create an expression, you must create a script that tests whether certain conditions exist. 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 Whether Returns Are Inserted After Instructions in Text Templates](#) and [Understand How Returns Are Inserted After Instructions](#).

## 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  **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.

The following is an example of 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 TF AND AGE( Client's Birth Date DA ) > 25 AND  
Residence MC = "Pennsylvania"
```

5. Click **OK**. HotDocs inserts the expression around the part of the template you selected.

In some templates, you may have multiple versions of text, of which you want to include only one version in the document. You can condition each section using a series of **ELSE IF** (tells HotDocs that if the preceding IF expression is *false*, HotDocs must test this expression and insert text into the assembled document if it's true; otherwise move to the next expression) and **ELSE** (tells HotDocs that if all the preceding conditions (IF *and* ELSE IF expressions) are false, the part of the template that follows ELSE should be included) expressions. The process of creating these expressions is similar to creating them for True/False variables. To understand how to do this, refer to the topic [Condition Text Using True/False Variables](#), and instead of choosing **ELSE IF True/False variable** at the **IF Field** dialog box, choose **ELSE IF expression**.

**Note:** 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. Also, a true/false expression can be only 250 characters long. If longer, you must use a Computation variable.

# 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  **IF 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. (See [Condition Text Using True/False Variables](#).)
  - Create a true/false expression. (See [Condition Text Using True/False Expressions](#).)
  - Click the  **Edit Component** button and type a prompt in the Prompt box 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](#).)

## 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.

### Notes:

- 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 IF 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 [Condition Text Using True/False Variables](#) and [Condition Text Using True/False 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** box, or you can drag the models to the **Script** box 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
```

# 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.

## 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 Whether 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 INSERT Instructions

---

## Overview: Insert Templates into Templates

You can insert one template into another by using an INSERT instruction. For example, you might want to:

- Include boilerplate text in multiple documents.
- Include a set of related documents in one main template so users can choose the document they want.

During document assembly, when HotDocs finds an INSERT instruction, it stops assembling the main document so it can assemble the inserted document. When it finishes, it continues assembling the main document.

In addition to inserting existing text templates, you can also convert part of your current template into an inserted template so you can insert it into other templates.

**Warning:** You cannot place INSERT instructions in template headers or footers. Additionally, you cannot place INSERT instructions in a table.

**Note:** Templates can be inserted from any location. See [Inserted Template Locations](#) for details.

You can also use an ASSEMBLE instruction to assemble additional documents from within one template. For example, say an answer a user provides while assembling a document means they must then assemble another document. Rather than make them assemble it manually, you can have HotDocs assemble it automatically. You do this by inserting an ASSEMBLE instruction in the template. Then, once the first assembly finishes, HotDocs will start an additional assembly. (See [Differences Between ASSEMBLE and INSERT Instructions](#).)

# Make Part of a Template into an Inserted Template

You can select part of a template and turn it into a separate, inserted template. Once you do this, you can insert it in other templates as well. For example, perhaps you have a section of a template that could be used in other templates. You can select this section of text, create an inserted template from it, and then use this inserted template in other related templates.

When you create inserted templates by selecting existing template text, the inserted template uses the same component file as the parent template. This means you should save the newly created template to the same folder as the parent template. If you save it to a different folder, a new, empty component file will be created for the template, which means if the new template uses variables from the parent template, you must copy those variables into the new component file. (See [Inserted Templates and Shared Component Files](#) and [Copy Components From One File to Another](#).)

## 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** box.
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.

**Warning:** You cannot place INSERT instructions in template headers or footers. Additionally, you cannot place INSERT instructions in a table.

### Notes:

- You can let the user choose which templates they insert by surrounding INSERT instructions with IF instructions. (See [Condition Text Using True/False Variables](#) and [Condition Text Using True/False 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. (See [Change a Dialog's Options](#).)
- If you're going to allow the user to select several documents for assembly at the same time, make sure there's a section break or page break at the bottom of each template so each assembled document starts on a new page.
- 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.

# Insert an Existing Template

You can create templates that contain the sections of documents you use over and over again—like signature blocks, legal headers, and so forth—and then insert them in your text templates wherever you need them.

**Note:** Templates can be inserted from any location. See [Inserted Template Locations](#) for details.

## 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.
6. Once you have created an INSERT instruction, you can edit it by placing your cursor in the instruction and clicking the  **INSERT Field** button again. (See [Edit an INSERT Template Instruction](#).)

**Warning:** You cannot place INSERT instructions in template headers or footers. Additionally, you cannot place INSERT instructions in a table.

### Notes:

- In some situations, you may want to assemble separate documents but keep all these separate documents contained in single document file. When you insert a template for this purpose, headers and footers specific to each inserted template must remain intact and should not be overwritten by the headers/footers defined for the parent template. See [Use Headers and Footers in Inserted Word Templates](#).
- 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.

# Edit an INSERT Template Instruction

Once you create an INSERT instruction in your template, you can edit it at any time—including making changes to the text of the inserted template.

## To edit an INSERT Template instruction

1. At the template, place the cursor in the INSERT field and click the  **INSERT Field** button. The **Insert 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** to edit the text of the inserted template. (Once the template is open, make any necessary changes. Click the  **Save and Close** button to save and close the inserted template.)

**Warning:** You cannot place INSERT instructions in template headers or footers. Additionally, you cannot place INSERT instructions in a table.

## Inserted Templates and Shared Component Files

Related templates often share many of the same variables. If you point an inserted template's component file to the main template's component file, both templates can share the same variables and other components. On the other hand, if there are not many shared variables between the templates, pointing may not be worthwhile. (See [Use One Component File for Multiple Templates](#) and [Make Templates Stop Sharing Component Files](#).)

When you insert an *existing* template, HotDocs does not automatically point its component file. When HotDocs processes the INSERT instruction during assembly, it uses the inserted template's own component file. This makes sense since the inserted template probably contains variables and other components not found in the component file of the main template. If you want both templates to use the same component file, you must manually point the component file of the inserted template. (Also, if the inserted template already contains variables, dialogs, and so forth, you must copy those existing components into the shared component file).

When you select text and create a *new* template, HotDocs *automatically* points the component file of the new template to the component file of the main template—as long as the template is saved to the same folder as its parent. This ensures the main template and the inserted template will continue to share a common set of variables and other components. If the inserted template is saved in a different folder, a new, empty component file is created. If the inserted template uses variables, you must copy components from the parent template into the new component file. (See [Copy Components From One File to Another](#).)

# Use Headers, Footers, and Margins in Inserted Word Templates

In Microsoft Word, every template has a header and footer, even if no text is defined for it. This is especially important to understand as you are working with headers/footers in inserted templates. The Microsoft Word help file contains a lot of important information about inserting headers and footers in your templates and documents. You should read this information before attempting to use headers and footers, particularly when using them in inserted templates.

When using headers, footers, and margins in your inserted RTF templates, you must first determine the purpose of the inserted template—will the inserted text be inline with that of the parent template, or will it be used to create a set of documents contained within a single document file? Following is a discussion of both situations.

## Inserting “Inline” Documents

An inserted template can merge a section of text so that it is "inline" with the text into which it is being inserted. This is useful if you have an optional paragraph or clause that must be inserted at a specific place within the document.

By default, when you insert a template for this purpose, any headers/footers in the inserted template will be defined by the parent template, and any headers/footers used in the inserted template will be deleted. This is correct.

For instructions on inserting inline text in a template, see [Insert an Existing Template and Make Part of a Template into an Inserted Template](#).

## Inserting Separate Documents (with Separate Headers) in a Single Document File

An inserted template can also merge a section of text as though it were separate from the document into which it is inserted. For example, you may want to assemble separate documents but keep all these separate documents contained in a single document file. When you insert a template for this purpose, headers and footers specific to each inserted template must remain intact and should not be overwritten by the headers/footers defined for the parent template.

When working with headers and footers in inserted templates, you must think of each inserted template as a *section* of the main template. This means that if you want inserted template text to have headers/footers that are different from the parent template, you must define section breaks in your inserted template.

### To use different headers or footers in your inserted templates

1. In the parent template, define the header/footer (or define no header/footer at all).
2. Place your cursor at the place in the parent template where you want the inserted template to be merged and place a next-page section break.
3. Click the  **INSERT Field** button. The **INSERT Field** dialog box appears.
4. Click the  **Open** button next to the **Template to insert** box and locate the template file you are inserting.
5. At the **INSERT Field** dialog box, select either **Keep header of inserted template** or **Keep footer of inserted template**. (Select both if you want the document to maintain both the header and the footer.)
6. Click **OK**.

7. If inserting multiple templates, repeat this process for each inserted template.

Once you define headers/footers in the inserted template, any text that appears in the main template *after* the INSERT instruction will use the headers/footers that were defined in the inserted template.

At times, you may want a cover page for the document that has its own unique header/footer (for example, a notice or disclaimer). For the document to be assembled correctly, you must use a combination of section breaks both in the main template as well as in the inserted templates.

**Warning:** You cannot use the **Keep header** and **Keep footer** options with ASSEMBLE instructions.

### To use a different first-page header in the parent template

1. In the parent template, display the header/footer field.
2. Select the **Different first page** option and specify your header/footer text.
3. Place a continuous section break *before* the first INSERT instruction but after the text of the main template.
4. Place your cursor somewhere *after* the continuous section break and click **Page Setup (File menu)**. The **Page Setup** dialog box appears.
5. Clear the **Different first page** option and apply the change to the current section only (**Preview group**).

## Defining Margins in Inserted Templates

When working with margins in inserted templates, it's once again important to determine the purpose of the inserted template. If you want a portion of text merged as part of the main document, you most likely will want the margin for both sections of text to be the same. However, if your inserted template will produce a separate document within the document file, you may want custom margins.

### To use custom margins in inserted templates

1. At the top of the inserted template, insert either a next-page section break (if the inserted template produces a separate document) or a continuous section break (if the inserted template will be merged inline with the parent template).
2. In the section that contains the text, define your margins.
3. At the bottom of the inserted template, insert another section break.

**Note:** When you select the **Different odd and even** property for headers/footers, this property is assigned for the entire document—not just sections in the document. This means headers/footers in inserted templates must have the same property set in the main template or headers/footers will not be assembled in the main document correctly.

# Start a New Assembly From a Template

You can use the ASSEMBLE instruction to add templates to the Assembly Queue (a dialog box that lists documents waiting to be assembled). Unlike the INSERT instruction, an ASSEMBLE instruction adds the template to the assembly queue and then waits until the main document is finished assembling before it starts assembling the new, added template. (See [Differences Between ASSEMBLE and INSERT Instructions](#).) 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.

**Note:** Templates can be assembled from any location. See [Inserted Template Locations](#) for details.

## To insert an ASSEMBLE instruction in a Word template

1. At the template, position the cursor in the template where you want the ASSEMBLE instruction.
2. 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.
3. Click the **Field type** drop-down button and choose **ASSEMBLE**.
4. Click the  **Browse** button next to the **Template to assemble** box and locate the template you want to assemble.
5. Click **OK**. The instruction is inserted in the template.

## To insert an ASSEMBLE instruction in a WordPerfect template

1. At the template, 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.rtf"»
```
6. Delete the text **INSERT** from the instruction and type the keywords **ASSEMBLE**. For example:

```
«ASSEMBLE "subpoena.rtf"»
```

**Warning:** If the template you are inserting 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 name of the form template file.

## To insert an ASSEMBLE instruction in a form template

1. At the template, create a field and attach a Computation variable to it. (See [Create a Form Field and Attach a Variable to a Field](#).)
2. At the **Computation Editor**, enter the **ASSEMBLE** instruction, including the path and file name (if necessary) of the template you want to assemble. For example:

```
ASSEMBLE "subpoena.hft"
```

**Notes:**

To test an ASSEMBLE instruction, you must exit the template completely and test it from the library.

You can add command-line options to an ASSEMBLE instruction (for example, *ASSEMBLE "subpoena.rtf /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.rtf /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.)

# Using SPAN Instructions

---

## Overview: SPAN Fields

**Note:** 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.

# Allow Users to Edit the Text of an Assembled Document

**Note:** 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.

**Warning:** 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. The **SPAN Field** dialog box appears.
3. Enter a component name in the **Span** box.
4. Optionally, click the  **Edit Component** button and enter a title in the **Title** box. The title will be used to identify the editable text in the **Document Text Editor**.

## Notes:

- 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](#).
- 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 IF, REPEAT, and SPAN Instructions](#) and [Match IF, REPEAT, and SPAN Instructions with END Instructions](#).) You can also define the field color for SPAN instructions at **HotDocs Options**. (See [Define Colors for Fields and Instructions](#) and [Assign Colors to Fields 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

**Note:** 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. The **SPAN Field** dialog box appears.
3. Enter a component name in the **Span** box.
4. Click the  **Edit Component** button and enter a title in the **Title** box. The title will be used to identify the editable text in the **Document Text Editor**.

## Notes:

- To allow users to edit existing paragraph text, see [Allow Users to Edit the Text of an Assembled Document](#).
- 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.
- SPAN instructions in template headers and footers are ignored during assembly.
- 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 Instructions with END Instructions](#).) You can also define the field color for SPAN instructions at **HotDocs Options**. (See [Define Colors for Fields and Instructions](#) and [Assign Colors to Fields 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

---

## Overview: 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.

### Clause

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 will 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 after assembly, add clauses from the library to the document.
- 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.

**Note:** For additional information on the underlying functionality of clauses, see [Understand How Clauses Work](#).

# Understand How Clauses Work

When working with clauses and clause libraries, it is useful to understand the underlying functionality. This includes understanding 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 (.DOT, .RTF, .WPT):** This is the file that contains the text of the clause component. When you create a clause template file (see [Create Clauses at the Template](#), or [Create Clauses at the Clause Library](#)), HotDocs generates an eight-character file name based on the clause component name, appends the appropriate file name extension, and 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. Where you create the clause determines which component file is used. For example, if you create clauses at a template, the clause components are added to the template's component file. But if you create clauses at a clause library, HotDocs creates a new component file that is associated with the clause library and stores the clause components 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 is a library file that 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.

See [Create Clauses at the Clause Library](#) and [Add Existing Clauses to a Clause Library](#).

- **Clause Archive file (.HCL):** This is a file that 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 the clause template files are extracted on an as-needed basis 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.

**Note:** If you are uncertain about a clause's shortened name or file format, you can view the name in the **Clause identifier** box at the **Clause Editor**. (See [Edit an Existing Clause](#).)

# 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.

### Notes:

- To access the clause library toolbar using the keyboard, press **F10**.
- You can also open a clause library either by clicking on the **HotDocs** drop-down menu in the HotDocs toolbar and choosing **Clause Library**, or by right-clicking in the template and choosing **Clause Library** from the shortcut menu. (Word users)

# Create a Clause Library at the HotDocs Library

At a template library, you can create a clause library that contains any number of clauses. These clauses can be assembled and added 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 will be used 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**. The **New Template** dialog box appears.
3. Click the **Type** drop-down button and select a clause library option for the word processor you are using. Your options include **Word RTF Clause Library**, **Word DOT Clause Library**, and **WordPerfect Clause Library**.
4. Type a name for the clause library in the **File name** box.
5. Type a title for the clause library in the **Title** box (or accept the suggestion HotDocs makes).
6. Optionally, type a description in the **Description** box. The description appears when the user views the clause library's properties at the HotDocs template library.
7. Click **OK**. HotDocs creates and opens the clause library.

When the clause library opens, you will notice that it does not contain any clauses yet. You must create the clauses that will appear in the clause list. See [Create Clauses at the Clause Library](#) for details.

When a clause library is created directly at a HotDocs template library, no template file is created. However, a component file (using the same file name as the clause library) is created automatically to store information for components used in the clause library.

**Note:** A clause library can be used to assemble any number of documents and can even be inserted into any template if you want to associate the clauses with a specific document. (See [Insert a Clause Library into a Template](#).)

# 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  **Add**. 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** box.
6. Optionally, type a description for the folder in the **Description** box. 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  **Sort**. (See [Customize a Clause Library](#).)

# 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** box. (50-character limit.)
4. Optionally, enter a title for the clause in the **Title** box. 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. Click **OK**. HotDocs opens a new, empty text template.
7. 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.
8. When you are finished, click the  **Save and Close Template** button at the HotDocs toolbar. The clause is added to the clause library.
9. Repeat this process for all the clauses you need to create.

**Warning:** 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  **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.)

**Note:** 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](#).)

# 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](#)), 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** box, 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** box. 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 edit a clause library

1. Open a clause library for editing. (See [Open and Close a Clause Library](#).)
2. Perform any of the following tasks:

To	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  <b>Arrange</b> 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  <b>New</b> . HotDocs opens the <b>Clause Editor</b> where you can assign properties to a clause component. Once you assign a name and click <b>OK</b> , a text template opens where you can enter your clause text. (See <a href="#">Create Clauses at the Clause Library</a> .)
Add an existing clause to the clause library	Click  <b>Add</b> . HotDocs opens the <b>Add Clause</b> dialog box where you can select an existing clause and add it. (See <a href="#">Add Existing Clauses to a Clause Library</a> .)
Add a folder to the clause library	Click  <b>Add</b> . When the <b>Add Clause</b> dialog box appears, select <b>Folder</b> from the <b>Type</b> drop-down list. (See <a href="#">Add a Folder to a Clause Library</a> .)
Remove a clause or a folder from the clause	Select the clause or folder you want to

library	remove and click  <b>Remove Item</b> .
Make changes to a clause in the clause list	Select the clause and click  <b>Edit</b> . HotDocs opens the <b>Clause Editor</b> where you can change the clause component properties. (To edit the text of the clause, click <b>Edit Clause Text</b> at this dialog box.) (See <a href="#">Edit an Existing Clause</a> .)
Rearrange clauses in the clause list in alphabetical order	Select the folder whose contents you want to arrange and click the  <b>Sort</b> button. HotDocs displays the <b>Folder Sort Options</b> dialog box where you can choose either <b>Ascending</b> or <b>Descending</b> order.
Change the clause or folder title, or add a description	Select a clause or folder and click  <b>Properties</b> . The <b>Item Properties</b> dialog box appears where you can type a new title in the <b>Title</b> box, or type a description in the <b>Description</b> box. Both the title and the description appear in the clause library's <b>Properties</b> tab.
Print a list of the clauses	Click the  <b>Print</b> button. HotDocs prints a list of all the clauses with their associated clause identifiers.
Open the component file for the clause library	Click the  <b>Component Manager</b> button. (See <a href="#">Use Component Manager to Work with Components</a> .)
View the properties of the clause or preview the clause text	Click the <b>Properties</b> tab or the <b>Preview</b> tab, respectively.
Search the list of clause names and clause descriptions for a specific string of text	Select <b>Find</b> 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 <b>Find</b> .

Any time you make changes at the clause library file, those changes are automatically saved.

**Note:** To access the clause library toolbar using the keyboard, press **F10**.

# Create Clauses at the Template

When the text for the clauses you want to create already exists in a single template, it's probably easiest to create individual clauses right in the template instead of at the clause library.

By default, when you create clauses at the template, HotDocs merges an **INSERT Clause** instruction into the template without adding the clause to the clause library. You can however, have HotDocs bypass the **INSERT Clause** instruction and just add the clause to the clause library. Both options are described below.

## 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  **INSERT 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** box. (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** box. 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** box. (See [Create a Prompt for a Variable](#).)
  - If you selected **INSERT clause only IF selected** at the **INSERT Field** box, 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.

## To create a 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** box.
3. Enter a title for the clause in the **Title** box. 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. Click **OK**. The clause is added to the clause list in the library.

**Warning:** 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  **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.)

### Notes:

- To insert an existing clause in the template, position your cursor where you want the clause inserted and

click the  **INSERT 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** box to browse for and select the file. Click **OK** when you are finished.

**Note:** 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.hcl"»`); or it can be inserted using a full file path (`«INSERT "C:\My Documents\ClauseLib.hcl"»`); or it can be inserted using a reference path (`«INSERT "^referencePath\ClauseLib.hcl"»`).

# 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 at the clause library

1. At the clause library, select the clause you want to edit and click  **Edit**. The **Clause Editor** appears. (See [Open and Close a Clause Library](#).)
2. Make any changes to the clause component by editing the **Properties** or the **Resources**.
3. Optionally, click **Edit Clause Text** to edit the text of the clause. (When you are finished, click the  **Save and Close Template** button at the HotDocs toolbar.)

## To edit a clause using Component Manager

1. Either at the template toolbar or the clause library toolbar, click the  **Component Manager** button. The **Component Manager** window appears.
2. Select the clause you want to edit and click the  **Edit Component** button. The **Clause Editor** appears.
3. Make any changes to the clause component by editing the **Properties** or the **Resources**.
4. Optionally, click **Edit Clause Text** to edit the text of the clause. (When you are finished, click the  **Save and Close Template** button at the HotDocs toolbar.)

## To edit a clause at the template

1. Open the template that contains the **INSERT Clause** field. (See [Edit a Template](#).)
2. Insert your cursor in the **INSERT Clause** instruction and click the  **INSERT Field** button. The **INSERT Field** dialog box appears.
3. Make any changes to the clause field.
4. Optionally, click the  **Edit** button to edit the clause component's **Properties** or **Resources**. (Once there, click **Edit Clause Text** to edit the text of the clause. When you are finished, click the  **Save and Close Template** button at the HotDocs toolbar.)

**Warning:** 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  **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.)

**Note:** To edit the clause *field* (or the instruction in the template that inserts the clause), place your cursor in the instruction and click the  **INSERT Field** button. Once there, you can edit the clause component as well as the clause text.

# 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

1. 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 Clauses at the 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** box 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.

**Note:** 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.

# Convert Clauses to New File Formats

## Warnings:

- If you are converting your clauses to RTF, please use Template Manager. See [Convert Templates and Clauses to Microsoft RTF](#).
- Converting clauses requires several complicated steps that, if not followed entirely, may cause unexpected problems. You should always back up your files before converting any templates or clauses. If you'd like, you can contact HotDocs Professional Services to convert your clauses for you.

If you change word processors, you must convert your templates and clause libraries to the new word processor format. (Information on converting a single text template can be found here: [Convert a Single Template to a New File Format](#).)

There are five parts to converting clause formats:

- Extract the clauses from the clause archive and add them to a temporary library.
- Convert the clauses to the new format.
- Replace the old clauses in the clause archive file with the new, converted files.
- Update the properties of the clause components.
- Update the properties of the clause library.

## Part 1: To extract the clauses and add them to a temporary library

1. Locate the HotDocs clause archive (.HCL) file, in the same folder as the clause library file.

**Note:** To locate the clause archive file, select the template that was used to create the clause files (or the clause library if the library has been added to a template library) and then choose **Go To** from the **Template** menu. Windows Explorer opens the folder where the clause files are saved.

2. Use a compression program, such as WinZip or PKZip, to open the clause archive and extract the clause templates to a temporary location, such as C:\temp.
3. Create a new template library (see [Create a Library](#)) and save it to the temporary location.
4. At the temporary library, click the  **Add** button. The **Add Item** dialog box appears.
5. Click the  **Browse** button, locate and select all the clause template files you extracted to the temporary location, and then click **OK**. The **Add Item** dialog box appears again.
6. Click **OK**.
7. HotDocs displays a warning dialog box for each clause template, saying the component files are missing. Click **OK** to ignore this warning. The clauses are added to the library.

Now, you must convert your clause templates to the new word processor format. You use the same process you do when you convert a regular template to a new format.

## Part 2: To convert your clauses to the new format

1. In your temporary library, select a clause and click  **New Template**. The **New Template** dialog box appears.
2. Click the **Type** drop-down button and select the new template file format. HotDocs automatically changes the file name extension in the **File name** box.
3. Type a new title for the clause template in the **Title** box. (HotDocs does not accept backslashes when used in a title.)
4. Do *not* change the contents in the **Other file** box.

5. Click **OK**. HotDocs opens the template.

**Warning:** As HotDocs converts each clause, it creates new, empty component (.CMP) files. The original component file for the clause archive file (located in the template folder) will still be used for the converted clauses, so do not make any HotDocs-related changes to the template, such as attempting to edit or create components. Once the conversion is complete, you can delete these new component files.

6. Once the clause template is converted, save and close the template using the  **Save and Close Template** button.
7. Repeat these steps for each clause you want to convert.

### Part 3: To replace old clauses with newly formatted ones

1. Use a compression program to zip the converted clauses back into a clause archive (.HCL) file. (Create the archive file in the temporary folder you are using. Use the same name as the original clause archive file, including the .HCL file name extension, and add only the converted clauses to the archive.)
2. Locate the original clause archive (.HCL) and rename it to something such as *OldClauses.hcl*. (Once you know the conversion is successful, you can delete this file.)
3. Move the new clause archive (.HCL) file from the temporary folder to the working folder.

Now that you have converted the clauses to the new format, you must update the actual clause component.

### Part 4: To update the properties of the clause component

1. Open the template library that contains the actual clause library.
2. Select the clause library and click  **Edit**. The **Clause Library** window appears.
3. Select a clause and click  **Edit**. The **Clause Editor** appears.
4. In the **Clause identifier** box, replace the old file name extension with the new one and click **OK**.
5. Repeat this process for every clause in the clause library. When you are finished, close the clause library by clicking the **X** in the upper-right corner of the window.

Now that you have updated the clause components, you must update the clause library file.

### Part 5: To update the library properties of the clause library file

1. From the template library, click the  **Open Library** button and locate and open the clause library file. The clause library appears.
2. Select a clause and click the  **Properties** button. The **Item Properties** dialog box appears.
3. Notice that the file name includes the **Clause (/cl)** command-line option. Replace the file name extension with the correct extension and click **OK**.
4. When you are finished, open the original template library and test the clause library (either in the template library or the template). Once you have verified that the conversion was successful, you can remove the files in your temporary folder as well as remove the old clause archive file from the project folder.

**Note:** You may want to review the topic [Understand How Clauses Work](#) to help you understand the different clause and clause library files that are used.

# Using Instruction and Expression Models

## Overview: Instruction and Expression Models

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 instructions and expressions also include placeholders, which you must replace with a value. Possible values include text strings, number amounts, other models, or 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.](#))

This section contains subsections for instruction models as well as expression models. Both include descriptions of each model and examples of how each model can be used. The following table contains a list of these models:

Instruction Models	Expression Models	
ADD TEXT TO MULT_CHOICE	Enter a Date	MIN( NUM, NUM )
ASCEND VAR	Enter a Number	MONTH OF( DATE )
ASK DIALOG	Enter some Text	MONTHS FROM( DATE, DATE )
ASK VAR	Enter True or False	MULT_CHOICE = TEXT
ASSEMBLE "FILENAME"	ABSOLUTE VALUE( NUM )	MULT_CHOICE != TEXT
CLEAR MULT_CHOICE	AGE( DATE )	NOT TRUE_FALSE
CONCEAL VAR	ANSWERED( DIALOG )	OTHER( MULT_CHOICE_VAR )
DEBUG	ANSWERED( VAR )	POSITION( TEXT, TEXT )
DECREMENT NUM_VAR	COUNT( DIALOG )	POWER( NUM, NUM )
DEFAULT VAR TO VALUE	COUNT( MULT_CHOICE_VAR )	REMAINDER( NUM, NUM )
DESCEND VAR	COUNTER	REPLACE( TEXT, TEXT, TEXT, NUM )
ERASE VAR, ERASE DIALOG	DATE - NUM DAYS	RESULT

FILTER COMPUTATION_VAR	DATE - NUM MONTHS	ROUND( NUM, NUM )
FORMAT "LIST_FORMAT"	DATE - NUM YEARS	SELECTION( MULT_CHOICE_VAR, NUM)
GRAY ALL, GRAY VAR	DATE + NUM DAYS	SPACE( TEXT, TEXT )
HIDE ALL, HIDE VAR	DATE + NUM MONTHS	STRIP( TEXT, TEXT, TRUE_FALSE, TRUE_FALSE )
IF EXPRESSION, ELSE IF, ELSE, END IF	DATE + NUM YEARS	SUM( COMPUTATION_VAR )
INCREMENT NUM_VAR	DATE OF( NUM, NUM, NUM)	SUM( NUM_VAR )
INSERT "FILENAME"	DAY OF( DATE )	TEXT CONTAINS TEXT
LANGUAGE "CODE"	DAY OF WEEK( DATE )	TODAY
LIMIT NUM	DAYS FROM( DATE, DATE )	TRUNCATE( NUM, NUM )
OMIT VAR	FIRST( TEXT, NUM )	UNANSWERED
PLAY "MACRO"	FORMAT( VALUE, "EXAMPLE" )	UNION( MULT_CHOICE_VAR, MULT_CHOICE_VAR)
QUIT	INTEGER( TEXT )	VALUE( VAR, EXPRESSION )
REPEAT DIALOG, END REPEAT	LAST( TEXT, NUM )	YEAR OF( DATE )
REQUIRE ALL, REQUIRE VAR	LENGTH( TEXT )	YEARS FROM( DATE, DATE )
SET VAR TO VALUE	MAX( NUM, NUM )	ZERO( NUM_VAR )
SHOW ALL, SHOW VAR	MID( TEXT, NUM, NUM )	
UNGRAY ALL, UNGRAY VAR		
WHILE EXPRESSION, END WHILE		

# 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 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.
- 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** 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, it is suggested 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. For complete instructions, see [Use the Script Editor](#).

# 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 in the **Computation Editor**, the **Script** tab of the **Dialog Editor**, and at the **Expression** box of the **IF Field** dialog box. 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	Meaning
=	<p>The two items in the comparison are of equal value.</p> <p>Example:</p> <pre>Birth Date = 17 Dec 1989 Employee Name = "Louisa Gehrig"</pre>
!=	<p>The two items in the comparison are <i>not</i> of equal value.</p> <p>Example:</p> <pre>IF Exhibit A != TRUE IF Plaintiff Gender != "Male"</pre>
<	<p>The first item in the comparison has a lesser value than the second item.</p> <p>Example:</p> <pre>Account Balance &lt; 9000 COUNTER &lt; 10</pre>
>	<p>The first item in the comparison has a greater value than the second item.</p> <p>Example:</p> <pre>Dependent Age &gt; 18</pre>

<=	<p>The first item in the comparison is less than or equal to the second item.</p> <p>Example:</p> <p>Client Age &lt;= 65</p> <p>COUNTER &lt;= 2</p>
>=	<p>The first item in the comparison is greater than or equal to the second item.</p> <p>Example:</p> <p>Taxed Income &gt;= 75000</p>

Arithmetic Operator	Meaning
+	<p>Add the different components of the script together.</p> <p>Example:</p> <p>Value 1 + Value 2</p> <p>Street Address + ", " + City + ", " + State</p>
-	<p>Subtract the different components of the script from each other.</p> <p>Example:</p> <p>Monthly Income - Amount of Owed Child Support</p>
*	<p>Multiply the different components of the script.</p> <p>Example:</p> <p>Purchase Price * 0.625</p>
/	<p>Divide the different components of the script.</p> <p>Example:</p> <p>Yearly Salary / 12</p>

Logical Operator	Meaning

AND	<p>The statement to the left <i>and</i> the statement to the right must <i>both</i> be true.</p> <p>Example:</p> <p>IF Client is Married AND Client has Children</p>
OR	<p>The statement to the left <i>or</i> the statement to the right must be true.</p> <p>Example:</p> <p>IF Single OR Widowed</p>
NOT	<p>The two items in the comparison must not be equal to each other.</p>

The final operator, the parentheses ( ), instructs HotDocs to perform the operation inside the parentheses first.

# Use Line Breaks, Paragraph Ends, and Tabs in Computation Scripts

**Warning:** 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** box, either manually enter the characters, or use a dot code. Use the following information:

To Insert	Do one of the following
A line break	Press <b>Enter</b> or <b>Shift+Enter</b> . When you do this, HotDocs creates a new line of text (↵) in the same paragraph.  <b>Note:</b> 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  <b>Auto Format</b> button.
	Insert a <b>Line Break</b> dot code («.lb») at the place you want HotDocs to start a new line. (See <a href="#">Insert Characters in Text Strings</a> .)
A paragraph end	Press <b>Ctrl+Enter</b> . When you do this, HotDocs inserts a paragraph mark (¶) and starts a new paragraph of text.
	Insert a <b>Paragraph Mark</b> dot code («.pm») at the place you want HotDocs to start a new paragraph. (See <a href="#">Insert Characters in Text Strings</a> .)
A tab character	Press <b>Ctrl+T</b> .  If you want the <b>Tab</b> key to insert a tab character (instead of you pressing <b>Ctrl+T</b> ), click the  <b>Options</b> button and select <b>Tab key inserts a tab in scripts</b> . Now, whenever you press <b>Tab</b> or <b>Shift+Tab</b> , HotDocs will insert a tab in the script instead of taking you to another field in the dialog box.
	Insert a <b>Tab Character</b> dot code («.tc») at the place you want the text to be tabbed. (See <a href="#">Insert Characters in Text Strings</a> .)

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. At the **Script** or **Expression** box, complete any of the following tasks:

To	Do This
Have HotDocs automatically complete keywords, component names, and Multiple Choice options as you type	Place your cursor in the <b>Script</b> box and press <b>Ctrl+Spacebar</b> . 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 <b>Enter</b> to merge it into your script.  To keep seldom-used keywords out of the auto-complete list, click the  <b>Options</b> button and clear <b>Include seldom-used keywords in auto-complete list</b> .
Access just a list of components	Press the <b>F5</b> key, select the component, and press <b>Enter</b> .
Access just a list of keywords	Press <b>Shift+F5</b> , select the keyword, and press <b>Enter</b> .
Display a syntactical hint of how an instruction or expression should be used	Place your cursor inside the keyword and press the <b>F7</b> 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  <b>Auto Format</b> button.
Undo (or cancel) an action you just performed	Click the  <b>Undo</b> button. HotDocs reverses the change you made.
Redo (or re-implement) an action you just	Click the  <b>Redo</b> button. HotDocs reapplies

performed	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  <b>Cut</b> button or the  <b>Copy</b> 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  <b>Paste</b> button.
Find a specific string of text in the script	Click the  <b>Find</b> button and enter your search text in the <b>Find what</b> box. To find the next instance of the text, click the  <b>Find Next</b> button.  <b>Note:</b> To find only those instances of text that are complete words, select <b>Find whole words only</b> . To find only those instances that have the same capitalization as the text for which you are searching, select <b>Match case</b> .
Find a specific string of text in the script and replace it with another string of text	Click the  <b>Find and Replace</b> button. Enter the search text for which you are searching in the <b>Find what</b> box, and then enter the replacement text in the <b>Replace with</b> box. Once you have entered the required text, click <b>Replace</b> (replaces the current instance of found script with the script you suggest), <b>Replace All</b> (replaces all instances of found script text with the script you suggest), or <b>Find Next</b> (finds the next instance of the script text).  To find and replace script text in a specific block of script, select the block of script first and then click the  <b>Find and Replace</b> button. At the <b>Find and Replace</b> dialog box, select <b>Replace only in selected text</b> .
Move your cursor to a specific location in the script	Click the  <b>Go To</b> button. This displays the <b>Go To</b> 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  <b>Indent</b> button or the  <b>Outdent</b> button. (Click repeatedly to increase or decrease the indent.)

<p>Cause HotDocs to ignore a section of the script when it processes it, or insert a comment in the script</p>	<p>Highlight the block of the script you want to comment and click the  <b>Comment Block</b> 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  <b>Uncomment Block</b> button.</p>
<p>Match an IF or REPEAT instruction with its END IF or END REPEAT instruction (or vice versa)</p>	<p>Place your cursor inside the instruction, right-click, and select <b>Match IF/REPEAT</b> from the shortcut menu. (You can also press <b>Ctrl+M</b>.)</p>
<p>Highlight an entire IF or REPEAT instruction block (meaning everything between a beginning and ending IF/REPEAT instruction)</p>	<p>Place your cursor inside the instruction, right-click, and select <b>Select IF/REPEAT</b> from the shortcut menu. (You can also press <b>Ctrl+Shift+M</b>.)</p>
<p>View helpful information while using the script editor, including accessing help topics for the different instructions and expressions you can use in your script</p>	<p>Click the  <b>HotDocs Help</b> button.</p>
<p>Customize the way the script editor works</p>	<p>Click the  <b>Options</b> button and make your changes. (See <a href="#">Change Script Editing Options</a>.)</p>
<p>Insert line breaks and tab characters in a script</p>	<p>See <a href="#">Use Line Breaks and Tabs in Computation Scripts</a>.</p>
<p>Assign formatting characteristics to literal text strings in a script</p>	<p>Insert the corresponding dot code. See one of the following topics for details:</p> <ul style="list-style-type: none"> <li>■ <a href="#">Change Font Properties of Text</a></li> <li>■ <a href="#">Insert Characters in Text Strings</a></li> <li>■ <a href="#">Add Punctuation and Capitalization to Sentences</a></li> </ul>

**Notes:**

- 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 [Overview: Instruction and Expression Models](#).

# Instruction Models

---

## ADD TEXT TO MULT\_CHOICE; CLEAR MULT\_CHOICE

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 MC
REPEAT Committee Members DI
ADD Name TE TO Committee Chairperson MC
END REPEAT
```

In this example, the CLEAR instruction first removes any existing options from the Multiple Choice variable *Committee Chairperson MC*. Then, the script repeats the *Committee Members DI* 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 MC* 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 MC
ADD "Single|Client is single" TO Marital Status MC
ADD "Married|Client is married" TO Marital Status MC
ADD "Divorced|Client is divorced" TO Marital Status MC
```

In this example, the Multiple Choice variable *Marital Status MC* 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.

## 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 *First Name TE* first, it sorts by *Last Name TE*:

```
""  
  
REPEAT Client Information DI  
  
ASCEND Last Name TE  
  
RESULT + First Name TE + " " + Last Name TE + "  
  
"  
  
END REPEAT
```

The first empty set of quotation marks sets the computation value to nothing. Then, the repeated dialog, *Client Information DI*, 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 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 DI* dialog if the user is a first-time buyer:

```
«IF First Time Buyer TF»  
«ASK Buyer Information DI»  
«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 *Current Insurance TE* variable is asked:

```
IF MONTHS FROM( Date Of Previous Filing DA , TODAY ) >= 13
ASK Current Insurance TE
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 Within 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 TF»  
  
«ASSEMBLE "Cover Letter.rtf"»  
  
«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.

**Note:** You can add command-line options to an ASSEMBLE instruction (for example, *ASSEMBLE "subpoena.rtf/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.rtf /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.)

# CONCEAL VAR

Placeholder	Replace With
VAR	A variable used in an answer source (pick list)

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 TE* and a *Creditor Address TE*. However, in a second template that uses the same dialog and answer source, *Creditor Address TE* 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 TE
```

Answers for *Creditor Address TE* 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 TE* in this script as well, which would keep *Creditor Address TE* 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:

- [Overview: 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 TE* is defaulted to the literal value of *Sam Jones*:

```
DEFAULT Attorney Name TE TO "Sam Jones"
```

In this example, when HotDocs processes this script, it first determines whether *Attorney Name TE* has been answered. If it has, the DEFAULT instruction has no effect and HotDocs uses the answer already given. If *Attorney Name TE* 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 TE TO Client Name TE
```

In this example, when HotDocs processes this script, it first determines whether *Trustee Name TE* has been answered. If it has, the DEFAULT instruction has no effect and HotDocs uses the answer already given. If *Trustee Name TE* has not been answered, HotDocs then checks to see what answer has been given for *Client Name TE* and suggests that as the answer. It is important to note, however, that if *Client Name TE* is unanswered, *Trustee Name TE* will likewise be unanswered.

**Warning:** Do not use the DEFAULT instruction in the script of a repeating 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.

### Notes:

- The DEFAULT and SET instructions both assign answers to variables. See [Differences Between SET and DEFAULT Instructions](#) 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 <i>true</i> or <i>false</i> 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 DI  
  
FORMAT "A, B, and C"  
  
FILTER No Corporate Vendors CO  
  
RESULT + Vendor Name TE  
  
END REPEAT
```

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.

### Notes:

- 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, b, and c</i> or <i>a; b; or 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 DI  
FORMAT "A, B, and C"  
RESULT + Degree TE  
  
END REPEAT  
  
RESULT
```

Once a user has provided the list of educational degrees in *Education Information DI*, the FORMAT instruction ensures that the requisite commas are in place.

**Note:** FORMAT instructions should be placed immediately after the REPEAT instructions for repeated dialogs.

## 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 MC*, controls which questions are asked:

```
GRAY ALL  
  
UNGRAY Family Status MC  
  
IF Family Status MC = "Married, with children"  
UNGRAY ALL  
  
ELSE IF Family Status MC = "Married, no children"  
UNGRAY Spouse Name TE  
  
ELSE IF Family Status MC = "Separated, with children"  
UNGRAY Number Of Children NU  
  
END IF
```

The initial GRAY ALL instruction dims all variables in the dialog, making them inactive. Then the *Family Status MC* 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.

# HIDE ALL, HIDE VAR SHOW ALL, SHOW VAR

Placeholder	Replace With
VAR	A variable used in an answer source (pick list)

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 MC* 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 MC

IF Document Type MC = "Will"

SHOW Executor Name TE

SHOW Testator Name TE

ELSE IF Document Type MC = "Trust"

SHOW Trustee Name TE

SHOW Grantor Name TE

END IF
```

**Warning:** 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 <i>true</i> or <i>false</i>

You can make parts of templates conditional by using IF instructions. A conditional part of a template 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, telling HotDocs 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 TF»
```

```
In addition, «Employee Name TE» shall be allowed «Number of  
Vacation Days NU:ten» for vacation time. «Employee Name TE» 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 Status MC = "Complete"  
"Upon finishing the project..."  
ELSE IF Status MC = "In Process"  
"While working on the project..."  
ELSE  
"Before working on the project..."  
END IF
```

### Notes:

- 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.

- Please see [Overview: 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 NU TO 1
SET Prospect Count NU TO 0
WHILE ANSWERED(Applicant Name TE[Applicant Count NU])
IF Applicant Years of Schooling NU[Applicant Count NU] >= 4
INCREMENT Prospect Count NU
SET Prospect Name TE[Prospect Count NU] TO Applicant Name
TE[Applicant Count NU]
END IF
INCREMENT Applicant Count NU
END WHILE
```

# INSERT "FILENAME"

Placeholder	Replace With
FILENAME	The file name and folder path of another HotDocs template

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.

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 will include sections that are used frequently, 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 TF»  
  
«INSERT "Attorney Information Block.rtf"»  
  
«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.

## Notes:

- For detailed information on inserting templates, see [Overview: Insert Templates into Templates](#).
- Templates can be inserted from any location, as long as you specify the correct folder path information. (See [Inserted Template Locations](#) for examples.)
- Inserted files 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. See [Use Headers and Footers in Inserted Word Templates](#).

# LANGUAGE CODE

Placeholder	Replace With
CODE	Any of the following language codes: <b>ENG</b> (English) <b>DEU</b> (German) <b>DES</b> (Swiss German) <b>DEA</b> (Austrian German) <b>FRA</b> (French) <b>NLD</b> (Dutch) <b>ESN</b> (Spanish) <b>ITA</b> (Italian)

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 DA: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»
```

## Notes:

- For more detailed information on using foreign language templates, see [Create a Foreign Language Template](#).
- You must obtain the appropriate foreign language DLL file before the LANGUAGE instruction will work in your templates. Contact your HotDocs sales representative for more information.

# 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 DI*, 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
```

## Notes:

- 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** box. (Make sure **Spreadsheet** is selected as the **Style**.)

# OMIT VAR

Placeholder	Replace With
VAR	A variable used in an answer source (pick list)

This 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 spreadsheet. OMIT 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 NU
```

The variable would be asked on the dialog, however, so the user could answer it.

See [Suggest an Answer Source for a Dialog](#) 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 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.

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.rtf"»  
  
«PLAY "Standard_Format"»
```

### To insert a PLAY instruction in a template

1. At the template, position the cursor in the template.
2. Click the **HotDocs** menu in the HotDocs toolbar and choose **Other Field** from the list of options. The **Other Field** dialog box appears. (WordPerfect users: If you're automating a WordPerfect template, you insert a PLAY instruction by copying and pasting an existing variable and then replacing the text between the chevrons (« ») with the PLAY instruction.)
3. Click the **Field type** drop-down button and choose **PLAY**.
4. In the **Macro name** box, enter the name of the macro you want to run.
5. Click **OK**. The instruction is inserted in the template.

See [Specify a Template for Storing Post-Assembly Macros](#) for more information.

**Note:** 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

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 QUIT instruction allows you to close the variable without losing the work you have done. It 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 MC = " ____ "  
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  **Uncomment Block** button.)

**Warning:** 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 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 DI»  
  
Editor: «Editor First Name TE» «Editor Last Name TE»  
  
«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 DI  
  
RESULT + "Editor: " + Editor First Name TE + " " + Editor Last  
Name TE + "  
  
"  
  
REPEAT Book Information DI  
  
RESULT + "Book Title: " + Book Title TE + "  
  
" + "Author: " + Author First Name TE + " " + Author Last Name TE +  
"  
  
"  
  
END REPEAT  
  
END REPEAT
```

You can create up to three levels of sublists by nesting REPEAT instructions. (See [Overview: Create Lists Within a List](#).)

### Notes:

- For more information about creating lists of answers, see [Overview: Include Lists in Your Documents](#).
- 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  **Match Fields** button.

# REQUIRE ALL REQUIRE VAR

Placeholder	Replace With
VAR	Any variable used in a dialog

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 DA
```

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.

## Notes:

- 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](#).

## 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 [Create a SET Instruction](#).

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 TF
SET Client Name TE TO Plaintiff Name TE
END IF
```

**Warning:** 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.)

### Notes:

- 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"*)

# WHILE EXPRESSION

## END WHILE

Placeholder	Replace With
EXPRESSION	<p>An expression that results in <i>true</i> or <i>false</i></p> <p><b>Note:</b> A True/False expression can be as complicated as it needs to be, but it must result in either <i>true</i> or <i>false</i>. For example, the expression <b>YEARS FROM( Child's Birth Date, TODAY )</b> produces a number (the age of a person), not a true or false value—it is not a True/False expression. But the expression <b>YEARS FROM( Child's Birth Date, TODAY ) &gt; 17</b> can only result in <i>true</i> or <i>false</i>. It is a True/False expression.</p>

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 NU TO 0

REPEAT Beneficiary Information DI
  INCREMENT Signer Count NU
  SET Signer Name TE[Signer Count NU] TO Beneficiary Name TE
END REPEAT

REPEAT Fiduciary Information DI
  SET Lookup NU TO 1
  WHILE Lookup NU <= Signer Count NU AND Fiduciary Name TE != Signer
  Name TE[Lookup NU]

```

```

INCREMENT Lookup NU
END WHILE
IF Lookup NU > Signer Count NU
INCREMENT Signer Count NU
SET Signer Name TE[Signer Count NU] TO Fiduciary Name TE
END IF
END REPEAT

```

In the first part of this script, the *Beneficiary Information DI* dialog is repeated, and as answers are entered, their values are set to be used for *Signer Name TE* (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 DI* 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 DI* dialog. (When you insert the REPEAT instruction for the *Signer Information DI* 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 NU TO 1
WHILE Count Index NU <= LENGTH(Account Number TE)
IF MID(Account Number TE, Count Index NU, 1) = " "
SET Account Number TE TO FIRST(Account Number TE, Count Index NU -
1) + LAST(Account Number TE, LENGTH(Account Number TE) - Count
Index NU)
ELSE
INCREMENT Count Index NU
END IF
END WHILE

```

This script uses a temporary counter (*Count Index NU*) 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.

# Expression Models

---

## 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** box. 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** box. 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** box. 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** box. 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	Replace With
NUM	A number value for which you want the absolute value returned. Can be a Number variable or a fixed 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 NU )
```

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 NU < 0»
```

```
Your account is $«Absolute Value of Final Balance CO» overdrawn.
```

```
«ELSE»
```

```
You have $«Absolute Value of Final Balance CO» in your account.
```

```
«END IF»
```

## AGE( DATE )

Placeholder	Replace With
DATE	A date value, which you want to check against the current date. This can be a Date variable or a fixed date 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 DA )
```

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 DA ) < 18
ASK Parent/Guardian Information DI
END IF
```

## ANSWERED( DIALOG )

Placeholder	Replace With
DIALOG	A dialog name

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 DI*). From within this dialog, a user could open an inserted dialog that asks questions about the decedent's assets (*Decedent's Assets DI*). Later in the template, you could ask additional questions based on whether these dialogs have been answered:

```
IF ANSWERED( Decedent Information DI ) AND ANSWERED( Decedent's
Assets DI)

ASK Asset Distribution Information DI

END IF
```

## ANSWERED( VAR )

Placeholder	Replace With
VAR	A variable

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*.

**Warning:** 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 (*Address 2 TE*) in a custom dialog; however, not all users will provide information for that variable. In the template text, you can surround *Address 2 TE* with an IF expression that merges that variable into the document—only if the user answers it:

```
«Name TE»  
«Address 1 TE»  
«IF ANSWERED( Address 2 TE )»  
«Address 2 TE»  
«END IF»  
«City TE», «State MC» «ZIP TE»
```

In this script, the ANSWERED expression is used with an IF expression to insert or remove *Address 2 TE*, based on whether the user has provided that information. (Without it, the default unanswered text, *\*\*\*Address 2 TE\*\*\**, would be inserted in the assembled document.)

**Note:** 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**. (See [Control How HotDocs Processes a Variable](#).)

## COUNT( DIALOG )

Placeholder	Replace With
DIALOG	A dialog name

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.

In the following example, HotDocs determines if there is more than one fiduciary. If there is, it adds certain prefixes and plural abbreviations to the fiduciary title so the paragraph is structured correctly.

```
I appoint «REPEAT Fiduciary Information DI:a, b, and c»«Fiduciary Name TE:LIKE THIS»«END REPEAT» as «IF COUNT( Fiduciary Information DI ) > 1»Co-«Fiduciary Title TE»s«ELSE»«Fiduciary Title TE»«END IF».
```

This expression uses the COUNT instruction to determine if *Fiduciary Information* is answered more than once. If it is, HotDocs inserts the *Co-* prefix before *Fiduciary Title TE*, and then inserts the plural *s* at the end of the variable. For example, in the assembled document, the answer may be inserted as *Co-Executors* or *Co-Personal Representatives*.

**Note:** 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	Replace With
MULT_CHOICE_VAR	A Multiple Choice variable with the <b>Select</b> option set to <b>All That Apply</b>

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 TE ) = 1
"city"
ELSE IF COUNT( Publicity Tour City TE ) > 1
"cities"
END IF
```

# 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.

For example, a user may want to create a word processor table that contains a numbered list of clients:

```
«REPEAT Client Information DI»  
  «COUNTER». «Client Name TE»  
«END REPEAT»
```

In the example above, *Client Information DI* repeats the *Client Name TE* 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 DI  
ASCEND Child Birth Date DA  
IF COUNTER = COUNT( Children DI )  
  RESULT + "The youngest child is " + Child Name TE  
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.

**Note:** 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

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 DA - 14 DAYS

## DATE - NUM MONTHS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number 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 DA - 4 MONTHS
```

## DATE - NUM YEARS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number 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 DA - 2 YEARS
```

## DATE + NUM DAYS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number 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 DA*:

```
Purchase Date DA + 90 DAYS
```

## DATE + NUM MONTHS

Placeholder	Replace With
DATE	A date value, such as a Date variable
NUM	A number 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

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 DA + 30 YEARS
```

## DATE OF( NUM, NUM, NUM )

Placeholder	Replace With
NUM	A number value representing the day of the month (from 1 to 31)
NUM	A number value representing the month (from 1 to 12)
NUM	A number value representing the year (should be four digits)

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 DA <= DATE OF ( 15, 8, YEAR OF( TODAY ) )
Employee Name TE + " 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	Replace With
DATE	A date 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 DA ) = 1
SET Start Accruing Date DA TO Hire Date DA
ELSE
SET Start Accruing Date DA TO DATE OF( 1, MONTH OF( Hire Date DA +
1 MONTHS ), YEAR OF( Hire Date DA + 1 MONTHS ) )
END IF
```

## DAY OF WEEK( DATE )

Placeholder	Replace With
DATE	A Date variable

This expression determines on which day of the week a specific date falls and converts that value to an integer. 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 DA ) = 7
Payment Date DA + 2 DAYS
ELSE IF DAY OF WEEK( Payment Date DA ) = 1
Payment Date DA + 1 DAYS
ELSE
Payment Date DA
END IF
```

## DAYS FROM( DATE, DATE )

Placeholder	Replace With
DATE	A date value, such as a Date variable
DATE	A date value, such as a Date variable. (These can be in any order.)

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 missed the payment deadline, HotDocs merges a warning into the document:

```
IF DAYS FROM( Purchase Date DA, 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.

## FIRST( TEXT, NUM )

Placeholder	Replace With
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	A number value, such as a Number variable or a fixed number value. It specifies the number of characters you want returned.

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 TE, 1 ) + FIRST( Client Middle Name TE, 1 )  
+ FIRST( Client Last Name TE, 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( Last Name, 4 ) + Case Number
```

## FORMAT( VALUE, "EXAMPLE" )

Placeholder	Replace With
VALUE	Any Number, Date, or True/False variable
"EXAMPLE"	A format example (in quotation marks) you want used with the value. Must be in a format HotDocs can recognize.

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 Items Purchased DI  
  
RESULT + Item TE + ", " + FORMAT( Amount NU, "$9,999.00" ) + "  
  
"  
  
END REPEAT
```

In this script, HotDocs first sets the computation value to *nothing*. Then HotDocs repeats *Items Purchased DI* and then places the answers for both *Item TE* and *Amount NU* (which is formatted to appear as a text value) in the same text string, separated by a comma. If *Items Purchased DI* is answered more than once, HotDocs manually inserts a hard return (as shown before the END REPEAT) to create a column of amounts.

## INTEGER( TEXT )

Placeholder	Replace With
TEXT	A text value, such as a Text variable

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 TE*, must first be converted to an integer before it can be used in the comparison:

```
IF Call Time TE CONTAINS "p"

INTEGER( Call Time TE ) + ( INTEGER( MID( Call Time TE, 1 +
POSITION( Call Time TE, ":" ), 2 ) ) /60 ) > 5.5

ELSE

INTEGER( Call Time TE ) + ( INTEGER( MID( Call Time TE, 1 +
POSITION( Call Time TE, ":" ), 2 ) ) /60 ) > 17.5

END IF
```

In this script, *Call Time TE* 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 TE* 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 non-afternoon time value—that is, one that is either in 24-hour format or in the morning (A.M.).

## LAST( TEXT, NUM )

Placeholder	Replace With
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	A number value, such as a Number variable or a fixed number value. It specifies the number of characters you want returned.

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 TE, 4 )
```

In the next example, the user wants to make the answer to *Item* 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 TE, 1 ) = "s"  
OR LAST ( Item TE, 1 ) = "z"  
Item TE + "es"  
ELSE Item TE + "s"  
END IF
```

**Note:** 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	Replace With
TEXT	A text value, such as a Text variable. HotDocs counts the characters in this value and assigns a numeric 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 TE ) <= 150»
```

```
The novel's plot description is as follows: «Plot Description TE»
```

```
«ELSE»
```

```
The novel's plot description is as follows:
```

```
«Plot Description TE»
```

```
«END IF»
```

## MAX( NUM, NUM )

Placeholder	Replace With
NUM	A number value, such as a Number variable
NUM	A number value, such as a Number variable (These values can be in any order.)

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 NU* or the *Monthly Expenses NU*, depending on which is the greater value:

```
MAX( Monthly Salary NU, Monthly Expenses NU )
```

## MID( TEXT, NUM, NUM )

Placeholder	Replace With
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	A number value, such as a Number variable or a number you type. This number specifies where HotDocs starts “returning” characters.
NUM	A number value, such as a Number variable or a number you type. It specifies the number of characters you want returned.

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 TE* (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 TE, 2, 3 )
```

```
MID( Telephone Number TE, 7, 3 )
```

```
LAST( Telephone Number TE, 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	Replace With
NUM	A number value, such as a Number variable
NUM	A number value, such as a Number variable. (These values can be in any order.)

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 NU* and *Labor Costs NU*, and returns the lesser of the two values:

```
MIN( Shipping Costs NU, Labor Costs NU )
```

## MONTH OF( DATE )

Placeholder	Replace With
DATE	A date value such as a Date variable

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 DA + 3 MONTHS ), YEAR OF ( Hire  
Date DA + 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	Replace With
DATE	A date value, such as a Date variable
DATE	A date value, such as a Date variable. (These values can be in any order.)

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 DA, 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.

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 MC = "Credit Card"
  ASK Credit Card Information DI
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 MC != "Credit Card"
  INSERT "Alt payment method.RTF"
END IF
```

**Note:** 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.

## 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> .

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 green card information:

```
«IF NOT US Citizen TF»  
«ASK Visa Information DI»  
«END IF»
```

## OTHER( MULT\_CHOICE\_VAR )

Placeholder	Replace With
MULT_CHOICE	A Multiple Choice variable that has either the <i>Other</i> option specified, or the <i>None of the Above</i> option specified

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 MC = OTHER( Lending Agent MC )
  ASK Lending Agent City TE
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 MC) = "None of the Above"
  "No representative selected"
ELSE
  FORMAT (Company Representative MC, "a, b, and c")
END IF
```

## POSITION( TEXT, TEXT )

Placeholder	Replace With
TEXT	A text value, such as a Text variable
TEXT	The character or character string for which you want to search

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 TE*, and returns a number value, representing its numeric position in that given text string.

```
POSITION( Case Number TE, "-" )
```

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 TE CONTAINS "p"  
  
INTEGER( Call Time TE ) + ( INTEGER( MID( Call Time TE, 1 +  
POSITION( Call Time TE, ":" ), 2 ) ) /60 ) > 5.5  
  
ELSE  
  
INTEGER( Call Time TE ) + ( INTEGER( MID( Call Time TE, 1 +  
POSITION( Call Time TE, ":" ), 2 ) ) /60 ) > 17.5  
  
END IF
```

This script first determines if the value of *Call Time TE* 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 non-afternoon time value—that is, one that is either in 24-hour format or in the morning (A.M.).

## REMAINDER( NUM, NUM )

Placeholder	Replace With
NUM	A number value, such as a Number variable, to be divided (a numerator)
NUM	A number value, such as a Number variable, by which to divide (a denominator)

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.

## POWER( NUM, NUM )

Placeholder	Replace With
NUM	A number value, such as a Number variable, to be raised to a power
NUM	The exponent (or number that indicates the operation of repeated multiplication)

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*.

## REPLACE( TEXT, TEXT, TEXT, NUM )

Placeholder	Replace With
TEXT	The name of a Text variable or the string of characters you need to search
TEXT	<p>The text for which you are searching</p> <p>You can search for the following special characters: (Make sure you include the backslash.)</p> <ul style="list-style-type: none"> <li>\\       backslash character</li> <li>\h       non-breaking hyphen</li> <li>\l       line break</li> <li>\p       paragraph mark</li> <li>\s       non-breaking space</li> <li>\t       tab</li> </ul>
TEXT	<p>The text you want to use as a replacement</p> <p>You can search for the following special characters: (Make sure you include the backslash.)</p> <ul style="list-style-type: none"> <li>\\       backslash character</li> <li>\h       non-breaking hyphen</li> <li>\l       line break</li> <li>\p       paragraph mark</li> <li>\s       non-breaking space</li> <li>\t       tab</li> </ul>
NUM (optional)	<p>Controls the number of times the character is replaced</p> <p>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.</p>

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 TE*, replaces them with a comma and space, and then SETs that result to the Text variable *Single-Line Address TE*:

```
SET Single-Line Address TE TO REPLACE(Multi-Line Address TE, "\l",
", ")
```

## 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.

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 DI  
  
RESULT + Editor First Name TE + " " + Editor Last Name TE + ",  
Editor" + "  
  
"  
  
END REPEAT
```

In this computation, the RESULT expression returns the value of *Editor First Name TE* and adds it to the value of *Editor Last Name TE*. 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	Replace With
NUM	A number value, such as a Number variable, to be rounded
NUM	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

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	Replace With
MULT_CHOICE	A Multiple Choice variable
NUM	A number value that indicates which selected option to return

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 DI*) that asks which employees are supposed to work on a given assignment (using the Multiple Choice variable, *Employee Names MC*). As HotDocs repeats this list, the UNION expression adds each selected, original name from each repetition to a new Multiple Choice variable, *Unique List MC*

Once all of the unique answers have been added to *Unique List MC*, HotDocs then uses the SELECTION expression to retrieve each individual answer from *Unique List MC*. The result of the script merges these names in a report.

```
ERASE Unique List MC (Tip: Unique List MC and Employee Names MC are
both Multiple Choice variables that contain the same options.)

REPEAT Employee List DI

    SET Unique List MC TO UNION(Unique List MC, Employee Names MC)

END REPEAT

ERASE Project Participant MC

SET Index NU TO 1

WHILE SELECTION(Unique List MC, Index NU) != ""

    SET Project Participant TE[Index NU] TO SELECTION(Unique List MC,
Index NU)

    INCREMENT Index NU

END WHILE
```

## SPACE( TEXT, TEXT )

Placeholder	Replace With
TEXT	A Text variable or other expression that produces a text result
TEXT (optional)	<p>A character or text string that can be used in place of the space character</p> <p>You can replace the space character with any alphanumeric character, including any of the following special characters: (Make sure you include the backslash.)</p> <ul style="list-style-type: none"><li>\\       backslash character</li><li>\h       non-breaking hyphen</li><li>\l       line break</li><li>\p       paragraph mark</li><li>\s       non-breaking space</li><li>\t       tab</li></ul>

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 TE + " " + SPACE(Client Middle Name TE) + Client  
Last Name TE
```

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 TE, "\l") +  
SPACE(Address Line 2 TE, "\l") +  
SPACE(City TE, ",") + SPACE(State TE) + SPACE(Zip Code TE)
```

## STRIP( TEXT, TEXT, TRUE\_FALSE, TRUE\_FALSE )

Placeholder	Replace With
TEXT	The name of a Text variable, or the string of characters you need to search
TEXT	<p>The character or string of characters for which you want to search</p> <p>You can strip any alphanumeric characters from a text string, including the following special characters: (Make sure you include the backslash.)</p> <p>\\       backslash character</p> <p>\h       non-breaking hyphen</p> <p>\l       line break</p> <p>\p       paragraph mark</p> <p>\s       non-breaking space</p> <p>\t       tab</p>
TRUE_FALSE (optional)	The value of <i>TRUE</i> if characters should be stripped from just the beginning of the text
TRUE_FALSE (optional)	The value of <i>TRUE</i> if characters should be stripped from just the end of the text

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 Description TE TO STRIP(Description TE, " !.,?", 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*).

## SUM( COMPUTATION\_VAR )

Placeholder	Replace With
COMPUTATION_VAR	A repeated Computation variable

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 TE*, *Item Amount NU*, and *Item Quantity NU*. For each line item, you create a fourth variable, a computation called *Total Amount CO* that multiplies *Item Amount NU* by the number of items the user purchases (or *Item Quantity NU*). You can then add all of the *Total Amount CO* values and receive one sum total:

```
SUM( Total Amount CO )
```

## SUM( NUM\_VAR )

Placeholder	Replace With
NUM_VAR	A repeated Number variable

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 NU*:

```
SUM( Monthly Payment NU )
```

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 NU ) > ( Monthly Income NU * 0.36 )
```

## TEXT CONTAINS TEXT

Placeholder	Replace With
TEXT	A text value, such as a Text variable
TEXT	A text value, such as a Text variable. Any text you type must be in quotation marks.

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 TE* 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 TE

"The Commonwealth of «State Name TE»"

ELSE

"The State of «State Name TE»"

END IF
```

**Note:** 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* 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.

# TODAY

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 DA, TODAY )
```

**Note:** You also can create a Date variable called TODAY. See [Create a Date Variable that Inserts the Current Date](#).

## TRUNCATE( NUM, NUM )

Placeholder	Replace With
NUM	A number value, such as a Number variable
NUM	A number value specifying the number of places (0-7) to the right of the decimal point at which to truncate the number

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 NU* 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 NU = TRUNCATE( Rent NU, 0 )  
  FORMAT( Rent NU, "NINE DOLLARS" )  
ELSE  
  FORMAT( Rent NU, "NINE DOLLARS AND TWELVE CENTS" )  
END IF
```

Specifically, this computation compares the actual value of *Rent NU* 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.

**Note:** 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 NU 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.

**Note:** 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\_VAR, MULT\_CHOICE\_VAR)

Placeholder	Replace With
MULT_CHOICE	A Multiple Choice variable
MULT_CHOICE	A Multiple Choice variable

This expression creates a single list of all unique options (answers) that have been selected across two or more 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 DI*) that asks which employees are supposed to work on a given assignment (using the Multiple Choice variable, *Employee Names MC*). As HotDocs repeats this list, the UNION expression adds each selected, original name from each repetition to a new Multiple Choice variable, *Unique List MC*.

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 MC*. The result of the script merges these names in a report.

```
ERASE Unique List MC (Tip: Unique List MC and Employee Names MC are  
both Multiple Choice variables that contain the same options.)
```

```
REPEAT Employee List DI
```

```
SET Unique List MC TO UNION(Unique List MC, Employee Names MC)
```

```
END REPEAT
```

```
ERASE Project Participant MC
```

```
SET Index NU TO 1
```

```
WHILE SELECTION(Unique List MC, Index NU) != ""
```

```
SET Project Participant TE[Index NU] TO SELECTION(Unique List MC,  
Index NU)
```

```
INCREMENT Index NU
```

```
END WHILE
```

## VALUE( VAR, EXPRESSION )

Placeholder	Replace With
VAR	The name of a variable
EXPRESSION (optional)	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 <b>0</b> for Number variables <b>1 JAN 1800</b> for Date variables <b>FALSE</b> for True/False variables "" for Multiple Choice variables

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 NU*, 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 NU + Product Cost NU + VALUE(Sales Tax NU)

## YEAR OF( DATE )

Placeholder	Replace With
DATE	A date value, such as a Date variable

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( Date of Hire DA + 3 MONTHS ), YEAR OF( Date  
of Hire DA + 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	Replace With
DATE	A date value, such as a Date variable
DATE	A date value, such as a Date variable (These values can be in any order.)

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 DA, Birth Date DA )
```

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 DA, Retirement Date DA ) >= 25»
```

```
We would like to honor you at a retirement party on «Party Date DA»  
at «Party Time TE» at «Party Location TE».
```

```
«END IF»
```

## ZERO( NUM\_VAR )

Placeholder	Replace With
NUM	A Number variable

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 NU* to return 0 if the Number variable *Shipping Charge NU* is unanswered.

Total Price NU + ZERO( Shipping Charge NU )

**Note:** You can suggest default answers for unanswered variables of other types. See [VALUE\( VAR, EXPRESSION \)](#) for details.

# Testing Automation

---

## Overview: 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. Additionally, if your templates are performing sluggishly, you can examine interview and dialog update counts to determine where your scripts can be simplified and performance improved. 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](#).)
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 Assemble a Document](#).)
7. Click **Close** (**File** menu) to close the test assembly window.

### Notes:

- 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 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. (See [Customize a Computation Variable](#) and [Use the Script Editor](#).)
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. Optionally, 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**. (See [Simultaneously Edit a Template and Assemble a Document](#).)
5. Click **Close** (**File** menu) to close the test assembly window.

### Notes:

- You can save your test answers in a test answer file. See [Use 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 will allow you to save the script. (See [Resolve Syntax Errors in a Template or 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 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.

### Notes:

- 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  **Test 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  **Test 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.

### Notes:

- 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 [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  **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** box, type **Test Answer File**.
5. In the **Title** box, 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 as you edit the underlying template

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.

**Note:** 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.

# 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, REPEAT, and SPAN instructions used in the template or a script must have corresponding END IF, END REPEAT, and END SPAN 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 Instructions with END Instructions](#).)
- **Syntax errors in a computation, dialog 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.)

**Note:** 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. (See [Understand the HotDocs Scripting Language](#) for information on writing scripts.)

# 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

1. At the template, click the  **Go to Field** button. The **Go To Field** dialog box appears.
2. Enter a number in the **Field number** box.
3. Click **OK**. Your cursor is taken to that field.

## Overview: 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 feature with both text and form templates.

## To view variable usage

1. Test assemble the document. (See [Test Assemble a Text or Form 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, double-click the variable.

# 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.

## 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. Warnings can include the following:

Warning	What It Means
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.  <b>Example:</b>  An example of this would be the following script:  <code>SET TF Variable TO TRUE</code>

	<p>... (later in the template)</p> <p>ASK Dialog</p> <p>(where <i>Dialog</i> uses the variable, <i>TF Variable</i>.)</p> <p>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.</p> <p>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.</p>
<p>Variables that are asked more than once in the same interview</p>	<p>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.</p>
<p>Variables that are referred to in a dialog script and are set to a value later</p>	<p>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.</p> <p><b>Example:</b></p> <p>An example of this would be the following script:</p> <pre>SET Client TO "Husband" ASK Husband Information SET Client TO "Wife" ASK Wife Information</pre> <p>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:</p> <p>The following information applies to the «Husband or Wife».</p> <p><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>.</p>

	<p>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.</p>
<p>Variables that are not asked when referred to but are asked later</p>	<p>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 <b>Ask automatically</b> 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.</p>

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.

**Note:** If you receive an error that includes a field number, you can quickly go to that specific field in the template by clicking the  **Go To Field** button in the HotDocs Edit toolbar. See [Go to a Specific Field in the Template](#) for details.

# Move From Document to Template During Testing

**Warning:** 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.

**Warning:** 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 Text or Form 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.

**Warning:** 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.)

# Use the HotDocs Debugger

---

## Overview: 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 or scripts in Debug mode. You do this by inserting a DEBUG instruction in the template or script you think may be causing the unexpected result. Then, you test the template or script 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 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.

# Insert Debugging Instructions in Templates and Scripts

To debug your template or script, you must first insert a DEBUG instruction in the template or script. Then, you must test assemble the template (or test the 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.

## To insert a DEBUG instruction in a text template

1. Open the text template for editing. (See [Edit a Template.](#))
2. In the template, place your cursor where you want to begin debugging.
3. 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: If you're automating a WordPerfect template, you insert a DEBUG instruction by creating a variable (or by copying and pasting an existing variable) and then replacing the text between the chevrons (« ») with the DEBUG instruction, for example, «DEBUG».)
4. Click the **Field type** drop-down button and choose **DEBUG**.
5. Click **OK**. The instruction is inserted in the template.
6. Click the  **Assemble** 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 has been enabled—click the  **Enable Debugging** button.)
7. Step through the template using the Debugger. (See [Step Through a Template or Script](#) for an explanation on how to do this.)

## To insert a DEBUG instruction in a form template

1. Open the form template for editing. (See [Edit a Template.](#))
2. Create a field in the form template where you want to begin debugging. (See [Create a New Field.](#))
3. Select the field and click the  **Field Properties** button. The **Field Properties** dialog box appears.
4. In the **Variable** box, type **DEBUG**. (Close the **Field Properties** dialog box when you are finished.)
5. Click the  **Test Assemble** 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 has been enabled—click the  **Enable Debugging** button.)
6. Step through the template using the Debugger. (See [Step Through a Template or Script](#) for an explanation on how to do this.)

### To insert a **DEBUG** instruction in a script

1. Open the template for editing. (See [Edit a Template](#).)
2. Edit the computation or dialog whose script you want to debug. (See [Edit a Variable](#) or [Edit a Custom Dialog](#).)
3. Insert the instruction **DEBUG** at the place in the script where you want to begin debugging.
4. Click **Test**. The test assembly window appears along with the **HotDocs Debugger** window, which stops at the instruction following the **DEBUG** instruction. (If the Debugger window does not appear, make sure debugging has been enabled—click the  **Enable Debugging** button.)
5. Step through the script using the Debugger. (See [Step Through a Template or Script](#) for an explanation on how to do this.)

# Step Through a Template or Script

Once you have inserted a DEBUG instruction in a script or in the template, you can begin stepping through the template 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 or script:

When you first test assemble a template or script, HotDocs processes (or runs through) the entire template 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 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 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. 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.

To debug your template 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 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:

To	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	<p>Click  <b>Step Into</b>. The Debugger steps into that field or line to show how HotDocs has processed it using the answers you have provided, specifically:</p> <ul style="list-style-type: none"><li>■ 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.</li><li>■ If you are in a Computation variable field, the Debugger will step into the computation and show you how the script is being processed.</li><li>■ If you are in a regular (non-computed)</li></ul>

	<p>variable field, the Debugger will move you to the next field or line.</p> <p>As you step into components and instructions, you will notice that you are adding these levels of processing to the processing stack.</p>
<p>Ignore a field or line in a script</p>	<p>Click  <b>Step Over</b>. The Debugger will not step through any computations or subtemplates referenced in the field unless it encounters another explicit DEBUG instruction.</p> <p>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.</p>
<p>Move to the next item on the processing stack</p>	<p>Click  <b>Step Out</b>. The Debugger will move you out of the current level of debugging and move you to the next item down on the processing stack.</p> <p>For example, if you are debugging a computation in an inserted template, clicking <b>Step Out</b> will move you out of the computation and back to the inserted template. If you click <b>Step Out</b> again, the Debugger will return you to the parent template.</p>
<p>Stop processing any <i>current</i> DEBUG instructions</p>	<p>Click  <b>Continue</b>. 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.</p> <p><b>Warning:</b> 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  <b>Enable Debugging</b> button in the test assembly window toolbar.)</p>
<p>Cancel the debugging session</p>	<p>Click  <b>Stop</b>. HotDocs closes the Debugger window and disables the Debugger. You are returned to the test assembly window.</p>

	<p><b>Note:</b> You can start the Debugger again by clicking the  <b>Enable Debugging</b> button in the test assembly window toolbar.</p>
<p>View the current field or line of script and any answers used in processing it</p>	<p>Click the <b>Default Watch List</b> tab, which shows a list of variables referenced in the field or script as well as any answers for the variables.</p> <p>As HotDocs processes REPEAT instructions, it displays the value for the current iteration (or index) in the <b>Answer</b> column. As you step through the REPEAT instruction, you can watch the Debugger increment this index. The Debugger also displays the REPEAT COUNTER.</p>
<p>Track the answers for specific variables as the Debugger steps through the interview or script</p>	<p>Click the <b>Custom Watch List</b> tab. See <a href="#">Add Variables to the Debugger Watch List</a> for details.</p>
<p>View the processing stack, or path of execution</p>	<p>Click the <b>Processing Stack</b> tab.</p> <p>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.</p>
<p>Keep HotDocs from debugging the template or script without removing the DEBUG instruction</p>	<p>Click  <b>Stop</b> at the Debugger window. This closes and disables the Debugger window. (To start another debugging session, click the  <b>Enable Debugging</b> button. HotDocs reprocesses the interview or script, which forces HotDocs to process the DEBUG instruction, thus causing the Debugger to open again.)</p>

# 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.)
3. 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.

**Note:** 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.

# Automating HotDocs Server Templates

---

## Overview: Create Templates for Use with HotDocs Server

You can create templates for use with HotDocs Server, which is a Web-based version of HotDocs that allows users to assemble documents over the Internet or a corporate intranet. Since interviews are displayed in a Web browser and document assembly occurs on a Web server, users do not need to have HotDocs installed on their computers. All they need is a compatible Web browser and a word processor to display assembled documents or an application such as Adobe Reader to display assembled forms.

Once a template is automated, it can be published to a Web server where users can access it. Once selected for assembly, an interview is presented in the user's browser to gather answers required by the template. When finished, the user submits the answers back to the Web server where the document is assembled. The user can then choose to download the document and answers, save the document and answers to a Web server, or a combination of these options.

For example, a company's human resources department can integrate its automated templates into an intranet site using HotDocs Server. Employees within the company can go to the intranet site, select a template, provide the required information, and then submit the information to the appropriate HR representative—all from a Web browser. Depending on specifications set by the Web server administrator, employees can also print a copy of the finished document from the Web server, save the document to their computers, or simply store the submitted answers either on the Web server or on their computers.

When developing templates for use with HotDocs Server, you must complete the following steps:

- **Enable templates for use with HotDocs Server.** You can enable templates for use with HotDocs Server. Once enabled, HotDocs warns you if you try to use a feature that is not allowed in a HotDocs Server template. In addition, you can test assemble a HotDocs Server-enabled template using a Web browser to display the interview.
- **Publish templates for use with HotDocs Server.** In addition to the template files, HotDocs requires two other files in order to correctly assemble documents on a server—the HotDocs Variable Collection (.HVC) file and the interview definition (.JS) file. These files are generated when you publish the template for use with HotDocs Server.
- **Upload your templates and supplementary files to a Web server running HotDocs Server.** Once the correct files are created, they must be copied to a Web server. You can use a custom uploading process, or you can upload them manually.

Of course, a key requirement for using templates with HotDocs Server is running the application on a Web server to which you have access. Normally, this will be a server maintained by your company, but it might also be a server maintained by a Web site hosting service. Using HotDocs Server requires the skills of a Web page designer or server administrator. For details on obtaining a HotDocs Server license, contact your HotDocs sales representative.

# Differences Between Desktop and Browser-based Interviews

When you use HotDocs Server to assemble a document, it displays a HotDocs interview in your Web browser. This interview is very similar in appearance and function to the interview displayed when using the desktop version of HotDocs. There are some fundamental differences between the two versions of HotDocs, however. These differences limit the features you can implement or change the way features function between the two versions.

## Features Not Allowed in HotDocs Server Templates

The following features are not compatible with HotDocs Server. Using any one of these features will prevent a template from being assembled using HotDocs Server. (An error message is displayed when you try to test the template in a browser or publish it for use with HotDocs Server.) In some cases, there are simple ways of accomplishing the same task without using the feature, however.

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 computation scripts	HotDocs Server cannot process INSERT instructions in computation scripts. However, INSERT instructions can be processed when they are inserted directly in the template. <b>Warning:</b> Templates referenced in an INSERT instruction must be saved to the same folder as the template. Do not include a full file path to the inserted template.
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 HotDocs Server works with answer files, this is not allowed.
Database components	Database components cannot be used in HotDocs Server

	templates. However, you can achieve much of the same functionality using database integration. (See the HotDocs Server documentation for a description.)
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.
Scripting in plain text resources or prompts	You cannot use scripting (such as IF instructions or REPEAT instructions) in a resource or prompt.

## Features That Work Differently in HotDocs Server Interviews

The following features are allowed in HotDocs Server templates, but they work differently in desktop and server interviews. For example, the end result may be the same, but the different user interfaces require changes to the implementation of some features. There are also some features that are ignored when they are used in HotDocs Server templates. Unlike the features not supported in HotDocs Server templates (see above), these features 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. When assembling documents using HotDocs Server, answer sources are ignored, however. (The <b>Select</b> button does not appear on the dialog.) You can use a database integration to achieve similar results, however. (See the HotDocs Server helps for a full description on how to create this type of integration.)
Automatically selecting a Multiple Choice option when the variable is unanswered	At the Multiple Choice Variable Editor, you can click the <b>Options</b> tab and specify which option should be automatically selected if the variable is unanswered when displayed. Although these "default" options are ignored in HotDocs Server 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 server interviews, but the way answers are formatted as they are

	entered is slightly different in each version.
Example formats	<p>You can include variable fields in variable prompts and dialog element text by inserting the name of the variable within chevrons (« »). You can also specify an example format for variables within the chevrons. For example, to display a text answer in ALL CAPS, insert the variable name as follows: <b>«Text Var:LIKE THIS»</b> (where <i>Text Var</i> is the name of your variable).</p> <p><b>Warning:</b> The FORMAT instruction is not supported in dialog element text or variable prompts. For example, <b>«FORMAT( Text Var, "LIKE THIS" )»</b> or <b>FORMAT "a, b, and c"</b> does not work in a HotDocs Server interview. This instruction can be used to format text in the assembled document, however.</p>
Script link dialog elements	You can use a script link dialog element to add a button to a dialog that processes a Computation variable. However, in a HotDocs Server template, the Computation variable cannot include an ASK instruction or otherwise cause a dialog to be asked. (HotDocs Server ignores ASK instructions in script link computation scripts.)
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 (HotDocs Server) interviews. Web links that launch such databases do not appear in desktop interviews. (See <a href="#">Add a Button to a Dialog that Links to a Database.</a> )
Image dialog elements	Images you include in a dialog must be saved to the same folder as the template. Also, bitmap (.BMP) images cannot be displayed in a browser interview.
Variable field (<< >>) in Multiple Choice option prompts and merge text	HotDocs Server does not process variable fields (« ») in Multiple Choice option prompts. For example, if you use <b>«Employee Name»</b> as the prompt for a Multiple Choice option, rather than displaying the answer for the <i>Employee Name</i> variable as the prompt, HotDocs Server displays <b>«Employee Name»</b> . Variable fields in merge text are also not processed in the interview, but they are processed when the answer is merged in the assembled document.
Resources for variables and dialogs	HotDocs Server only supports Plain Text and URL resources. All other resource types (HTML Help, Windows Help, Folio Infobase, Custom Program) are ignored in HotDocs Server interviews. (No resource is displayed for the dialog or variable.) In addition, HotDocs displays all URL resources as hyperlinks that open separate windows; the actual Web page does not appear in the resource pane.

End of Interview dialog	HotDocs Server interviews do not include an <i>End of Interview</i> dialog. Selecting <b>Hide End of Interview dialog</b> at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab) has no effect on the server interview
Question and Answer summaries	HotDocs Server does not have an option for creating Question and Answer summaries. Selecting <b>Use variable names in summaries</b> at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab) has no effect on the server interview.
"TypeHere" bookmarks	Since HotDocs Server does not send assembled documents to the word processor, selecting <b>Move to the "TypeHere" bookmark</b> at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab) is not applicable to HotDocs Server templates.
Prevent users from using an answer file to assemble a document	Since answer file selection and processing is handled by your HotDocs Server Web application, the <b>Do not use answer files</b> property at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab) does not affect server interviews.
Max WHILE iterations	HotDocs Server does not honor the <b>Maximum WHILE iterations</b> value specified at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab). (You should be careful not to create an infinite WHILE loop in your templates.)
Max processing stack depth	HotDocs Server does not honor the <b>Maximum processing stack depth</b> value specified at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab). (You should be careful to avoid infinitely recursing (processing) a computation.)
Unanswered variable placement	HotDocs Server does not honor the <b>Unanswered variable placeholder</b> selected at the <b>Component File Properties</b> dialog box ( <b>Assembly</b> tab). It always uses the default ( <b>*** Variable ***</b> ). (This placeholder is honored in assembled documents, however.)
Press the Enter key to insert a paragraph mark in a multi-line text field	HotDocs Server ignores the Text variable option that allows users to press the <b>Enter</b> key to insert a new paragraph mark in multi-line answers. (The <b>Enter</b> key always inserts a line break.)
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. HotDocs Server correctly removes the ampersand when the prompt is displayed in the interview, but does not

	make the following character an accelerator key.
ASCEND and DESCEND instructions	The ASCEND and DESCEND instructions, which are used for sorting repeated dialogs, are ignored in HotDocs Server interviews. They can be used to sort answers in the assembled document, but not to display a sorted list of answers in the interview.
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 in resources	Dot codes used in Computation variables are not processed when the computation is used in a component resource.

# Enable Templates for Use with HotDocs Server

You can enable a template for use with HotDocs Server at the **Component File Properties** dialog box. Once enabled, you can view an interview for the template in a Web browser to see how it will appear when assembled using HotDocs Server. (See [View an Interview in a Web Browser](#).) In addition, when this option is set, HotDocs displays a warning message whenever it detects a feature used in the template that is not supported by HotDocs Server.

For example, HotDocs Server cannot assemble templates containing clauses or clause libraries, database components, or certain computation or dialog scripts. HotDocs disallows use of these features in HotDocs Server-enabled templates. Additionally, there are some features you can use when automating and assembling, but that may not function correctly when the interview is displayed in a Web browser. (See [Differences Between Desktop and Browser-based Interviews](#).)

The following types of templates may be enabled for use with HotDocs Server:

Type	Description
Interview template	Interview templates must contain an INTERVIEW computation and <b>Use custom interview</b> must be selected (and specified) at the <b>Interview</b> tab of the <b>Component File Properties</b> dialog box. (See <a href="#">Create an Interview Template</a> and <a href="#">Change Component File Properties</a> .) All interview templates have a .CMP file name extension.
Text template	Text templates may be created using Microsoft Word or WordPerfect. Word templates must be created in rich text (.RTF) format and WordPerfect templates must be created in WordPerfect (.WPT) format. (Assembled text documents may be viewed in the corresponding word processor.)
Form template	Form templates are created using HotDocs Automator. HotDocs Server supports both HotDocs (.HFT) and PDF-based (.HPT) form templates. (HotDocs® PDF Advantage Server Edition is required to assemble PDF-based templates using HotDocs Server.) (Assembled form documents may be viewed in HotDocs Filler or, if saved as PDF, in Adobe Reader.)

## To enable templates for use with HotDocs Server

1. Open the template's component file for editing. (See [Open and Close Component Manager](#).)
2. Click the  **Component File Properties** button. The **Component File Properties** dialog box appears.
3. Click the **HotDocs Server** tab and select **Enable template for use with HotDocs Server**.
4. Select any of the following options:
  - Select **Interview outline initially showing** to cause the interview outline to appear when the interview is first displayed. The interview outline lists all of the dialogs in the interview. Users can use the interview outline to navigate through the interview.
  - Select **Allow user to hide/show interview outline** to let users show and hide the interview outline as needed.
  - Select **Resource pane initially showing** to cause the resource pane to appear when the interview is first displayed. The resource pane shows helpful information about questions in

the interview.

- Select **Allow user to hide/show resource pane** to let users show and hide the resource pane as needed.
- Select **Instant Update initially ON** to have HotDocs Server 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.)
- Select **Allow user to turn Instant Update on/off** to let users control how frequently HotDocs updates the interview.
- Select **Single-page interview initially ON** to have HotDocs Server show all of the dialogs at once, rather than individually. (Because all of the interview questions are presented on a single page, this options is best used with templates with shorter interviews.)
- Select **Allow user to turn single-page interview on/off** to let users choose whether to view all dialogs at once or individually.

**Notes:**

- HotDocs Server-enabled templates can still be assembled using the desktop version of HotDocs.
- You can automatically enable all new templates you create for use with HotDocs Server. See [Enable All Templates for Use with HotDocs Server](#).

## Add a Button to a Dialog that Links to a Database

In HotDocs Server templates, you can add a link to a dialog that opens a URL. This is useful if you want to allow users to retrieve answers from a database.

For example, the site administrator can create a Web page at the specified URL that presents information from a database and allows users to select the appropriate data. Once the data is selected, the Web page is closed and the variable fields in the dialog are filled with answers from the database.

For information on linking a dialog to such an application, see [Add a Hyperlink to a Dialog](#).

# View 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. To do this, you must have Microsoft Internet Explorer 6.0 or greater installed. The interview will not appear if you attempt to use Netscape Navigator or another browser.

## To view an interview in a Web browser

1. Edit the template you want to test assemble using a Web browser. (See [Edit a Template](#).)
2. Ensure the template is enabled for use with HotDocs Server. (See [Enable Templates for Use with HotDocs Server](#).)
3. Once enabled, press the **Shift** key and then click the  **Test Assemble** button. 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).

**Note:** You can also view an interview in a Web browser when assembling a document from the template library. To do this, either click **Test in Browser** (**Template** menu) or press the **Shift** key as you click the  **Assemble** button.

# Publish Templates for Use with HotDocs Server

For each template you create for Web-based assembly using HotDocs Server, you must also create a HotDocs variable collection (.HVC) file, as well as an interview definition (.JS) file. The variable collection file stores all of the variables used in a given template, while the interview definition file defines (in JavaScript) how a Web browser displays the interview. In order to generate these files, you must use the Publishing Wizard to publish your templates for use with HotDocs Server. Then, once published, all these files must be uploaded to a Web server running HotDocs Server. (See [Overview: Upload Published Files to a Web Server](#) for details.)

## To publish templates for use with HotDocs Server

1. Enable all templates you want to publish for Web-based assembly. (See [Enable Templates for Use with HotDocs Server](#).)
2. 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.
3. Select **Template files for use with HotDocs Server** from the **Publish as** group
4. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder where you want HotDocs to save the published files.
5. Optionally, if you want to upload the published files at the end of the Publishing Wizard, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server](#).)
6. Click **Next**. The **Additional Files** dialog box appears.
7. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

### Notes:

- As you are publishing your files, you may receive a HotDocs Publishing error. This is because your template contains automation that is unsupported in HotDocs Server. When HotDocs detects such a feature, it displays an error message, describing the error it found. You cannot publish the files until you remove the feature that is causing the error. (See [Differences Between Desktop and Browser-based Interviews](#).)
- If you're publishing an interview template (or .CMP file), you must make sure the component file has the **Use INTERVIEW computation** component file property set for it. (See [Create an Interview Template and Change Component File Properties](#).)

**Warning:** 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.

## Upload HotDocs Server Files to a Web Server

If you are creating templates to be used with HotDocs Server, for each published template, four files need to be copied to the Web server that is running HotDocs Server: the template (.RTF or .WPT) file, the component (.CMP) file, the variable collection (.HVC) file, and the interview definition (.JS) file. You can either move these files manually (using an FTP client or other program, such as Microsoft FrontPage), or you can have HotDocs upload them at the end of the publishing process. To have HotDocs upload them for you, see [Create Custom HTML Documents for Template Uploading](#) and [Upload Templates to a Web Server](#).

Once you finish creating, publishing, and uploading your template files to the Web server, the remainder of the work of integrating your templates with HotDocs Server is done by a Web site administrator. You can find detailed instructions on how to set up the Web server with the HotDocs Server software package. (Contact your HotDocs sales representative for details.)

# Automating Form Templates

## HotDocs Automator Toolbar Buttons

HotDocs offers several ways for you to complete tasks in HotDocs Automator. How you access these commands depends on your personal preferences—for example, some developers prefer to use menus, while others prefer to use buttons.

### To choose a command

1. Edit a form template. (See [Edit a Form Template](#).) The HotDocs Automator window opens.
2. Select a command, based on the following:
  - Click a menu name to display the associated commands, and then choose a command to execute it.
  - Click a toolbar button to perform a certain common task, such as printing or creating a variable field.
  - Right-click anywhere on a form to display a shortcut menu, and then choose a command from the shortcut menu to execute it.

The following table gives a summary of the Automator toolbar buttons, many of which are also on the assembly window toolbar:

Button Name	What It Does
 <b>HotDocs Library</b>	Opens the HotDocs template library. If the library is already open, it brings it to the front.
 <b>New Template</b>	Creates a new, untitled template.
 <b>Open Form</b>	Opens an existing template.
 <b>Save Form</b>	Saves any changes you've made to the current template and component file.
 <b>Print Form</b>	Prints the template currently displayed.
 <b>Field Properties</b>	Displays the <b>Field Properties</b> dialog box, where you can change how the answer the user enters is merged in the assembled document.

 <b>Answer Font</b>	Displays the <b>Font</b> dialog box, where you can change the font face, size, or style for the answer.
 <b>Detect Field</b>	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.
 <b>Align Fields</b>	Displays the <b>Align</b> dialog box, where you can arrange the size and layout of multiple fields.
 <b>Undo</b>	Removes the most recent changes you've made to the template.
 <b>Redo</b>	Reapplies any changes you may have undone.
 <b>Variable Field</b>	Displays the <b>Variable Field</b> dialog box, where you can attach a new variable to a field, or edit the merge field properties of an existing variable.
 <b>Edit Component</b>	Displays the <b>Variable Editor</b> , where you can edit the component properties of the selected variable.
 <b>Component Manager</b>	Opens <b>Component Manager</b> , where you can work with the components used in the template.
 <b>Test Assemble</b>	Starts a test assembly, showing how the current template will appear to users during assembly.
 <b>Fill Tool</b>	Allows you to click on a field and enter a test answer.
 <b>Select Tool</b>	Creates new fields (on double-click); also selects and moves existing fields.
 <b>Select Text and Fields Tool</b>	<b>(HotDocs form templates (.HFT) only)</b> Creates new fields (on double-click); also allows you to select underlying template text.

 <b>Scroll Tool</b>	Scrolls the template vertically and horizontally.
 <b>Zoom Tool</b>	Changes the view, increasing or decreasing the magnification.
 <b>Page Width</b>	Changes the view to fit the left and right edges of the template in the window.
 <b>Full Page</b>	Changes the view to fit the top and bottom edges of the template in the window.
 <b>Show Fields</b>	Shows and hides the colored HotDocs variable fields.
 <b>Show Variables</b>	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 <a href="#">Preview the Formatting of Answers in a Field.</a> )
 <b>Show Thumbnails</b>	Shows and hides the <b>Thumbnails</b> pane. You can click on a thumbnail (a small image of each page in the template) to move through the template.
 <b>Previous Page</b>	Changes the view to show the previous page.
 <b>Next Page</b>	Changes the view to show the next page.
 <b>HotDocs Options</b>	Opens the <b>HotDocs Options</b> dialog box where you can customize how HotDocs works.
 <b>What's This?</b>	Changes the mouse cursor to a <b>What's This?</b> help cursor. When you click on a button or property with this cursor, a help window appears, summarizing the purpose for that option.

**Notes:**

- 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.

# Start and Exit HotDocs Automator

To create and edit form templates, you must open HotDocs Automator. To edit saved form documents, you must start either HotDocs Filler or Automator.

## To start HotDocs Automator

- Complete one of the following steps:
  - At the **Start** menu, click **Programs > HotDocs 2008 > HotDocs Automator**. HotDocs opens an empty Automator window.
  - At the HotDocs template library, choose **HotDocs Automator** from the **Tools** menu.

## To close HotDocs Automator

- Choose Exit (**File** menu). (You can also press **Alt+F4**.)

**Note:** See [Edit a Form Template](#) for instructions on opening a template and editing it.

# 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:

To	Do This
Scroll the form horizontally and vertically	Click the  <b>Scroll Tool</b> 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  <b>Zoom Tool</b> button and then click on the form. The magnification on the form changes.  To zoom out, press the <b>Ctrl</b> or <b>Shift</b> key while clicking with the  <b>Zoom Tool</b> button.
View the next or previous page in the form	Click the  <b>Next Page</b> button or the  <b>Previous Page</b> button.
Go to any page in the form	Choose <b>Go To (Edit menu)</b> . The <b>Go To Page</b> 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 form <b>Warning:</b> The <b>Find</b> command works only in form template (.HFT) file	Choose <b>Find (Edit menu)</b> . The <b>Find</b> dialog box appears, where you can enter the text for which you are searching on the form. (To perform a case-sensitive search, select <b>Match case</b> .)  To find the next instance of text, choose <b>Find Next (Edit menu)</b> .

# Create a Form Template

## Warnings:

- 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, to create a form template, you must use HotDocs PDF Advantage, Professional Edition, which lets you create and save form templates in PDF format.
- While PDF Advantage is required for the creation of all *new* templates, PDF Advantage is not required for editing *existing* .HFT files.

Using HotDocs PDF Advantage and 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.

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.

## Part 1: To save the document as a PDF File

**Warning:** If the document is already a PDF, you do not have to complete this part. Skip to Part 2.

1. Open the document in a Windows program with printing capability.
2. Select the **Print** command for that program.
3. At the printing dialog box, select **HotDocs PDF Driver** as the printer.
4. Assign any properties you need to the PDF (such as security settings, watermarks, and so forth.)
5. Click the **Print** or **OK** button. The **Save As** dialog box appears.
6. If necessary, browse and select the location for the file.
7. In the **File name** box, type a name for the new file.
8. At the **Save as type** drop-down list, select **PDF** as the file type.
9. Click the **Save** button. The PDF file is created and stored in the location you chose.

## Part 2: To save the PDF file as PDF-based form template

1. Run HotDocs Automator. (See [Start and Exit HotDocs Automator](#).)
2. Click the  **Open Form** button. The **Open** dialog box appears.
3. At the **Files of type** drop-down list, select **PDF Files (\*.pdf)**. The dialog box changes to show any PDF files in the current folder.

4. Browse to the location where you saved the PDF file you created earlier, in Part 1.
5. Select that file and click  **Open**. The form appears in Automator.
6. Select **Save As** (**File** menu). The **Save As** dialog box appears.
7. In the **File name** box, type a name for the new file.
8. At the **Save As** drop-down list, select **HotDocs PDF Template (\*.hpt)**.
9. Click **Save**. The file is saved as a PDF-based form template.

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 [Overview: Template and Component Files](#).)

**Notes:**

- Unlike text templates, form templates aren't automatically added to a HotDocs library when they are created. You must manually add them to the library. See [Add Templates and Other Files to a Library](#) for details.
- 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.)

**Warning:** Information in this topic applies to **HotDocs PDF Advantage** users only. For details on purchasing a license for PDF Advantage, contact your HotDocs sales representative at (800) 500-3627.

# Save Existing HFT Files as PDF-based Form Templates

## Warnings:

- 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, to create a form template, you must use HotDocs PDF Advantage, Professional Edition, which lets you create and save form templates in PDF format.
- While PDF Advantage is required for the creation of all *new* templates, PDF Advantage is not required for editing *existing* .HFT files.

With PDF Advantage, 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.

**Warning:** 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** boxes. (Tip: Information you type in these boxes 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. )
6. Click **OK**. The PDF-based form template is created.

You can now open the new template in HotDocs Automator to review and work with the template. (See [Edit a Form Template](#).)

# 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. Then 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.

**Note:** 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.

# 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 **Insert Pages** (**Edit** menu). The **Insert Pages** dialog box appears.
3. Click the **Form containing pages to insert** drop-down box and select the file from which you want to copy the pages (or 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 originating file into the current file. (See [Use Component Manager to Work with Components](#).)

### Notes:

- 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**.

# 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** boxes.
4. Click **OK**. The page(s) you specified are deleted.

# Update the Underlying Text in a Form Template

Frequently, the courts or other agencies to which 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...** drop-down 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.
5. 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](#).)

# 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:

To	Do This
Open an additional copy of the current form	Choose <b>New Window</b> .
Display all open forms arranged from left to right	Choose <b>Tile Vertical</b> .
Display all open forms arranged from top to bottom	Choose <b>Tile Horizontal</b> .
Display all open forms so they overlap from top-left to bottom-right	Choose <b>Cascade</b> .

# Globally Change Properties in a Template or Group of Templates

**Warning:** 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 Form Template Files 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.

## Part 1: Create a source template

1. At the Automator window, click the  **New Template** button. A new, blank form template appears.
2. 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.)
3. Apply any changes to these fields that you want to be made globally.
4. Specify any template or addendum properties.
5. Click the  **Save** button, and name the source template.

**Warning:** 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.

## Part 2: Apply the global changes from the Property Replicator tool

1. In the source template, click **Property Replicator** (**Tools** menu). The **Property Replicator** dialog box appears.
2. In the **Source template** box, click the  **Browse** button to browse and select the source template you created earlier. This file contains the formatting you want to apply globally.
3. In the **Destination templates** box, click the  **Browse** button to browse and select the files.

**Note:** 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.

4. 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.
5. After specifying the properties to be changed, click **Replicate**. The **Replicate Properties** dialog box appears, showing which template files have been modified.
6. Click **OK**. The **Property Replicator** dialog box appears again.
7. Click **Close**.

# 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:

To	Do This
Save changes to the currently open template and component file	Click the  <b>Save</b> button.
Close the currently open template	Click <b>Close (File menu)</b> .
Close all open templates	Click <b>Close All (File menu)</b> .

**Note:** You can have HotDocs automatically save back-up copies of form templates you are automating. See [Protect Forms By Backing Them Up](#) for details.

# Add Templates and Other Files to a Library

After you create a library, you can add files to it. This includes folders, template files, Web addresses, and so forth. You can also add these types of files to an existing library.

When you create a new text or interview template, the template is automatically added to the library. Clause libraries created at the template library are likewise added to the library automatically. Form templates, however, often must be added to the library manually.

## To add an item to a 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**, **Model Document**, **Clause Library**, **Auto-Assemble File**, **Web Address**, and **Folder**.)
4. Click the  **Browse** button next to the **File name** box 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** box, 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** box. The description appears in the **Properties** tab when the library item is selected.

### Notes:

- You can simultaneously add multiple files to a library. However, doing this doesn't allow you to assign properties (such as titles and descriptions) to each individual file. To add multiple files, select all the files you want to add in the folder you are browsing. Once they are added, you can modify the properties. (See [Change the Properties of a Library Item](#).)
- You can also add items by dragging files from a Windows Explorer folder to the library. This method allows you to add files like PDF files, word processor documents, and so forth to the library. Once you add the file, you may need to update the item properties (such as add a title or description). See [Change the Properties of a Library Item](#) for details.
- 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.

# 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.

# Have HotDocs Generate a Default Interview

When you assemble a template, HotDocs displays an interview, based on variables and instructions it finds in the template. An *outline* of the interview is displayed in the left pane of the assembly window.

This interview is comprised of dialogs that contain one or more questions, which the user must answer to assemble a complete document. As users answer questions, the outline updates to show which questions are still unanswered.

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, if you need control over the order questions are asked in the interview, you can create a custom interview component. See [Define a Custom Interview](#) for details.)

## To generate a default interview for a template

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 **Interview** tab. The window changes to show interview options.
4. Select **Generate default interview**.

Now, whenever this template is assembled, the questions in the interview will be asked based on the order they are asked in the template.

If you want to control the appearance of just one or two dialogs in the interview, you can insert ASK instructions at the place in the template where you want your questions asked. See [Control When Your Dialogs Appear](#) for details. If you want to use a custom interview script, see [Define a Custom Interview](#).

# Define a Custom Interview

When you assemble a template, HotDocs displays an interview, based on variables and instructions it finds in the template. An *outline* of the interview is displayed in the left pane of the assembly window.

This interview is comprised of dialogs that contain one or more questions, which the user must answer to assemble a complete document. As users answer questions, the outline updates to show which questions are still unanswered.

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, 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 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.

One reason why it may be beneficial to create a custom interview is to speed up the process with which HotDocs displays and updates the interview. To explain, during assembly, the interview that is presented to the user is dynamic. This means that each time a user enters an answer, the entire interview is updated to reflect any changes caused by that answer. Depending on the complexity of the template and the frequency with which variables and instructions are used in the template, this updating may take longer than expected because HotDocs must process each field. By creating a custom interview, you can create a script that asks these components and processes these instructions just once, thereby reducing the number of times HotDocs has to process each field. This can improve assembly speed considerably. (For a detailed description of how a complex template can slow interview speed, see [Interviews and Complex Text Templates](#) as well as [Overview: HotDocs Interviews](#).)

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.

## Part 1: To create a computation that contains your scripting

1. At the template (which should be completely automated), open Component Manager. (See [Open and Close Component Manager](#).)
2. Select **Computation Variables** from the **Components** drop-down list and click the  **New Component** button. HotDocs opens the **Computation Editor**.
3. Enter a name in the **Variable name** box. (The component can use any name, including INTERVIEW.)
4. Using a series of instructions, specify how you want variables in the template to be asked, 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 are 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.

## Part 2: To specify that the interview component be used to generate the interview

1. With Component Manager still open, click the  **Component File Properties** button. The **Component File Properties** dialog box appears.
2. Click the **Interview** tab and select **Use custom interview**.
3. Specify the name of the Computation variable in the **Interview component** box.

Now, HotDocs will use this computation to generate the interview.

**Notes:**

- 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** box.
- You can test your custom interview computation to make sure each variable is asked correctly. To do this, clear **Use custom interview** at the **Component File Properties** dialog box, insert the interview component at the beginning of the template, and test assemble it. (See [Test Assemble a Text Document](#).) If HotDocs displays any dialogs after the expected final dialog, that means some variables were not included in the script and you must go back and modify the script. (Make sure you remove the computation from the template once you are finished testing, and remember to select **Use custom interview** again.)
- When you publish a template for use with HotDocs Server that has the **Use custom interview** property set, HotDocs uses the computation to build the interview definition (or .JS) file. (See [Publish Templates for Use with HotDocs Server](#).)

# 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.

### 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 a Field](#).)
  - 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](#).)

#### Notes:

- 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**.

# 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). See [Overview: Create and Customize Variables](#).

## To attach variables to fields

1. Edit a form template. (See [Edit a Form Template](#).)
2. Create the field to which you want to attach the variable. (See [Create a Form Field](#).)
3. 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 include Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information.

### Notes:

- 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 fields in the template, 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 of the fields before you click the  **Variable Field** button. (See [Group Fields So Answers Can Flow From One Field to Another](#).)

# 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:

Social Security Number

S	o	c	i	a	l	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 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** boxes.

When a user answers the variable, each character of the answer will be merged in its own field.

**Note:** 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 [Create a Pattern for a Text Variable](#)) 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\_XX\_XXXX* 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*).

# Remove a Variable from a Field

Once you've attached a variable to a field, you can remove it.

## To remove a variable from a field

1. Edit a form template. (See [Edit a Form Template](#).)
2. Select the field whose variable you want to remove.
3. Choose **Clear Variable** (**Field** menu). The variable is removed from the field.

# Show and Hide Variable Names and Field Backgrounds

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 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.
3. To hide or show variable names
4. Edit a form template. (See [Edit a Form Template](#).)
5. Click the  **Show Variables** button.

**Note:** You can control whether field variables and field backgrounds are shown or hidden when you open form templates. See [Control How Forms Appear when Opened](#) for details.

# 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.

## 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 a Field](#).)
  - 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](#).)

### Notes:

- 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:
  - Press the **Ctrl** key while clicking each field.
  - 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.

HotDocs Automator provides several other methods for selecting fields or canceling the selection, as described in the following table:

To	Do This
Select all the fields on the current page	Click <b>Select All</b> ( <b>Edit</b> menu). (You can also press <b>Ctrl+A</b> .)
Cancel the selection of all selected fields	Click outside the fields, or press the <b>Esc</b> key.
Cancel the selection of only one of a group of selected fields	Hold down the <b>Ctrl</b> key and click that field.
Add fields to your group of selected fields without canceling the selection of those already selected	Hold down the <b>Ctrl</b> key and click the new fields.

### Notes:

- Click the  **Show Fields** button to show and hide field colors.
- 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 Fields](#).)

# 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.

## To create a check-box field

1. Edit a form template. (See [Edit a Form Template](#).)
2. Double-click in the field margins, 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**.

**Note:** When assigning a True/False variable, if you want to merge a character other than an **X**, indicate the character in the **Format** box. To change the font for the field, see [Adjust Appearance of Check-Box Fields](#). When assigning a Multiple Choice variable, you should group the fields before attaching the variable. See [Group 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](#).)

### Notes:

- 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](#).)
- You may need to change other properties of the field, as well, such as horizontal and vertical alignment, borders and margins, and so forth.
- 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). See [Overview: Create and Customize Variables](#).

## To attach variables to fields

1. Edit a form template. (See [Edit a Form Template](#).)
2. Create the field to which you want to attach the variable. (See [Create a Form Field](#).)
3. 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 include Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information.

### Notes:

- 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 fields in the template, 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 of the fields before you click the  **Variable Field** button. (See [Group Fields So Answers Can Flow From One Field to Another](#).)

# 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  **Detect** button. This button 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.

# 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 template, 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 **Order/Size** 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** boxes. (You can also click the up or down arrows for each field to change the distance.)

### Notes:

- If a field you want to move is part of a group, you must first ungroup the fields. (See [Ungroup 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.)
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** 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.

# 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.)

# Copy One or More 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 [Copy Components from One File to Another](#) 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](#).)
3. 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.)

### Notes:

- 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](#).)

# Resize a Field

*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 template. 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 the 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 **Order/Size** 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.

# 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 **Order/Size** 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 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.

**Note:** 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 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 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.

## 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](#).)

### Notes:

- When fields are grouped, you cannot change properties for individual fields. You must first ungroup the fields. (See [Ungroup 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](#).)

# Ungroup 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.)
4. 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.

**Warning:** 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.

# 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](#).)
2. 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  **Edit 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).

**Note:** You can also create a wizard for a table. A table wizard appears as a spreadsheet button (  ) instead of as an answer wizard button (  ), 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.

# Change Field Properties

---

## Preview the Formatting of Answers in a 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.

**Warning:** 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 Variable Names and Field Backgrounds](#).)
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 box 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 box. The field in the underlying form updates to show the text you entered.

# Show and Hide Variable Names and Field Backgrounds

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 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.
3. To hide or show variable names
  - Edit a form template. (See [Edit a Form Template](#).)
  - Click the  **Show Variables** button.

**Note:** You can control whether field variables and field backgrounds are shown or hidden when you open form templates. See [Control How Forms Appear when Opened](#) for details.

# 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.

## 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 following information:

Type of Field	Default Color	Description
Edit field	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 field	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 <b>X</b> or other character. (See <a href="#">Create a Check-Box Field</a> .)
Resource field	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 <a href="#">Create a Resource Link on a Form</a> .)  Resource fields provide help for the form in general, instead of for a specific variable or dialog.
Control field	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.
--	--	------------------------------------

**Note:** You can change the default colors used for form fields at the **HotDocs Options** dialog box. (See [Change Colors in HotDocs Automator.](#))

# 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 your users' computers.

## 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.

**Note:** You can specify default font properties for all new fields you create at HotDocs Options. See [Adjust Appearance of Edit Fields](#) for more details.

# Format Lines and Paragraphs of Text in a 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:

To	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 <b>First line indent</b> 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 <b>Lines per inch</b> box.
Indicate how many lines can appear in the field	Enter the number of lines in the <b>Maximum lines</b> 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 <b>Max chars/line</b> box. When the number of characters exceeds this limit, text will wrap to the next line.

# 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:

Applicant Name 0°      Applicant Name 90°      Applicant Name 180°      Applicant Name 270°

### Notes:

- 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 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.

# Keep Contents of Fields from Printing

*These instructions can also be used to keep field text from printing 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 during direct-fill assembly.
3. Click the  **Field Properties** button. The **Field Properties** dialog box appears.
4. At the **Type** tab, select **Non-printing field**.

# 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.
4. Select the field and click the  **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. This keeps any text in the field from appearing on the assembled document, which makes the field completely invisible.

# 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  **Edit Component** button. The **Text Variable Editor** appears.
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 box.
  - **Image** displays the resource as graphic file on which the user clicks. Enter the file name of the graphic in the **Image file name** box.
  - **Bar code** displays the resource in bar code format. Enter the text you want to encode in the **Bar code text** box.
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 cross out static text on a form both at the **Form Document** tab of the assembly window and in HotDocs Filler.*

Sometimes a form template 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](#).)

**Note:** You can use the alignment controls on the **Field Properties** dialog box to position the strike-through text more accurately.

# 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](#).)
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.

**Note:** To preview the circle (see [Preview the Formatting of Answers in a 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](#).)
  - 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.
  - 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). (Tip: With a True/False variable, you can display one graphic if the answer is *true* and another graphic if it's *false*. To do this, enter two file names separated by a forward slash (/) in the **Format** box. )
3. Click **OK** at the variable editor and at the **Variable Field** dialog box.
4. With the field still selected, click the  **Field Properties** button. The **Field Properties** dialog box appears.
5. At the **Type** tab, select **Image** from the **Display type** group.
6. Optionally, change the **Field type** to **Control** to restrict the user's ability to modify the field. (See [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. (See [Change the Field Type](#).)
8. Optionally, click the  **Variables** button in the Automator toolbar to view the image.

The image is visible when a user assembles a document and views the **Form Document** tab.

# 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 a Field or Change the Borders and Margins of a Field.](#))

There are two methods for creating a bar code:

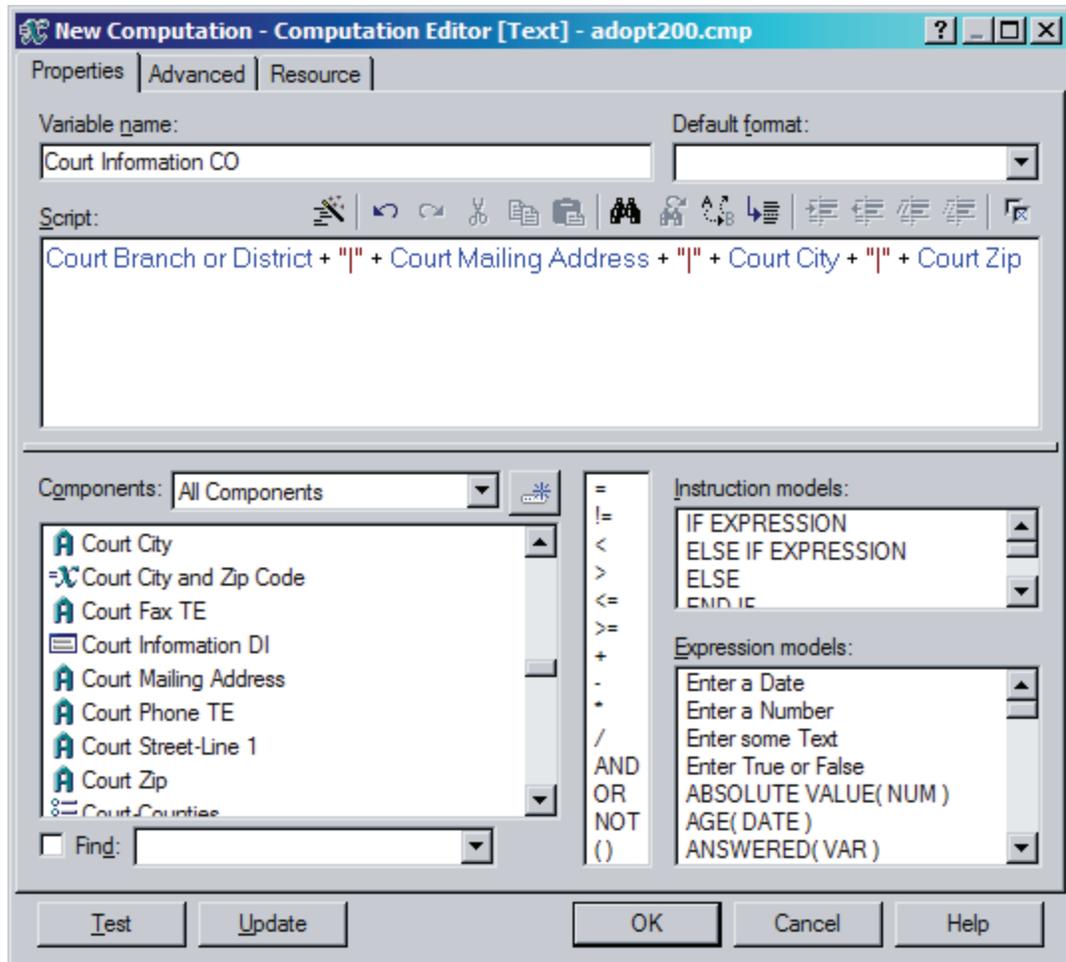
- **Create a bar code using a variable:** When the answers you want to convert to a bar code must be supplied by the user, you can create a variable, such as a Computation variable, which will then merge the answers in the field and convert them to bar code format.
- **Create a bar code using predefined 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.](#))
3. 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 [Overview: Instruction and Expression Models.](#))

### Example:

Using a distinct delimiter character in the script makes it easier to distinguish separate answers when the bar code is scanned into a database.



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 (|).)

**Notes:**

- 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](#).)
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** 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 steps 4 and 5 for each field in the group, using the same group name as you assigned to the first field.

Now, when the user answers the question, text that does not fit in the first field will overflow to the next field named in the group.

**Note:** To span text across multiple fields on the same page, see [Group Fields So Answers Can Flow From One Field to Another](#).

# 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 pre-defined 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		
<i>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.</i>		
Name	Phone Number	Relationship to Applicant
Beneficiary Name	Beneficiary	Relationship to

## 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](#).)
4. 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.

### Notes:

- 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 pre-defined table. Generally, these tables ask for information in a fairly standard order—in rows that read from left to right. (See [Merge a List of Answers in a Standard Table](#).)

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.

## Example 1

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			

## Example 2

In this example, the table is divided into sections, but each section is actually a row in the table:

Transaction Code	First	Added Member's Name		Last	Mailing Address		
Transaction Code	First Name	Last Name		Address			
	City	State	Zip Code	Country	Year of Birth		
	City	State	Zip Code	Country	Year of Birth		
	Occupation	Sponsor's Name—Member and Club Number		Former Club		Gender	
	Occupation	Name of Sponsor		Previous Club		<input type="checkbox"/> Male <input type="checkbox"/> Female	
Transaction Code	First	Added Member's Name		Last	Mailing Address		
	City	State	Zip Code	Country	Year of Birth		
	Occupation	Sponsor's Name—Member and Club Number		Former Club		Gender	
	Occupation	Name of Sponsor		Previous Club		<input type="checkbox"/> Male <input type="checkbox"/> Female	
Transaction Code	First	Added Member's Name		Last	Mailing Address		
	City	State	Zip Code	Country	Year of Birth		
	Occupation	Sponsor's Name—Member and Club Number		Former Club		Gender	
	Occupation	Name of Sponsor		Previous Club		<input type="checkbox"/> Male <input type="checkbox"/> Female	

## To put a list in a nonstandard table

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. Select all of the fields for the first row (that is, 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 **Order/Size** tab, and in the **Tab Order** group, enter **1** in the **Row** box.
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, incrementing 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, incrementing the column number for each subsequent column.
8. Attach variables to the fields in Row 1. (See [Attach a Variable to a Field](#).)
9. Select all of the fields in the table and choose **Group as Table** (**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** (**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. (See [Create a REPEAT Instruction to Gather a List of Answers](#).)

# 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 **Order/Size** 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.
8. 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 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.

**Note:** 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](#).)

# 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 [Make Parts of Templates Conditional](#), [Condition Text Using True/False Variables](#), and [Condition Text Using True/False 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. (See [Condition Text Using True/False Variables](#).)
  - For expressions, drag expressions and variables into the **Expression** box. (See [Condition Text Using True/False Expressions](#).)

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.
4. Click the  **Open** button and 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** box.
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](#).)

**Note:** If you need to add content from one template to another, use the **Insert Pages** command. See [Add Pages to a Form Template](#).

# Field Overflow and Addenda

## Define Overflow Properties for a 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. (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. For example, in an addendum, grouped fields and tables are handled a little differently from regular fields. Grouped fields are shown in the addendum as just one answer. Answers from tables, by default, are also sent to the addendum as one item. That is, if one answer overflows, then all of the answers in the table are sent to the addendum. 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 the 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:

To	Do This
Automatically shrink an answer that overflows to a specific point size	Select <b>Shrink answer as needed</b> and enter a point size in the <b>points</b> box.
Automatically send an answer that overflows to the addendum	Select <b>Send answer to addendum</b> . To send only part of a multi-line answer to the addendum, select <b>Split answer</b> . (See <a href="#">Split a Multi-Line Answer Between the Form and the Addendum.</a> )
Define the text that appears in the answer field when an answer is sent to the	Enter the text in the <b>Cross-reference text</b> box.

addendum	
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the <b>Addendum label text</b> box.
Make an answer the first item on its addendum page	Select <b>Begin addendum entry on a new page</b> .
Make an answer the last item on its addendum page	Select <b>Begin new page following addendum entry</b> .

### To define the overflow properties for run-on fields

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:

To	Do This
Automatically shrink an answer that overflows to a specific point size	Select <b>Shrink answer as needed</b> and enter a point size in the <b>points</b> box.
Automatically send an answer that overflows to the addendum	Select <b>Send answer to addendum</b> . To send only part of a multi-line answer to the addendum, select <b>Split answer</b> .
Use a check box in the grouped fields to indicate the answer overflows  Sometimes forms include a check box that indicates when an answer is continued elsewhere, such as in the addendum or in an attachment. Choosing one of these options selects that check box when the answer overflows.	Select either <b>Indicator check box is first in field group</b> or <b>Indicator check box is last field in group</b> , depending on where the check box that indicates overflow is positioned.  Select <b>No addendum entry indicator check box</b> if there is no check box to indicate an answer has been sent to an attachment or addendum.
Define the text that appears in the answer field that an answer is sent to the addendum  If you want the indicator check box to be the only designation that a field overflows, enter	Enter the text in the <b>Cross-reference text</b> box.

NONE in this box.	
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the <b>Addendum label text</b> box.
Make an answer the first item on its addendum page	Select <b>Begin addendum entry on a new page</b> .
Make an answer the last item on its addendum page	Select <b>Begin new page following addendum entry</b> .

### To define the overflow properties for tables

1. Edit a form template. (See [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:

To	Do This
Automatically send an answer (including extra rows in the table) that overflow to the addendum	Select <b>Send answers to addendum</b> .
Define how much of the table is sent to the addendum when there is overflow	<p>In the <b>When sending answers to addendum</b> group, select one of the following options:</p> <ul style="list-style-type: none"> <li>■ Select <b>Send entire table</b> 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, <i>See 1 in Addendum</i>) is merged in the top row of the table and no other rows contain any answers.</li> <li>■ Select <b>Send complete rows</b> to send only rows that overflow to the addendum. For example, if an answer in a row overflows, the entire row is sent to the addendum. The overflow cross-reference text (for example, <i>See 1 in Addendum</i>) is merged in the affected row only.</li> <li>■ Select <b>Send individual cells</b> to send only the contents of a single overflowing field to the addendum. The overflow cross-reference text (for example, <i>See 1</i></li> </ul>

	<i>in Addendum</i> ) 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 <b>Cross-reference text</b> box.
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the <b>Addendum label text</b> box.
Make an answer the first item on its addendum page	Select <b>Begin addendum entry on a new page</b> .
Make an answer the last item on its addendum page	Select <b>Begin new page following addendum entry</b> .
Indent the answer a specific amount of space	Enter a number in the <b>Answer column indent</b> box (or click the up and down arrows to select a number).

# 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:

To	Do This
Always send the answer to the addendum without prompting the user	<p>Select <b>Send answer to addendum</b>.</p> <p>To send just the part of the answer that doesn't fit in the field to the addendum, select <b>Split answer</b>.</p>
Specify the text that will be merged in the field as well as the addendum if the user chooses just to send to addendum	<p>Enter the text in the <b>Cross-reference text</b> and <b>Addendum label text</b> boxes, respectively.</p> <p>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.</p> <p><b>Example:</b></p> <p><b>Cross-reference text:</b> See Addendum 1 Continued in Addendum 1  <b>Addendum label text:</b> Addendum 1 (con't) Addendum 1</p>

# 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, a user enters an answer that is too long for the form field. One option the user has 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. As the template developer, 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 <b>Page margins</b> group.
Define the text that appears at the top of each addendum page	Enter the text in the <b>Header text</b> box.  To learn how to merge page numbers in the header, see <a href="#">Use Answer Overflow and Addendum Text Codes</a> .
Define the text that appears at the bottom of each addendum page	Enter the text in the <b>Footer text</b> box.  To learn how to merge page numbers in the footer, see <a href="#">Use Answer Overflow and Addendum Text Codes</a> .
Change the font properties (including font face, size, and style) of the header or footer text	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, click the <b>Font</b> button and make the changes at the <b>Font</b> dialog box.
Define how much vertical space the header or footer text requires	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, enter a number in the <b>Height</b> box.
Change the alignment of the header or footer text	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, click the <b>Alignment</b> button and choose your alignment

	option.
Insert a variable in the header or footer text	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, click the  <b>Variable Field</b> button. The <b>Variable Field</b> 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 <b>Addendum entries</b> group, click the <b>Font</b> button and make your changes at the <b>Font</b> dialog box.
Define how much space there should be between the addendum label and the answer	In the <b>Addendum entries</b> group, enter a number in the <b>Indentation</b> box.
Define how much space there should be between each answer in the addendum	In the <b>Addendum entries</b> group, enter a number in the <b>Space between</b> box.
Make the addendum appear as a pleading paper	Select <b>Number lines to format as pleading paper</b> .

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.

# 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 <PAGES>* 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>	Current page in the document Example: <b>Header/footer Text:</b> Page <PAGE> <b>Inserted text:</b> Page 5
<PAGES>	Total number of pages in the document, including addendum pages Example: <b>Header/footer Text:</b> Total Pages: <PAGES> <b>Inserted text:</b> Total Pages: 5
<DOCPAGES>	Total number of pages in the document, without including the addendum Example: <b>Header/footer Text:</b> Total Document Pages: <DOCPAGES> <b>Inserted text:</b> Total Document Pages: 3
<PAGEINFO>	Current page in the document / Total pages in the document Example: <b>Header/footer Text:</b> Page <PAGEINFO> <b>Inserted text:</b> Page 4/5

<ADMPAGE>	<p>Current page in the addendum</p> <p>Example:</p> <p><b>Header/footer Text:</b> Addendum Page &lt;PAGE&gt;  <b>Inserted text:</b> Addendum Page 2</p>
<ADMPAGES>	<p>Total number of pages in the addendum</p> <p>Example:</p> <p><b>Header/footer Text:</b> Total number of addendum pages:  &lt;ADMPAGES&gt;  <b>Inserted text:</b> Total number of addendum pages: 3</p>
<ADMPAGEINFO>	<p>Current page in addendum / Total pages in addendum.</p> <p>Example:</p> <p><b>Header/footer Text:</b> Addendum Page &lt;ADMPAGEINFO&gt;  <b>Inserted text:</b> Addendum Page 2/4</p> <p><b>Note:</b> This code inserts nothing if there is only one page in the addendum (instead of inserting <b>1/1</b>).</p>

## 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 the Assembled Document
<REFNR>	<p>Addendum item number</p> <p>Example:</p> <p><b>Cross-Reference Text:</b> See Addendum Item &lt;REFNR&gt;  <b>Inserted text:</b> See Addendum Item 1</p>
<VARNAME>	<p>The name of the variable</p> <p>Example:</p> <p><b>Cross-Reference Text:</b> See '&lt;VARNAME&gt;' in Addendum  <b>Inserted text:</b> See 'Case Description' in Addendum</p>

<p>&lt;ROW&gt;</p>	<p>Table row number (merged in addendum only)</p> <p>Example:</p> <p><b>Addendum Label Text:</b> Row &lt;ROW&gt;, Column &lt;COLUMN&gt;  <b>Inserted text:</b></p> <p>1. Row 1, Column 1:  Bentley Jones</p> <p>Row 1, Column 2:  Father</p> <p>Row 2, Column 1:  Mariah Jones</p> <p>Row 2, Column 2:  Mother</p>
<p>&lt;COLUMN&gt;</p>	<p>Table column number (merged in addendum only)</p> <p>Example:</p> <p><b>Addendum Label Text:</b> Row &lt;ROW&gt;, Column &lt;COLUMN&gt;  <b>Inserted text:</b></p> <p>1. Row 1, Column 1:  Bentley Jones</p> <p>Row 1, Column 2:  Father</p> <p>Row 2, Column 1:  Mariah Jones</p> <p>Row 2, Column 2:  Mother</p>
<p>&lt;MEMBER&gt;</p>	<p>Table cell sequence number (merged in addendum only)</p> <p>Example:</p> <p><b>Addendum Label Text:</b> Cell Number: &lt;MEMBER&gt;  <b>Inserted text:</b> Cell Number: 5</p>

## 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) Example: <b>Code:</b> Page <PAGE:1> <b>Inserted Text:</b> Page 2
A	Uppercase letters Example: <b>Code:</b> Page <PAGE:A> <b>Inserted Text:</b> Page B
a	Lowercase letters Example: <b>Code:</b> Page <PAGE:a> <b>Inserted Text:</b> Page b
I	Uppercase Roman numerals Example: <b>Code:</b> Page <PAGE:I> <b>Inserted Text:</b> Page II
i	Lowercase Roman numerals Example: <b>Code:</b> Page <PAGE:i> <b>Inserted Text:</b> Page ii

# Automating Interview Templates

---

## Create an Interview Template

You can create an interview template, which can gather common information (such as court, attorney, or client information) and save the answers 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, users cannot generate documents from an assembled interview template—they only generate answer files.

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. (See [Suggest an Answer Source for Dialogs](#) and [Share Answers Between Two Dialogs](#).)

**Note:** 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 automatically lock answer file usage and save the answers to a specified answer file. See [Overview: Command-Line Options](#).

### 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**. The **New Template** dialog box appears.
3. Click the **Type** drop-down button and choose **HotDocs Interview Template**.
4. In the **File name** box, enter a file name. HotDocs automatically adds the correct extension (.CMP) to the file name. To save the template in a location other than the default *Templates* folder, include the folder path with the file name in the **File name** box. (You can check the default *Templates* location at the **HotDocs Options** dialog box. See [Change HotDocs Program File Locations](#).)
5. Enter a title for the interview template in the **Title** box (or accept the suggestion HotDocs makes). The title is what identifies the file in the library.
6. Optionally, enter a description in the **Description** box. Descriptions appear in the **Properties** tab of the library when the template is selected.
7. Click **OK**. The **Component Manager** window appears, with a single computation variable in it named **INTERVIEW**. (Your interview templates are not required to use this specifically named Computation variable. If you prefer, you can delete this component and create a new one. Simply specify the name of the new interview component at the **Interview** tab of **Component File Properties**.)

The interview component must contain the script that asks the questions in the interview. Variables and dialogs used in the script will be used to create the interview.

8. Select the interview component and click the  **Edit Component** button. The **Computation Editor** appears.
9. Create the variables you want answered, link them to dialogs, and then create a script that asks the dialogs you want users to see when they assemble this template. (See [Put ASK Instructions in a Computation Variable](#).)

## To edit an existing interview template

1. At the HotDocs library window, select the interview template and click  **Edit Template**. The **Component Manager** window appears.
2. Select the interview component and click the  **Edit Component** button. The **Computation Editor** appears.
3. Make any necessary changes.

### Notes:

- When converting interview templates from previous versions of HotDocs to HotDocs 2008 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. See [Define a Custom Interview](#) for details.
- 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** box as is). See [Create a Text Template Based on an Existing Template](#).

# Define a Custom Interview

When you assemble a template, HotDocs displays an interview, based on variables and instructions it finds in the template. An *outline* of the interview is displayed in the left pane of the assembly window.

This interview is comprised of dialogs that contain one or more questions, which the user must answer to assemble a complete document. As users answer questions, the outline updates to show which questions are still unanswered.

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, 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 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.

One reason why it may be beneficial to create a custom interview is to speed up the process with which HotDocs displays and updates the interview. To explain, during assembly, the interview that is presented to the user is dynamic. This means that each time a user enters an answer, the entire interview is updated to reflect any changes caused by that answer. Depending on the complexity of the template and the frequency with which variables and instructions are used in the template, this updating may take longer than expected because HotDocs must process each field. By creating a custom interview, you can create a script that asks these components and processes these instructions just once, thereby reducing the number of times HotDocs has to process each field. This can improve assembly speed considerably. (For a detailed description of how a complex template can slow interview speed, see [Interviews and Complex Text Templates](#) as well as [Overview: HotDocs Interviews](#).)

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.

## Part 1: To create a computation that contains your scripting

1. At the template (which should be completely automated), open Component Manager. (See [Open and Close Component Manager](#).)
2. Select **Computation Variables** from the **Components** drop-down list and click the  **New Component** button. HotDocs opens the **Computation Editor**.
3. Enter a name in the **Variable name** box. (The component can use any name, including INTERVIEW.)
4. Using a series of instructions, specify how you want variables in the template to be asked, 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 are 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.

## Part 2: To specify that the interview component be used to generate the interview

1. With Component Manager still open, click the  **Component File Properties** button. The **Component File Properties** dialog box appears.
2. Click the **Interview** tab and select **Use custom interview**.
3. Specify the name of the Computation variable in the **Interview component** box.

Now, HotDocs will use this computation to generate the interview.

**Notes:**

- 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** box.
- You can test your custom interview computation to make sure each variable is asked correctly. To do this, clear **Use custom interview** at the **Component File Properties** dialog box, insert the interview component at the beginning of the template, and test assemble it. (See [Test Assemble a Text Document](#).) If HotDocs displays any dialogs after the expected final dialog, that means some variables were not included in the script and you must go back and modify the script. (Make sure you remove the computation from the template once you are finished testing, and remember to select **Use custom interview** again.)
- When you publish a template for use with HotDocs Server that has the **Use custom interview** property set, HotDocs uses the computation to build the interview definition (or .JS) file. (See [Publish Templates for Use with HotDocs Server](#).)

# Linking Templates to a Database

---

## Overview: 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

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.

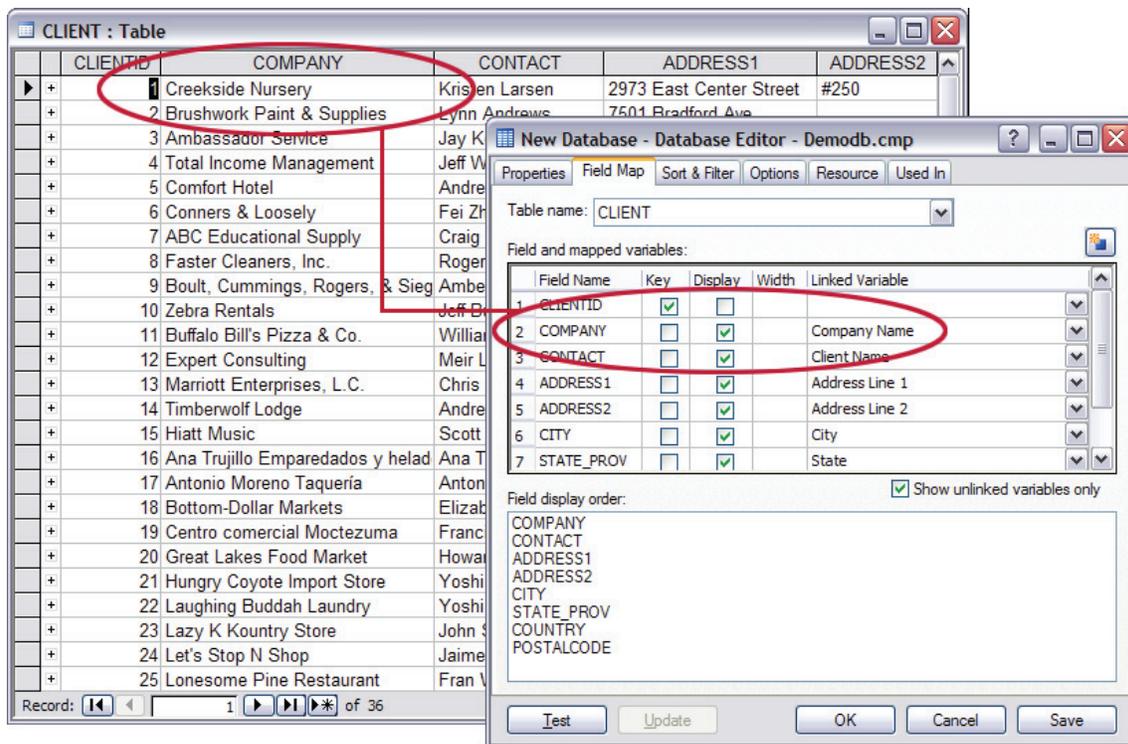
# Understand Database Connectivity

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 [Specify Connection Information Using ADO](#) or [Specify Connection Information 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.

The following example graphic shows a database table and how it can be referenced from a HotDocs database component:



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.](#))

**Note:** If you are converting database components created in HotDocs 5 to HotDocs 2006 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 HotDocs 5.x Database Components to HotDocs 2008.](#)

# Specify Connection Information 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.

Defining the connection between HotDocs and your database requires you to complete two parts:

- Create a database component in the template you want to connect to the database.
- Define the connection string.

## Part 1: To create a database component

1. 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** box.
5. Optionally, perform the following tasks:
  - In the **Title** box, enter a title for the database component. The title will be used in the interview outline and dialog title bar.
  - In the **Prompt** box, 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](#).)

Now that 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.

## Part 2: To build an ADO connection string

1. At the **Properties** tab of the Database Editor, click the **Connection properties are defined by** drop-down button and choose **an ADO connection string**.
2. Click the  **Edit** button next to the **Connection string** box. The **Data Link Properties** dialog box appears.

**Note:** 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.

3. Select the OLE DB provider you need from the list of providers and click **Next**. HotDocs displays

the **Connection** tab of the **Data Link Properties** dialog box.

**Note:** 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.

4. 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.
5. 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.)
6. Click the **Field Map** tab. The window changes to show how fields in the table will be linked to variables in the template.
7. 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](#).)

**Notes:**

■ 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 2008 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 [Specify Connection Information 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](#).

# Specify Connection Information 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 [Specify Connection Information Using ADO](#)), but in some situations, an OLE DB provider may either be incompatible or unavailable for you to use. (See [Use Supported Databases](#) 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.

There are two main parts to creating an ODBC-based connection:

- Create an ODBC data source.
- Associate the data source with a HotDocs database component.

## Part 1: To create an ODBC data source

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.
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** box.
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).

Now that you have 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.

## Part 2: To link a data source to a database component

1. Open the template you want to connect to the database. (See [Edit a Template](#).)
2. Click the  **Component Manager** button in the HotDocs toolbar. The **Component Manager** window appears.
3. Click the **Components** drop-down button and select **Databases** from the list. The list changes to show only database components.
4. Click the  **New Component** button. HotDocs displays the **Database Editor**.
5. Type a name for the component in the **Component name** box.
6. Optionally, perform the following tasks:
  - In the **Title** box, enter a title for the database component. The title will be used in the interview outline and dialog title bar.
  - In the **Prompt** box, 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](#).)
7. From the **Connection properties are defined by** drop-down list, select an **ODBC data source**.
8. From the **ODBC Data Source Name (DSN)** drop-down list, select the name of the data source you created in Part 1.
9. Click the **Field Map** tab. The window changes to show how fields in the table will be mapped to variables in the template.
10. 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](#).)

### Notes:

- 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 [Specify Connection Information Using ADO.](#))
2. When finished, click the  **Save** button next to the **Connection string** box. 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** box and locate the data link file. The component now looks in this separate file for connection information.

**Note:** 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.

**Warning:** 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.

# Convert ODBC-based Connections to ADO

Database connections created in HotDocs 5 were driven by ODBC connectivity, which required you to specify an ODBC data source name in your database component.

Now, when you connect to a database using HotDocs 2008, the primary means of connection is ADO. (See [Overview: Link Templates to a Database](#).) This means that when you convert database components created in HotDocs 5 to HotDocs 2008, HotDocs updates the connection to use the OLE DB provider for ODBC. This allows you to continue to use an ODBC data source, even though the primary mode of connection is ADO. While this type of connection is fully supported, performance of the database component may improve if you change your connection to use a native OLE DB provider for the database you are using.

**Warning:** If your database does not have a native OLE DB provider or if the OLE DB provider does not work, you should continue to use your existing data source, or the data source you created for HotDocs 5. During the conversion, HotDocs automatically selects the OLE DB provider for ODBC drivers and maintains its connection to your data source. You should not have to do anything to update your connection information.

You can change the type of connection by editing the database component, and, provided HotDocs can successfully communicate with the new OLE DB provider, it will maintain all the other properties of the connection—including links between the variables and fields and any filters you have created.

## To change an ODBC connection to ADO

1. Convert the template to HotDocs 2008 format by editing it. (See [Edit a Template](#).) (You can do this by clicking **Start > Programs > LexisNexis HotDocs 2008 > HotDocs**. Once the template library appears, you can either add your HotDocs 5 template to the library, or you can open an entire library of HotDocs 5 templates using HotDocs 2008. See [Add Templates and Other Files to a Library](#) or [Open a Library](#) for instructions.)
2. Once the template is converted to HotDocs 2008 format, open the existing database component for editing. (See [Edit a Database Component](#).) The **Database Editor** appears.
3. At the **Properties** tab of the Database Editor, click the **Connection properties are defined by** drop-down button and select **an ADO connection string**.
4. Click the  **Edit** button next to the **Connection string** box. The **Data Link Properties** dialog box appears.
5. Select the provider that corresponds to your database program. (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.)
6. Click **Next**. The **Connection** tab of the **Data Link Properties** dialog box appears.
7. Specify the correct database information, following the on-screen instructions provided for your specific provider and click **OK**. The **Database Editor** appears again.

Once the new connection information is specified, HotDocs should maintain all existing properties of the component.

8. Check the different properties of the database component (such as field mappings and database options) and then click **OK** to close the **Database Editor**.

# 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  **Edit 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](#).)

**Note:** 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.

**Warning:** 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.

By default, *all* fields in *all* records in the table appear during assembly. However, you can designate that only certain fields appear (see [Choose Which Database Fields Appear During Assembly](#)), control the number of records that appear by using a filter (see [Limit the Number of Database Records from Which a User Chooses](#)), and control the order the records appear by sorting them (see [Sort Records in a Database Table](#)).

### Notes:

- 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.

**Warning:** 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 box next to the field you want to designate as the key.

**Warning:** 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.

**Note:** 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** drop-down 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.
  - **Prompt** displays a dialog box that lets the user decide if the changed answer should be written back to the database.

**Note:** 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**.

# 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:

To	Do This
Allow the user to choose two or more records during assembly	Select <b>Allow multiple selections</b> . (Make sure the variables in the template are included in a <b>REPEAT Database</b> instruction so that all the answers can be merged correctly.) (See <a href="#">Generate a List of Records in an Assembled Document</a> .)
Force HotDocs to merge a list of filtered answers from a database into the document without allowing the user to make the selection	Select <b>Automatically select all filtered records</b> . (See <a href="#">Merge Filtered Records into the Document Automatically</a> .)
Have HotDocs check the database for new or updated data each time you assemble the document	Select <b>Automatically refresh linked answers</b> . (See <a href="#">Cause HotDocs to Update Selected Records Between Assemblies</a> .)
Have HotDocs retrieve data from a database and then disconnect from it	Select <b>Use disconnected (client) record set</b> . (See <a href="#">Choose a Database Cursor</a> .)
Give users the ability to sort columns in the database table in <b>ascending</b> or <b>descending</b> order	Select <b>Allow sorting by end user</b> . 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.

<p>Give users the ability to limit the number of records that appear in the table</p>	<p>Select <b>Allow filtering by end user</b>. 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.</p>
<p>Change the way HotDocs treats variables that may potentially be answered by the database</p>	<p>Click the <b>Treat linked variables as asked</b> drop-down button and select one of the following options: <b>Always</b> (HotDocs considers linked variables as always asked—it will not ask them, even if the database does not contain an answer), <b>Never</b> (HotDocs considers linked variables as never asked—it will ask them, even if the database contains an answer), or <b>Only if Answered from Database</b> (HotDocs asks a linked variable if no answer for the variable exists in the database). (See <a href="#">Ask Variables Already Answered by the Database</a>.)</p>
<p>Specify a user name and password so that information can be retrieved from the database</p>	<p>Type the user name and password in the <b>User name</b> and <b>Password</b> boxes, respectively. This information is encrypted and saved in the component file. (See <a href="#">Specify a User Name and Password for a Database</a>.)</p>

**Note:** 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. (See [Use Supported Databases](#) 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*.

**Warning:** 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](#).)
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. (See [Custom Command Sample](#) 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.

# 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:

To	Do This
Change the order in which the columns are presented	In the <b>Field display order</b> box, 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 <b>Width</b> column. This number must be larger than the number of characters in the column name.
Test the database component	Click <b>Test</b> . HotDocs displays the database table in the test assembly window.

### Notes:

- 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** box. 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 drop-down list of the second row and completing the condition, just as you did earlier.

**Note:** 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 let the user filter the list of records

1. Open the existing database component for editing. (See [Edit a Database Component](#).)
2. Click the **Options** tab. The window changes to show several options that control how the database component works.
3. Select **Allow filtering by end user**.

### Notes:

- 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 in one of two places: directly in the ADO connection string or in the **Options** tab of the database component. Connection strings, user names, and passwords are saved and encrypted in the component file (for safety purposes).

**Warning:** ADO connection strings are always displayed in plain text in the **Database Editor**. If you choose to store passwords in the connection string, anyone with component editing capabilities will be able to view the database password in plain text. Passwords entered at the **Options** tab are masked.

## 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** box.
4. Type the password in the **Password** box.

## To specify a user name and password in the ADO connection string

1. Open the database component for editing. (See [Edit a Database Component](#).) The **Database Editor** appears.
2. At the **Properties** tab, click the  **Edit** button next to the **Connection string** box. The **Data Link Properties** dialog box appears.
3. Click the **Connection** tab. The window changes to show several connection options, including user names and passwords.
4. Type the user name in the **User name** box and the password in the **Password** box. (You may need to clear the **Blank password** option.)
5. Select **Allow saving password**.

ADO connection strings stored within the component file are encrypted, but ADO connection strings stored in Microsoft Data Link (.UDL) files are not. Therefore, if you choose to store your connection information in a data link file, you should still specify the user name and password in the database component to keep them secure.

**Note:** 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** (A to Z, 1 to 9) or **Descending** (Z to A, 9 to 1).
5. Optionally, to sort on a second level, select a field at the **Then by** drop-down list and then choose **Ascending** (A to Z, 1 to 9) or **Descending** (Z to A, 9 to 1) .

## To let the user sort the list of records

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 database component options.
3. Select **Allow sorting by end user**.

### Notes:

- If you don't want the user to alphanumerically sort the records, clear **Allow sorting by end user**. The records in the table will then be presented to the user and merged into the assembled document (if you've allowed for multiple selection) in the order you specify.
- To remove any sorting options, click the **Sort by** drop-down button and select the blank line at the top of the list.
- If you are sorting a *repeated* database component, you can also assign sorting instructions at the **REPEAT Field** dialog box. For information about doing this, see [Tips on Filtering and Sorting Repeated Database Components](#).

# 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

1. 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  **Edit 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 REPEAT Database Instructions](#) for information about doing this.)
8. Click **OK**. The **REPEAT** instruction is inserted into the template.

If you have used a database filter to generate your list of records, you can have your answers automatically selected and merged into the assembled document. For details, see [Merge Filtered Records into the Document Automatically](#).

# 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 [Specify Connection Information 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.

**Note:** 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.

## Notes:

■ 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 All 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 Database Tables Appear

When HotDocs creates an interview for the user, it reads through the template and displays dialogs based on the variables it finds in the template text. If a variable is linked to a database table, HotDocs displays the database table instead.

If you want your database table to appear at 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 [Specify Connection Information Using ADO](#) and [Link Variables to Database Fields](#).)
5. Click **OK**. The **ASK Database** instruction is inserted in the template.

**Note:** 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	Server Cursors
<p>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 <i>advantages</i> of this are:</p> <ul style="list-style-type: none"> <li>■ End users will have fast scrolling and full ability to change sort order.</li> <li>■ The data is only pulled across the network one time.</li> <li>■ 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.</li> </ul> <p>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 <a href="#">Limit the Number of Database Records from Which a User Chooses.</a>)</p>	<p>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 <i>disadvantages</i> of a server cursor are:</p> <ul style="list-style-type: none"> <li>■ User scrolling can be slower, as only the displayed data is retrieved from the database at one time.</li> <li>■ 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.</li> <li>■ The user cannot change the sort order of the displayed results.</li> <li>■ If the user applies filters to very large result sets, performance and network bandwidth usage can be negatively impacted.</li> </ul> <p>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.</p> <p><b>Warning:</b> Because of these limitations, it is recommended you use client cursors—unless you have a specific reason to choose otherwise.</p>

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.

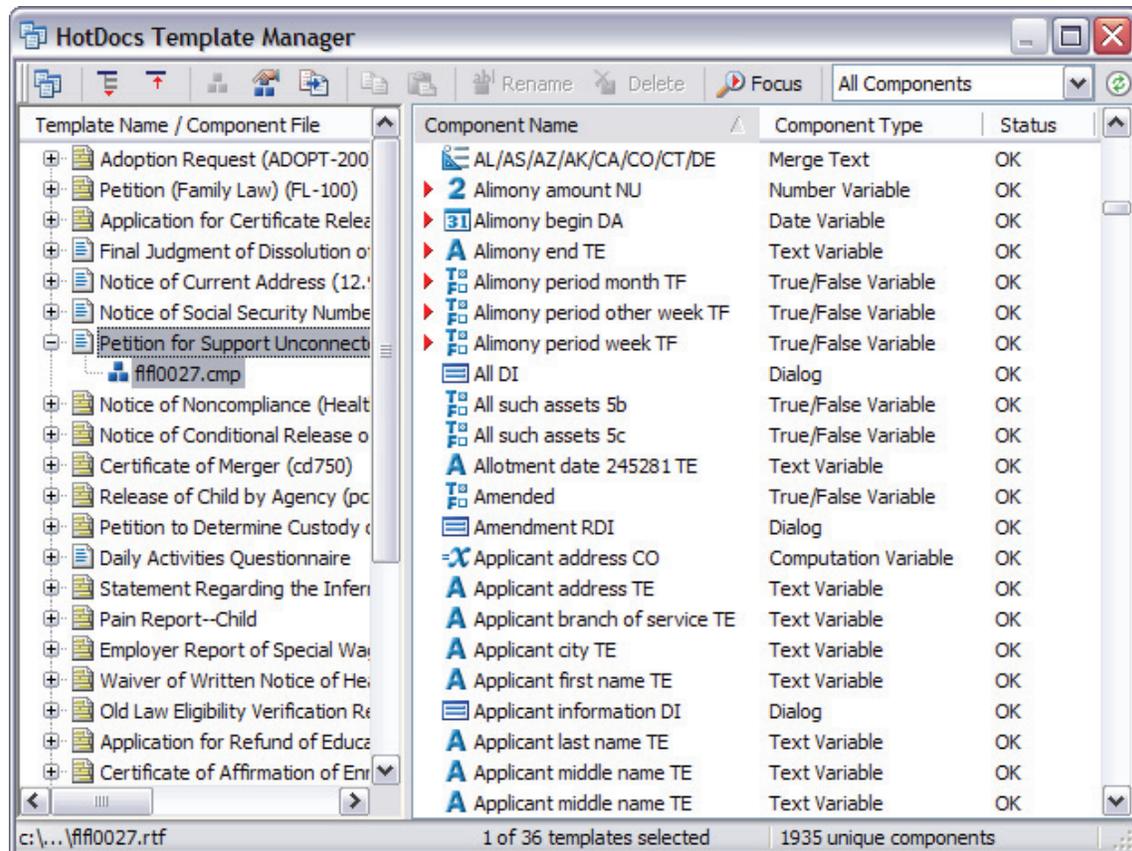
3. Choose one of the following options:
  - Select **Use disconnected (client) record set** to use a client cursor.
  - Clear **Use disconnected (client) record set** to use a server cursor.

# Using Template Manager

## Overview: 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.



## 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 2008 format or to RTF. 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.

# 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.

**Warning:** 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

1. 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 2008.
  - Convert templates and clauses to Microsoft RTF.
  - Rename templates.
  - View the contents of the explored component files.
  - Copy and paste components.
  - Rename components.
  - Delete components.

## To exit Template Manager

- At the Template Manager window, click the **X** in the upper-right corner of the window.

# View the Contents of a 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.

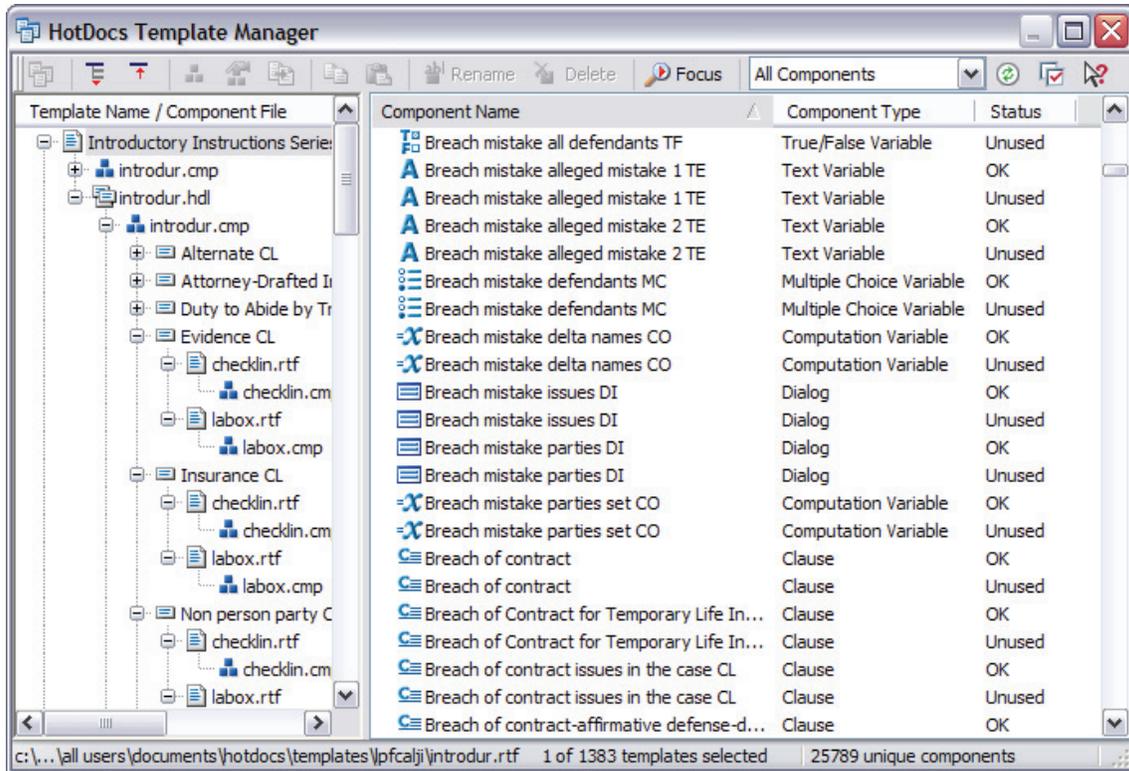
Together, the file list and component list show relationships between files and components. For example, when you select a file, Template Manager displays a  red marker next to the components used in the selected file. Likewise, when you select a component, HotDocs displays a  red marker next to the 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 [Work with Template Manager](#) 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  **Expand** button. (Click the  **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  icon.)

In this example, the template, *Introductory Instructions Series*, is expanded to show the template's component file (*Introduer.cmp*) and an inserted template (*Table.rtf*, which includes an ASSEMBLE instruction to assemble *Table1.rtf*). It also shows that a clause library with several clauses has been inserted (*Introduer.hd*).



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.

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 2008 format.
- Convert multiple templates to Microsoft RTF.
- Rename templates using Template Manager.
- Modify component file properties across multiple files.

## 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.

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.

**Note:** To access the Template Manager toolbar using the keyboard, press **F10**.

# Work with Template Manager

There are several things you can do to customize the view of Template Manager:

- Select specific files from the file list and click the  **New Window** button to limit the number of files and components you are viewing. HotDocs creates a new Template Manager window that shows only the selected files and their corresponding components. 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.
- Select a specific type from the **Components** drop-down list to show only specific component types.
- Select a file or files and click  **Focus** to show only the components for a selected template or group of templates. (Click  **Focus** again to show *all* components.)
- Click the  **Refresh Database** button to have HotDocs regenerate the component database, which updates the contents of the Template Manager window.
- Click the  **Template Manager Options** button to specify some specific settings for what information Template Manager displays in the component list.

**Note:** To access the Template Manager toolbar using the keyboard, press **F10**.

# 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.

# Convert Multiple Templates to Work with HotDocs 2008

**Warning:** 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 2008.

HotDocs 2008 uses a file format for component files that is different from HotDocs 5, HotDocs 6, and HotDocs 2005. (HotDocs 2006, HotDocs 2007, and HotDocs 2008 all use the same file format.) This means that all component files created in these earlier versions must be converted to HotDocs 2006-2008 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 2008 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 2006-2008** option is always selected by default. (This is correct.)
7. Optionally, select one of the options to convert the Word or WordPerfect templates to RTF. (See [Convert Templates and Clauses to Microsoft RTF](#) for details about this kind of conversion.)
8. Click **OK**. HotDocs converts all the selected files (and any dependent files referenced by the templates) to HotDocs 2006-2008 format.

# Convert Templates and Clauses to Microsoft RTF

**Warning:** Any time you make major changes to templates, component files, or other HotDocs files, it is a good idea to back up the files before making the change. Also, to convert templates or clause libraries to any file type other than RTF, you must convert each individual file manually. See [Convert a Single Template to a New File Format](#) and [Convert Clauses to New File Formats](#) for details.

If you change your word processor brand from WordPerfect to Microsoft Word, you can convert your templates and clause libraries to Microsoft Word RTF using Template Manager.

If you are already using Microsoft Word but your templates or clauses are in DOT format, you should consider converting them to RTF. This is because HotDocs assembles all Word templates using an RTF-based assembly process. Therefore, if your template or clause library is in DOT format, HotDocs must open it, convert it to RTF, and then assemble it. By automating the templates (and clauses) in RTF, 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 RTF 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.

**Warning:** 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 RTF

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 to convert and click the  **Convert Templates** button. The **Convert Files** dialog box appears, with the **Convert component files to work with HotDocs 2006-2008** option selected by default. (See [Convert Multiple Templates to Work with HotDocs 2008](#).)
5. Select **Convert WordPerfect templates and clauses to MS Word RTF** to convert your WPTs to RTF.

6. Select **Convert MS Word DOT templates and clauses to RTF** to convert your DOTs to RTF. HotDocs lists the files that will be converted in the box below these options.
7. Click **OK**. Template Manager confirms that you want to convert the files.
8. 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.

# 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.

**Note:** 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** box. 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.

**Note:** 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 re-point it to the shared file.

# 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.
2. From the file list, select the file or files you want to copy the component into.
3. 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.
4. Select additional component files from the list, clear the selected files, or accept the current selection.
5. 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.
6. Click **Paste**. Template Manager pastes the component into the file or files.

**Note:** You can also drag components from the component list to selected files in the file list to copy and paste them.

# 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** box.
3. From the **Following files** box, select the files you want to make the change in. (This box lists all of the files that contain the component or a reference to the component.)

**Warning:** 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.

**Warning:** If you delete a component that is either still referenced in the template or by another component, it deletes the component from the component file, but does *not* remove the reference. The component still appears in the component list in Template Manager; however, its status is **Missing**. You cannot assemble a document from a template that contains an  unresolved reference.

Components you remove using Template Manager are permanently deleted from the selected component files. (See [View the Component's Status](#).)

## 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.

# Publishing Templates

---

## Overview: 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 [Overview: Upload Published 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.

# 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.)

## To publish templates as standard template files

1. 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. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder where you want HotDocs to save the published files.

**Warning:** 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.

5. Optionally, if you plan to upload the published files, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server](#).)
6. Click **Next**. The **File Security** dialog box appears.
7. Optionally, specify any security options and click **Next**. (See [Protect Published Files](#) for details.) The **Additional Files** dialog box appears.
8. 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.)
9. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

**Note:** 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](#).)

# Publish a Template or a Template Set as an Auto-Install File (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.

## To publish one or more templates as individual auto-install files

1. 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 Individual auto-install files (HDI) 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. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder where you want HotDocs to save the published files.
5. Optionally, if you plan to upload the published files, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server](#).)
6. Click **Next**. The **File Security** dialog box appears.
7. Optionally, specify any security options and click **Next**. (See [Protect Published Files](#).) The **Auto-Install Target Library** dialog box appears.
8. 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.
9. Click **Next**. The **Additional Files** dialog box appears.
10. 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](#).)
11. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

## To publish a set of templates in a single auto-install file

1. 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 set in a single auto-install file (HDI) from the **Publish as** group.
3. If any of your templates contain INSERT or ASSEMBLE instructions, select **Include templates used in INSERT and ASSEMBLE instructions**.
4. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder where you want HotDocs to save the published files.
5. Optionally, if you plan to upload the published files, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server](#).)
6. Click **Next**. The **File Security** dialog box appears.
7. Optionally, specify any security options and click **Next**. (See [Protect Published Files](#).) The **Published Template Set** dialog box appears.
8. Enter a name for the HDI file in the **File name for published HDI file** box, and then enter a title for the file in the **Template set title** box.
9. Click **Next**. The **Auto-Install Target Library** dialog box appears.
10. 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.
11. Click **Next**. The **Target Library Shortcut** dialog box appears.
12. 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](#).)
13. Click **Next**. The **Additional Files** dialog box appears.
14. 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](#).)
15. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

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.

**Warning:** 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.

**Notes:**

- 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 Professional or Standard Edition. To keep users from editing the template, specify the **Lock Component Files** option at the **File Security** dialog box.
- 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](#).)

# Publish a Template or a Template Set as an Auto-Assemble File (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 [Upload Templates 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.

## To publish one or multiple templates as individual auto-assemble files

1. 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 **Individual auto-assemble files (HDA)** 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. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder where you want HotDocs to save the published files.

**Warning:** 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.

5. Optionally, if you plan to upload the published files, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server.](#))
6. Click **Next**. The **File Security** dialog box appears.
7. Optionally, specify any security options and click **Next**. (See [Protect Published Files for details.](#)) The **Additional Files** dialog box appears.
8. 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.](#))
9. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

## To publish a set of templates in a single auto-assemble file

1. At the HotDocs library, **select** the individual templates or folder of templates you want to publish and click the  **Publishing Wizard** button. The **Publishing Wizard** dialog box appears, showing the **Publishing Options** information.
2. Select **Template set in a single auto-assemble file (HDA)** 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. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder

where you want HotDocs to save the published files.

**Warning:** 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.

5. Optionally, if you plan to upload the published files, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server](#).)
6. Click **Next**. The **File Security** dialog box appears.
7. Optionally, specify any security options and click **Next**. (See [Protect Published Files](#) for details.) The **Published Template Set** dialog box appears.
8. Enter a name for the HDA file in the **File name for published HDA file** box, and then enter a title for the file in the **Template set title** box.
9. Click **Next**. The **Additional Files** dialog box appears.
10. 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.)
11. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

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 or Netscape Navigator, you must register HotDocs to work with the Web browser. (See [Register HotDocs to Work with Web Browsers](#).)

**Note:** 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](#).)

# Publish Templates for Use with HotDocs Server

For each template you create for Web-based assembly using HotDocs Server, you must also create a HotDocs variable collection (.HVC) file, as well as an interview definition (.JS) file. The variable collection file stores all of the variables used in a given template, while the interview definition file defines (in JavaScript) how a Web browser displays the interview. In order to generate these files, you must use the Publishing Wizard to publish your templates for use with HotDocs Server. Then, once published, all these files must be uploaded to a Web server running HotDocs Server. (See [Overview: Upload Published Files to a Web Server](#) for details.)

## To publish templates for use with HotDocs Server

1. Enable all templates you want to publish for Web-based assembly. (See [Enable Templates for Use with HotDocs Server](#).)
2. 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.
3. Select **Template files for use with HotDocs Server** from the **Publish as** group
4. Click the  **Browse** button next to the **Local folder for published files** box to specify the folder where you want HotDocs to save the published files.
5. Optionally, if you want to upload the published files at the end of the Publishing Wizard, click the  **Edit** button next to the **Web destination for published files** box 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 [Overview: Upload HotDocs Files to a Web Server](#).)
6. Click **Next**. The **Additional Files** dialog box appears.
7. Click **Finish**. HotDocs publishes your files and returns you to the HotDocs library.

### Notes:

- As you are publishing your files, you may receive a HotDocs Publishing error. This is because your template contains automation that is unsupported in HotDocs Server. When HotDocs detects such a feature, it displays an error message, describing the error it found. You cannot publish the files until you remove the feature that is causing the error. (See [Differences Between Desktop and Browser-based Interviews](#).)
- If you're publishing an interview template (or .CMP file), you must make sure the component file has the **Use INTERVIEW computation** component file property set for it. (See [Create an Interview Template and Change Component File Properties](#).)

**Warning:** 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.

# 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 [Register Published Templates for Use with HotDocs Player](#).)

**Warning:** If your users upgrade to HotDocs Standard or Professional and then attempt to edit any published templates you have registered, those templates will no longer work with HotDocs Player. Additionally, HotDocs PDF Advantage is required to assemble unregistered PDF templates using Standard or Professional. Thus, users who upgrade without purchasing a license for PDF Advantage will lose the ability to assemble PDF templates if the component files are modified. 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** box.
  - **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** box.
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 box, or click the calendar icon next to the box 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 box. (HotDocs automatically includes information about the expiration date.)

# Register Published Templates for Use With HotDocs Player

Often, when you want to distribute your templates to other users, you do so by publishing them in standard format, auto-install (HDI) format, and auto-assemble (HDA) format. Once published, any HotDocs Professional or Standard user can assemble documents using these files. A HotDocs Player user, however, cannot. This is because the templates have not been *registered* for use with HotDocs Player.

In order to provide templates that HotDocs Player users can use, you must register the templates. This requires you to have a special publishing key file, obtainable only through a special licensing agreement with HotDocs Ltd. Once obtained and saved to the HotDocs program folder, you can select the registration option during the publishing process and provide a password. Then, upon completing the publishing process, the templates will be registered and you can freely distribute your template set to HotDocs Player users.

## Acquiring a Publishing Key File

You obtain a publisher key file from HotDocs Ltd. by entering into a publishing contract, which normally assesses a fee for each registered template that is distributed. Once the contract is in place, you will be provided with a key file (Register.hdk). Typically, you may use these key files for about a year before you need to renew your license. Please contact a HotDocs sales representative for specific licensing details.

## Publishing and Registering Templates

Once you have obtained a publisher key file, you can register your templates so they can be used with HotDocs Player. Registration happens during the publishing process.

### To publish your templates for use with HotDocs Player

1. Copy **Register.hdk** to the HotDocs program folder (for example, *C:\Program Files\HotDocs 6*).
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.
4. Select **Register component files** and enter the password for the publisher key file in the **Password** box. (You should have received this password from the HotDocs 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.

**Note:** 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.

# 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  **Browse** button next to the **Icon to use** box and locate the icon (.ICO) file you want to assign to the library. (If you leave this box empty, HotDocs will use a default icon.)
  - Type a name in the **Folder to contain shortcut** box. 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 **Start > Program Files > LexisNexis HotDocs > HotDocs Professional** program folder.)
  - Type a shortcut name in the **Label for shortcut** box. 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 a Template or a Template Set as an Auto-Install File \(HDI\)](#).)
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.

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.

# Add Additional 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 a Template or a Template Set as an Auto-Install File \(HDI\)](#), or [Publish a Template or a Template Set as an Auto-Assemble File \(HDA\)](#).)
2. 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 box.
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 a Template or a Template Set as an Auto-Install File \(HDI\)](#), or [Publish a Template or a Template Set as an Auto-Assemble File \(HDA\)](#).)
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** box. HotDocs adds the extension **.HDP** to the file.
4. Click **Save**. The file is saved to the HotDocs *Publish* folder.

## To use a published settings file

1. Publish your templates or your template set. (See [Publish a Template as a Standard File](#), [Publish a Template or a Template Set as an Auto-Install File \(HDI\)](#), or [Publish a Template or a Template Set as an Auto-Assemble File \(HDA\)](#).)
2. At the first dialog box, **Publishing Options**, click **Load Settings**. The **Load Publish Settings** dialog box appears.
3. Select the publish settings (.HDP) file and click **Load**.
4. Complete the publishing process.

# 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*.
2. 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.)

**Warning:** When a user opens a library by clicking the HotDocs button in the word processor toolbar, no splash screens are displayed.

# Customize the Properties Tab of the Library Window

When a user selects an item in a HotDocs template library, information about the selected item appears in the **Properties** tab of the window. By default, HotDocs displays this information in a generic page that HotDocs generates. However, you can create a custom HTML document (or MHT Web Archive) that replaces this generic page. Custom HTML property pages can be created for individual files in a library, for an entire library, or for all files loaded into HotDocs.

By customizing the **Properties** tab of the library, you can add company logos, as well as select your own formatting and styles for the information that appears. You also have greater control over the information that appears in the tab by removing specific items. Specifically:

- To create a custom HTML properties page for a specific template, the HTML document must have the same name as the template, with an HTML or MHT file name extension—for example, Subpoena.htm or Subpoena.mht. This file must be saved in the same folder as the template.
- To create a custom HTML properties page for an entire library, the HTML document must have the same name as the library file, with an HTML or MHT file name extension—for example, Real\_estate.htm or Real\_estate.mht. It must be saved in the same folder as the library file.
- To create a custom HTML properties page for just the main folder in the library, the HTML document must have the same name as the library, followed by the word main—for example, Estate\_Planning\_Main.htm or Estate\_Planning\_Main.mht. It must be saved in the same folder as the library file.
- To create a custom HTML properties page for HotDocs to always use, the HTML document must be named *Properties.htm* or *Properties.mht*, and it must be saved in the default *Libraries* folder (for example, *C:\Documents and Settings\UserName\My Documents\HotDocs\Libraries*).

When a user selects an item in the library, HotDocs determines whether a custom properties page has been specified to display the properties for the item. To determine this, HotDocs searches first for an HTML page that shares the same file name as the selected file. If nothing is found, HotDocs then searches for an HTML page that uses the same name as the library file. If this search doesn't yield anything, HotDocs searches for the file *Properties.htm* (or *Properties.mht*). Finally, if no custom HTML page exists, HotDocs generates its own properties page.

When the **Properties** tab is customized and you select an item in the library, HotDocs takes the information it knows about the item (such as its title or description) and, using HTML SPAN tags, substitutes that information into the HTML document and displays it instead of the default **Properties** tab.

The following is sample HTML code you can customize to adapt the **Properties** tab of the HotDocs template library for your needs:

## Sample HTML Code

```
<html>

<body>

<table width="100%">

<!-- This section of HTML code controls the properties for selected
files in the library. You can format the text, remove properties
you don't want displayed, and so forth. -->

<SPAN id="ID_FILE">

  <tr>

    <td width="100%" colspan="2">

      <SPAN id="ID_HEADING"></SPAN>
```

```

        </td>
</tr>
<tr>
    <td width="5%">
        <p align="right">Title:</p>
    </td>
    <td width="95%"><SPAN id="ID_TITLE"></SPAN></td>
</tr>
<tr>
    <td width="5%">
        <p align="right">Type:</p>
    </td>
    <td width="95%"><SPAN id="ID_TYPE"></SPAN></td>
</tr>
<SPAN id="ID_FILENAME_ROW">
    <tr>
        <td width="5%">
            <p align="right">File name:</p>
        </td>
        <td width="95%"><SPAN id="ID_FILENAME"></SPAN></td>
    </tr>
</SPAN>
<tr>
    <td width="5%">
        <p align="right"><SPAN id="ID_FILE_PATH_TEXT">File
path:</SPAN></p>
    </td>
    <td width="95%"><SPAN id="ID_FILEPATH"></SPAN></td>
</tr>
<tr>
    <td width="5%">
        <p align="right">Description:</p>
    </td>

```

```

        <td width="95%"><SPAN id="ID_DESCRIPTION"></SPAN></td>
    </tr>
</SPAN>

<!-- This next section of HTML code controls the properties for
selected folders in the library. You can format the text, remove
properties you don't want displayed, and so forth. -->
<SPAN id="ID_FOLDER">
    <tr>
        <td width="100%" colspan="2">
            <SPAN id="ID_HEADING"></SPAN>
        </td>
    </tr>
    <tr>
        <td width="10%">
            <p align="right">Title:</p>
        </td>
        <td width="90%"><SPAN id="ID_TITLE"></SPAN></td>
    </tr>
    <tr>
        <td width="10%">
            <p align="right">Description:</p>
        </td>
        <td width="90%"><SPAN id="ID_DESCRIPTION"></SPAN></td>
    </tr>
</SPAN>
</table>
</body>
</html>

```

**Note:** If you publish your templates as HotDocs Auto-Install (.HDI) files, you can include the custom HTML properties pages as additional files in the HDI. The custom properties pages will then be installed to the same folder as the library (for pages that affect an entire library) or the *Templates* folder (for template-specific pages). When installing custom properties pages that affect an entire library, however, you *must* use Web Archive (.MHT) files instead of HTML files to ensure that HotDocs will install them to the same folder as the library.

# Automatically Update Published Template Sets

**Warning:** 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 File

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 <b>1</b> , but future versions of HotDocs may support additional values.
feedurl	A URL for the update document (RSS feed). For example, <b>http://www.yourcompany.com/update.rss</b> .
checkinterval	The number of days between each update check. If the interval is <b>0</b> , 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: <b>Sun, 03 Jun 1990 00:00:00 GMT</b> .
title	The title of the template set. For example, <b>Hobble Creek Real Estate Forms</b> .
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 ( <b>true</b> ) or not ( <b>false</b> ). Users can change this value at HotDocs Options ( <b>Tools &gt;</b>

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

```
<?xml version="1.0"?>
<updatecatalog>
  <formatversion>1</formatversion>
  <feedurl>http://www.HobbleCreek.com/update.rss</feedurl>
  <checkinterval>7</checkinterval>
  <publishdate>Sun, 03 Jun 1990 00:00:00 GMT</publishdate>
  <title>Hobble Creek Real Estate Forms</title>
  <enabled>true</enabled>
  <password></password>
  <notinstalleditems/>
  <discardeditems/>
  <installeditems/>
</updatecatalog>
```

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.

**Warning:** 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, <b>Hobble Creek Real Estate Forms</b> .

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 <b>description</b> 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: <b>File Update:</b> Installs an .HDI file. <b>Message:</b> 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 .HDI file. The date must be in the following format: <b>Sun, 03 Jun 1990 00:00:00 GMT</b> .

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.

```
<?xml version="1.0"?>
```

```
<rss version="2.0">
<channel>
<title>Hobble Creek Real Estate Forms</title>
<link>http://www.HobbleCreek.com/update.rss</link>
<description>Serving you since 1994</description>
<item required="true">
  <title>Real Estate Purchase Contract (REPC)</title>
  <link>http://www.HobbleCreek.com/repc.hdi</link>
  <description>This is the new form adopted by the State Real
Estate Commission.</description>
  <guid>00001</guid>
  <category>File Update</category>
  <pubDate>Tue, 21 Mar 2006 15:27:18 GMT</pubDate>
</item>
<item>
  <title>Visit the new Hobble Creek Real Estate Web Site</title>
  <link>http://www.HobbleCreek.com</link>
  <description>New features are added daily</description>
  <guid>00002</guid>
  <category>Message</category>
  <pubDate>Tue, 21 Mar 2006 15:27:18 GMT</pubDate>
</item>
<item>
  <title>Happy Holidays</title>
  <link></link>
  <description>Wishing you a happy holiday season from your friends
at Hobble Creek Real Estate.</description>
  <guid>00003</guid>
  <category>Message</category>
  <pubDate>Tue, 21 Mar 2006 15:27:18 GMT</pubDate>
</item>
</channel>
</rss>
```

# Uploading Published Files to a Web Server

---

## Overview: Upload HotDocs Files to a Web Server

You can upload HotDocs template files to a Web server. These include standard template files, auto-assemble 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.

# 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.

**Warning:** 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 *OhdUploadDebug.htm*, located in the *Local Settings\Temp* folder (for example, *C:\Documents and Settings\<username>\Local Settings\Temp\OhdUploadDebug.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.

See [Publishing Form Page Sample Code](#) 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.

See [Publishing Destination Page Sample Code](#) for an example of Publishing Destination Page. (This page uses the Dundas Upload control.)

# 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 template set. (See [Publish a Template as a Standard File](#), [Publish a Template or a Template Set as an Auto-Install File \(HDI\)](#), [Publish a Template or a Template Set as an Auto-Assemble File \(HDA\)](#).)
2. At the **Publishing Options** dialog box, click the  **Edit** button. The **Web Destination** dialog box appears.
3. Type an absolute URL in the **URL** box, or click the  **Browse** button to locate the URL.
4. Type a description of the Web site in the **Friendly name** box, 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.

**Note:** 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 Answer Files

---

## Overview: 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.

**Warning:** 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  **Properties** button. The **Component File Properties** dialog box appears.
3. Type the URL of the Web page to which you want the answers uploaded in the **Upload URL** box.
4. Type a description of the page to which the URL points in the **Friendly name** box.
5. 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 Type
T	Text
N	Number
D	Date
B	True/False (Boolean)
M	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.

Type	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`

### Notes:

- You can use HotDocs Server to process uploaded answers. See [Sample Code for Uploading an Answer File Using HotDocs Server](#) for an example of an ASP page that processes uploaded answers using HotDocs Server. (Contact your HotDocs sales representative for information on HotDocs Server.)
- 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.

# Using Command-Line Options

---

## Overview: Command-Line Options

You can use command-line options to control the way HotDocs assembles text and form documents. Most options can be placed 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, and for the program file's shortcut.

### 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 6\hotdocs6.exe**" (including the quotation marks) followed by a space and the options you want in the **Open** box. For example:

```
"C:\Program Files\HotDocs 6\Hotdocs6.exe" /tf=demoempl.rtf
```

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.

### 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** box, enter a space after the executable (.EXE) file and type the options you want.

### 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** box, 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.ans"
```

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.

#### Warnings:

- When HotDocs is started from the command line, it will continue to run, even after the processing of the command-line request is completed. To tell HotDocs to close after processing the command-line request, use the Exit HotDocs option.
- If you are using multiple options on a single command line, you must separate each option with a space character. If the command line includes space characters, you must enclose the path in quotation marks.

The following is a complete list of command-line options:

## Command-Line Options

Answer File ( /af)	Overlay Answer File ( /ov)
Answer Summary ( /as)	Paper Size ( /ps)
Clause Name ( /cl)	Paper Tray ( /pt)
Default Answer File ( /df)	Print ( /pr)
Discard Answers ( /da)	Print Answers Only ( /pa)
Edit Template ( /ed)	Print Both ( /pb)
Exit HotDocs ( /ex)	Print Copies ( /pc)
Finish Interview Action /fia	Print Duplex ( /pd)
Hide Library ( /hl)	Print Form Only ( /po)
HotDocs Auto-Assemble File ( /ha)	Print Without Dialogs ( /pw)
HotDocs Auto-Install File ( /hi)	Question Summary ( /qs)
Interview Scope ( /is=u)	Save Answers ( /sa)
Keep Interview Group ( /kig)	Save Answers Prompt ( /sap)
Library File ( /f)	Send to Word Processor ( /stw)
Lock Answer File ( /la)	Show Library ( /sl)
Lock Library ( /ll)	Start Interview Group ( /sig)
Model Document ( /mo)	Suggest Save ( /ss)
New Answer File ( /na)	Suggest Save New ( /ssn)
No Assembly Window ( /nw)	Suppress Unanswered Warning ( /sw)
No Interview ( /ni)	Template File ( /tf)
Output File ( /of)	

# 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.ans`). (You cannot, however, save an answer file back to the server.)

## Notes:

- You cannot assign the **Answer File** option to a HotDocs Auto-Assemble file.
- If using this option at the command line, include the Template File 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.

**Note:** If using this option at the command line, include the Template File and Output File options. If using this option at the library properties, include the Output File 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.ans**).

**Note:** If using this option at the command line, include the Template File 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.

## Notes:

- To disable all answer file usage (saving, selecting new, and so forth) during a given assembly, use the Lock Answer File option.
- If using this option at the command line, include the Template File option.

# 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 at the End of the Interview.](#))

# 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](#).)

**Note:** If using this option at the command line, include the Template File 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 auto-assemble (.HDA) file.

You can also specify a URL for the path and file name (for example, **/ha=<http://www.yoursite.com/hdfile.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.

# 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.

**Warning:** 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.

# 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.

**Note:** If using this option at the command line, include the Template File 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.

# Model Document

/mo

The **Model Document** command-line option indicates that the file referenced in the library is a model document. When you select the document in the library and click  **Assemble**, HotDocs will create an interview for the model document.

**Note:** See [Overview: Create a Model Document](#) for more information.

## 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 a file name extension, such as .ANS or .ANX.

**Note:** If using this option at the command line, include the Template File option.

# No Assembly Window

`/nw`

The **No Assembly Window** option causes HotDocs to assemble a document without displaying the assembly window. You must use the Output File option with it.

**Note:** If using this option at the command line, include the Template File, Answer File, and Output File options. If using this option at the library properties, include the Answer File and Output File options.

# 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.

## Notes:

- 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 and Answer File options. If using this option at the library properties, include the Answer File 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 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.

**Note:** If using this option at the command line, include the Template File option.

# 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.ans**).

## Notes:

- 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 option.

# Paper Size

`/ps=paper size`

**Warning:** This option is used with form documents only.

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.

## Notes:

- If a paper size is specified at the **Printing Properties** dialog box and the **Paper Size** command-line option is also used, the command-line option takes precedence.
- If using this option at the command line, include the Template File option.

# Paper Tray

`/pt=paper tray`

**Warning:** 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.

## Notes:

- If a paper source is specified at the **Printing Properties** dialog box and the **Paper Tray** command-line option is also used, the command-line option takes precedence.
- If using this option at the command line, include the Template File option.

# Print

`/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.

**Note:** If using this option at the command line, include the Template File option.

# Print Answers Only

/pa

**Warning:** 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.

**Note:** If using this option at the command line, include the Template File option.

# Print Both

/pb

**Warning:** 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.

**Note:** If using this option at the command line, include the Template File option.

# Print Copies

*/pc=number of copies*

**Warning:** 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** box at the **Print** dialog box.

**Note:** If using this option at the command line, include the Template File option.

# Print Duplex

/pd

**Warning:** 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.

**Note:** If using this option at the command line, include the Template File option.

# Print Form Only

/po

**Warning:** 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.

**Note:** If using this option at the command line, include the Template File option.

# Print Without Dialogs

/pw

**Warning:** 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.

**Note:** If using this option at the command line, include the Template File 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.

**Note:** If using this option at the command line, include the Template File option.

# Save Answers

/sa

This option, which is used in connection with an ASSEMBLE instruction, 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.

## Notes:

- 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 option.
- To give users the option of saving an answer file, rather than forcing them to save, use either the Suggest Save or the Suggest Save New option.

# Save Answers Prompt

`/sap`

This 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.

## Notes:

- 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 option.
- To give users the option of saving an answer file, rather than forcing them to save, use either the Suggest Save or the Suggest Save New 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.

**Note:** If using this option at the command line, include the Template File option.

# Show Library

/s1

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

This 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.

## Notes:

- 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 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 or the Save Answers Prompt options.

# Suggest Save New

`/ssn`

This 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.

## Notes:

- 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 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 or the Save Answers Prompt 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.

**Warning:** 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.

# Assembling Documents

---

## 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 [Overview: 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.

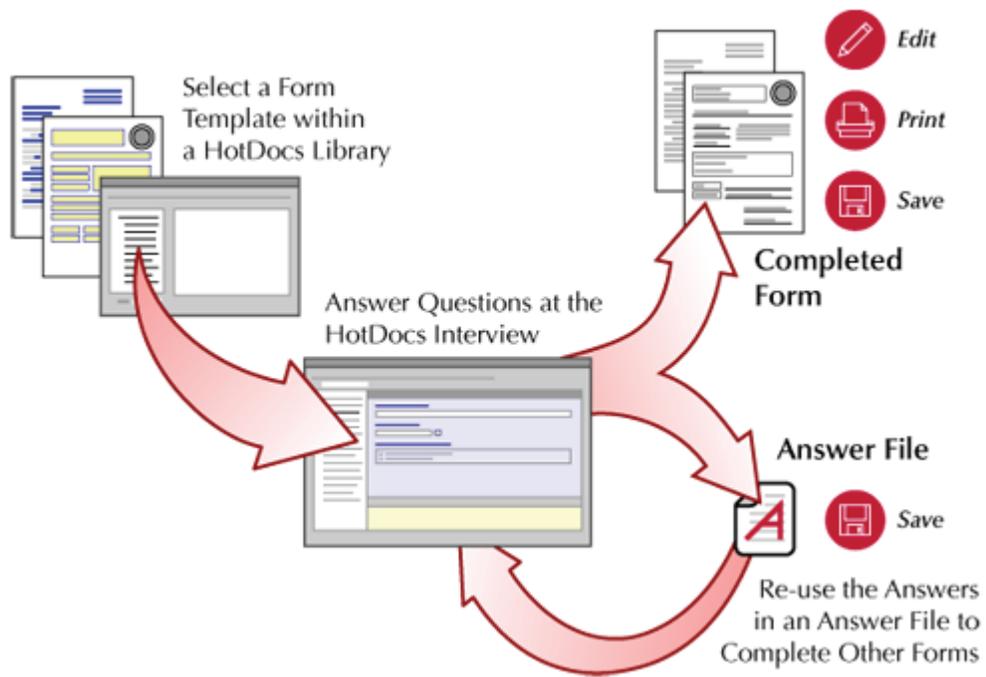
As you answer questions, HotDocs merges your answers into the assembled document, which you can view by clicking the **Document** tab at the assembly window. As you are reviewing the assembled document, if you see an answer that needs to be edited or changed, you can make the change right in the **Document 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.

Finally, if you have installed HotDocs® Compare, you can assemble different versions of your document and compare them against each other. This may be useful if you need to see how answering questions in the interview differently can affect the outcome of the document. (*HotDocs Compare is available to Microsoft Word users only.*)

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:



# Adjust HotDocs Windows and Panes

You can arrange the HotDocs windows to match your work preferences.

## To move a window

- Click on the title bar and drag the window to another location.

## To resize a window

1. There are three ways to adjust the size of the HotDocs windows:
  - Click the **Maximize** button to make the window fill the entire screen.
  - Click the **Minimize** button to hide the window on the Windows taskbar. (You can restore the minimized window by clicking the appropriate icon on the taskbar.)
  - 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 proportions.

## 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.

**Note:** 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. 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  **Instant Update** off.

## To control when HotDocs updates the interview outline or document

1. While viewing either the **Interview** tab or the **Document** tab at the assembly window, click the  **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).

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 will update 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.)

# 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.

### Notes:

- 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.

# 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.
2. In the **Find** box, 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 box is not case-sensitive.)
3. To view the entire library again, clear the **Find** check box.

**Note:** 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 [Overview: Assemble a Form Document](#) for additional instructions.)
2. Click  **Assemble**. 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.
4. 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**.)
5. 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.

**Note:** 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](#).)

# 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  **Select**.
  - 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 select this answer file for all subsequent documents you assemble.
3. At the **Answer File** dialog box, click **OK**. The assembly window appears.

### Notes:

- You can also use Windows Explorer to access your answer files. See [Overview: Use Answer Management](#) and [Manage Answer Files and Assembled Documents](#).
- 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 is distinguished from other templates in the library by its icon: . 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.

For example, a template library might include an interview template named *Client Information*. This template could gather personal information about each client—such as name, address, gender, birth date, and so forth. 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 requires client information, you can access the answers you've already entered. (See [Answer Questions in a Dialog Using an Answer Source](#).)

## 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. Instead, skip to step 3.
2. Select an answer file, then click **OK**. (See [Select an Answer File for Assembly](#).) HotDocs displays the assembly window.
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. (See [Answer Questions in a Dialog Using an Answer Source](#) for information on using an answer source.)
5. Close the assembly window.

# Assemble a Document Without Answering Any Questions

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 at the End of the 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.

# Merge Existing Answers into 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. When an overlay answer file is loaded, it overwrites 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). These answers could be used in assembling 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 avoid this, 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. (See [Select an Answer File for Assembly](#).)
3. Click **Overlay Answers** (**File** menu). 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).

**Warning: 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. (See [Add, Modify, and Delete Folders in Answer File Manager](#).)

**Note:** You can also specify an overlay answer file at the command line. (See [Overview: Command-Line Options](#) and [Overlay Answer File](#).)

# 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**. (See [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.

**Note:** 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.

# 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  **Auto Start** button in the assembly queue toolbar. When assembling multiple templates and  **Auto 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  **Auto Start** is *not* selected, HotDocs starts the first assembly, but then waits for you to start each subsequent assembly by clicking the  **Assemble** 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, see [Assembly Queue Statuses](#).)

## To work with templates in the assembly queue

1. At the HotDocs library window, click the  **Assembly 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 assembly queue:

To	Do This
Change the order in which pending or confirmed templates are assembled	Select an entry and click the  <b>Up</b> or  <b>Down</b> button. Templates that are currently assembling cannot be moved in the queue.
Remove a confirmed assembly	Select the assembly and click  <b>Delete</b> . Templates that are currently assembling cannot be removed from the queue.
Change whether assemblies start automatically or manually	Click  <b>Auto Start</b> .  When <b>Auto Start</b> is turned off, clicking the  <b>Assemble</b> 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

	 <b>Assemble</b> button there.
Manually start an assembly	Select an entry in the assembly queue and click  <b>Assemble</b> . (If one template is already assembling, you cannot start another assembly.)
Remove the entries for all completed assemblies	Click  <b>Clean Up</b> .

**Notes:**

- 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 [Overview: 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. (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. (See [Select Clauses During an Interview](#) for instructions on using this method.)

## 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. (See [Select an Answer File for Assembly](#).)
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 box, 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.
4. 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. 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:** Click  **Send the assembled document to the Clipboard** (or press **Ctrl+F11**) at the *End of Interview* dialog. You can then open an existing document and paste the text in the desired location.

**Note:** 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 specify an option that causes HotDocs to update the interview less frequently. (See [Update Your Interview Outline](#).)

# Navigate Through Answer-Gathering Dialogs

To assemble a document, you must answer interview questions. To help you, 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.
-  Repeats 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 every interview is usually the *End of Interview* dialog, which reports how many questions are unanswered. This dialog also provides options for working with the assembled document. For information on these options, or to learn about hiding or customizing the *End of Interview* dialog, see [Use the End of Interview Dialog](#).

## To navigate through HotDocs dialogs

1. At the HotDocs library window, select a template and begin assembling the document. (See [Assemble a Text or Form Document](#).)
2. The HotDocs assembly window appears. Complete any of the tasks listed in the following table:

To	Do This
Enter an answer in a dialog	<p>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.</p> <p>When entering a multi-line text answer, you can control whether the <b>Enter</b> key inserts a line break (↵) or a paragraph mark (¶) in the assembled Word document. To enter a line break, simply press <b>Enter</b>. To enter a paragraph break, press <b>Ctrl+Enter</b>. (In some situations, the template developer may have specified which break to use.)</p> <p>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.</p>
Move to another dialog in any order	Click a dialog icon in the interview outline.

<p>Go to the next or previous dialog</p>	<p>Click either the  <b>Next Dialog</b> button or the  <b>Previous Dialog</b> button.</p> <p>You can also press <b>Page Down/Page Up</b> or <b>Alt+N/Alt+P</b>.</p>
<p>Go to the next dialog or the previous dialog that contains an unanswered variable</p>	<p>Click either the  <b>Next Unanswered Dialog</b> button or the  <b>Previous Unanswered Dialog</b> button.</p> <p>You can also press <b>Ctrl+Page Up/Ctrl+Page Down</b>.</p>
<p>Go to the first or the last dialog in the interview</p>	<p>Click either the  <b>First Dialog</b> button or the  <b>Last Dialog</b> button.</p>
<p>Finish the interview and view the assembled document</p>	<p>Click the  <b>Finish Interview</b> button.</p> <p><i>Where</i> you view the assembled document depends on the <i>Finish Interview</i> action you've specified. See <a href="#">Control What Happens When You Finish an Interview</a>.</p>
<p>View additional information about a specific answer or the contents of a dialog</p>	<p>Click the  <b>Resources</b> button. (See <a href="#">Access Resources During Assembly</a>.)</p>
<p>View where the answer is merged into the document (Word users only)</p>	<p>Click the  <b>Go to Answer in Document</b> button. HotDocs displays the <b>Document</b> tab and moves the cursor to the first place in the document where the answer is merged.</p>
<p>Hide the interview outline</p>	<p>Clear <b>Interview Outline</b> (<b>View</b> menu).</p>
<p>Preview an assembled text document before sending it to the word processor</p>	<p>Click the <b>Document Preview</b> tab. (See <a href="#">Preview the Assembled Text Document</a>.)</p>
<p>View an assembled form document</p>	<p>Click the <b>Form Document</b> tab. (See <a href="#">View the Assembled Form Document</a>.)</p>
<p>View a list of all questions in the template</p>	<p>Click the <b>Question Summary</b> tab. (See <a href="#">View a Question Summary</a>.)</p>

View a list of questions and answers	Click the <b>Answer Summary</b> tab. (See <a href="#">View an Answer Summary</a> .)
View a spreadsheet of the variables and their current answers	Click <b>Variable Sheet</b> ( <b>View</b> menu) and then click the <b>Variable Sheet</b> tab. (See <a href="#">View the Variable Sheet</a> .)
Cancel the assembly process or close the assembly window	Choose <b>Close</b> ( <b>File</b> menu).

**Notes:**

- HotDocs updates the interview as you enter answers in a dialog. If you experience a significant delay when moving from one answer field to the next, you can specify an option that causes HotDocs to update the interview less frequently. (See [Update Your Interview Outline and Document](#).)
- 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. (See [Warn When HotDocs Reformats Variable Answers](#).)
- To display the different tabs along the bottom of the window (instead of the top), clear **Tabs at Top** (**View** menu).

# Access 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.

**Note:** Resources help you answer questions in the interview. For help using HotDocs, see [Get Help While Using HotDocs](#).

# 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:

To	Do This
Choose the format for the answer file	Click the <b>Type</b> drop-down button and select an answer file format. Your options include <b>.ANS</b> (native HotDocs format) and <b>.ANX</b> (XML-based format).
Enter the file name	At the <b>File name</b> box, enter a name. When you click <b>OK</b> , 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  <b>Browse</b> button and navigate to the location.
Enter the title that identifies the answer file in Answer File Manager	At the <b>Title</b> box, enter a name or accept the suggestion HotDocs makes.
Add information to help identify the file	At the <b>Description</b> box, type notes about the answer file's purpose or contents. (The description will appear at the <b>Properties</b> tab of the Answer File Manager. When searching for specific answer files, you can search for text in the description.)

### Notes:

- See [Overview: 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](#).)
2. 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](#).)

The assembly window appears again, using the newly selected answer file.

**Note:** When viewing the answer library, you can sort the answer files. To do this, select a folder and click  **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 box.

# Answer Questions in a Dialog Using an Answer Source

Template providers can link questions in a dialog to an answer source, or an answer file that contains existing answers. When a dialog is linked to one of these sources, a  **Select** button appears on the dialog. You can click the  **Select** button to open the answer source and select a record. You can also add a new record to the answers.

Each record contains answers to one or more questions. For example, the records may contain the names, addresses, and phone numbers of attorneys in an office. After selecting one attorney's record, that attorney's information will be merged into the answer fields. This saves you from having to reenter answers you use frequently as you assemble documents.

## To access an answer source

1. At the HotDocs library window, select a template and begin assembling the document. (See [Assemble a Text or Form Document](#).)
2. When a  **Select** button appears on a dialog, click it. The list of answers appears.
3. Complete any of the tasks listed in the following table:

To	Do This
Use an existing set of answers	Select the desired record, then click the <b>Select</b> button.
Add a new record	Select an empty row in the spreadsheet and click  <b>Edit Row</b> . Then enter your answers in the pop-up interview.  You can either gradually build the list, entering a different record each time you complete an interview; or you can enter several records at once. You can delete, edit, or add new records at any time. Answers are automatically saved as you add them to the answer source.
Insert a record in between other records	Select the record you want to be <i>after</i> the new record and click  <b>Insert Row</b> . Then click  <b>Edit Row</b> to enter your answers.
Edit an existing record	Select the record and click  <b>Edit Row</b> . Make your changes in the pop-up interview.
Delete a record	Select the record and click  <b>Delete Row</b> .

	The answers are deleted.
Sort or alphabetize columns in an answer source	<p>Click in the column on which you want to sort, right-click and choose <b>Sort</b> from the shortcut menu. The <b>Sort Spreadsheet</b> dialog box appears. Designate your sorting preferences: <b>Ascending</b> sorts the specified column from 1 to 9, and A to Z, while <b>Descending</b> sorts from 9 to 1, and Z to A and click <b>OK</b>.</p> <p>To sort on a second level, enter your preferences in the <b>and then</b> fields.</p>

An answer source may be populated in different ways. For example, the template provider can either add answers to the answer source during development, or allow you to add answers during the interview. Additionally, an answer source may be an Outlook Contacts list or a Time Matters Contacts or Matters record, which the template provider has integrated with HotDocs.

**Note:** If your dialog is linked to a Time Matters Contacts or Matters answer source and you change answers once they have been retrieved, you may be prompted to write the answers back to Time Matters.

# 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.

## To enter answers in a series of dialogs

1. At the repeated dialog, enter answers for the questions.
2. Click  **Next** to continue to the next repetition. The repetition number next to the dialog icon increases each time you enter a new set of answers.
3. To move to the next dialog, leave all of the answer fields empty and click  **Next**. (You can also click on the next dialog icon, or press **Page Down** or **Alt+N**.)
4. Complete any of the following options:

To	Do This
Remove a specific set of answers from the interview	Select the specific dialog in the interview outline and click the  <b>Delete Repetition</b> button. That set of answers is permanently removed. (You can also right-click on the dialog icon and select <b>Delete Repetition</b> from the shortcut menu.)
Insert a new dialog between existing dialogs	Select a dialog in the interview outline and click the  <b>Insert Repetition</b> button. A new, empty dialog is added before the current dialog. (You can also right-click on the dialog and select <b>Insert Repetition</b> from the shortcut menu.)
Move an existing dialog up or down in the interview outline	Select the dialog you want to move in the interview outline and click the  <b>Move Repetition Up</b> button or the  <b>Move Repetition Down</b> button. The dialog will be moved to the new location. (You can also drag a dialog icon in the outline and drop it in a new location.)

## To enter answers in a 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.
4. Complete any of the following options:

To	Do This
View the answers in their own dialog, rather than as part of the spreadsheet	Place your cursor in a row and click  <b>Edit Row</b> . This displays the variables for easier viewing.
Delete a row of answers	Place your cursor in the row you want to delete and click  <b>Delete Row</b> . The set of answers is permanently removed from the interview.
Insert a new set of answers	Place your cursor in the spreadsheet and click  <b>Insert Row</b> . A new, empty row is added to the spreadsheet above the current row.
Move a row in the spreadsheet to a new location	Click in the row you want to move and click either the  <b>Move Repetition Up</b> button or the  <b>Move Repetition Down</b> button. (You can also click the leftmost column number and drag it up or down in the spreadsheet to move that row.)
Sort the contents of a spreadsheet in alphanumeric order	<p>Click in the column on which you want to sort and click the  <b>Sort</b> button. The <b>Sort Spreadsheet</b> dialog box appears. Designate your sorting preferences. (<b>Ascending</b> sorts the specified column from 1 to 9, and A to Z; while <b>Descending</b> sorts from 9 to 1, and Z to A) and click <b>OK</b>.</p> <p>To sort on a second level, enter your preferences in the <b>and then</b> fields.</p> <p>If the template developer has specified any sorting options, answers will be merged in the document using those sort option. Otherwise the sort order you specify here will be used in the assembled document.</p>

**Notes:**

- 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 specify an option that causes HotDocs to update the interview less frequently. (See [Update Your Interview Outline.](#))

# 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.

**Note:** To see the child dialog in the interview outline, click the plus sign next to the parent dialog. The inserted dialog appears in the interview outline. You can also select **Expand All (View menu)**.

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.

### Notes:

- You can hide all inserted dialogs 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 specify an option that causes HotDocs to update the interview less frequently. (See [Update Your Interview Outline](#).)

## 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  **Edit Row**. A pop-up interview appears.
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 [Overview: 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 choose where the assembled clause text is placed. 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 the clauses with specific text in their title or description. In the text box, type the text you are searching for.
  - 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.
4. From the **Available** list, select each clause you want to use and click the  **Select** button to add it to the **Selected** list. If you need to remove a clause, select it, then click the  **Remove** button. (Press **Ctrl** or **Shift** to select multiple items.)
5. Optionally, to reorder the clauses in the **Selected** list, select and drag each 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**  in the navigation bar.
7. Answer any questions contained in the clause(s) you selected.

Any clauses you selected will be merged into the assembled document for the template.

# View an Assembled Document in Markup View

**Warning:** Markup View is supported only when assembling Microsoft Word documents.

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. (See [Specify How Documents Should Be Marked Up](#).)
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. Choose **Markup View (View menu)**. The document changes to show the marked up document.

### Notes:

- 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.
- If the template developer has used merge fields in question prompts, those merge fields can be displayed as markup fields. (Choose **Markup View (View menu)** while at 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)**.

# 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.

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 and Four-Digit Years](#).

- 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**.

## Notes:

- 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 specify an option that causes HotDocs to update the interview less frequently. (See [Update Your Interview Outline](#).)

# 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, where you can perform any of the following tasks (see [Customize the End of Interview Dialog](#) for information on choosing which options appear at this dialog):

To	Do This
Return to the first unanswered question in the interview	Click  <b>Go to the first unanswered question in the interview.</b>
Send the assembled text document to the word processor so you can view it in its final format	Click  <b>Send the assembled document to the word processor.</b>  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 <a href="#">Change Your Default Word Processor</a> .
View the assembled form document	Click the  <b>View the assembled form document at the Form Document tab</b> button. (See <a href="#">View the Assembled Form Document</a> .)
Insert the assembled document into an open word processor document	Click  <b>Paste the assembled document into the open word processor document.</b>  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 to the Clipboard (only available when assembling Microsoft Word documents)	Click  <b>Copy the assembled document to the Clipboard.</b>
Save a copy of the assembled document	Click  <b>Save the assembled document in a file.</b>

<p>Save an assembled text or form document as a PDF file</p> <p><b>Warning:</b> To save assembled documents as PDF files, you must have HotDocs PDF Advantage installed. For details on obtaining a license, contact your HotDocs sales representative.</p>	<p>Click  <b>Save the assembled document as a PDF.</b></p> <p><b>Warning:</b> For complete instructions on saving assembled documents as PDF files, see <a href="#">Save a Document as a PDF File</a>.</p>
<p>Close the assembly window without saving a copy of the assembled document</p>	<p>Click  <b>Close this window without saving the assembled document.</b></p>
<p>Choose which buttons are displayed in the <i>End of Interview</i> dialog</p>	<p>Click  <b>Choose which buttons are displayed on this dialog.</b></p>

**Notes:**

- 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](#).

## Enter Answers in Non-English Languages

During a HotDocs interview, HotDocs lets you enter answers in left-to-right reading languages other than English.

Answers entered in these languages will be merged correctly anywhere the variable is used—either in the assembled document or in other questions or text used in the interview. Answers can be saved in either a HotDocs answer file or an XML answer file.

# 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 assembly. (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** box, 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 information in the following table:

To	Do This
Ignore the current instance of the word and continue spell checking	Click <b>Ignore</b> .
Ignore all instances of the word and continue spell checking	Click <b>Ignore All</b> .
Correct only the current instance of the word and continue spell checking	Select an existing replacement from the <b>Change to</b> list (or type the replacement in the <b>Change to</b> box) and click <b>Change</b> .
Correct all instances of the misspelled word and continue spell checking	Select an existing replacement from the <b>Change to</b> list (or type the replacement in the <b>Change to</b> box) and click <b>Change All</b> .
Add the word to your personal dictionary so that the spelling checker will not question the word again	Click <b>Add</b> .
Display additional spelling alternatives for the unrecognized word	Click <b>Suggest</b> .  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 <b>Options</b> . (You can also change your options at the <b>HotDocs Options</b> dialog box. See <a href="#">Change Your Spell Checking Options</a> .)

# Assemble Documents Linked to a Database

---

## Overview: 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 one record from a database table

1. Assemble a document. (See [Assemble a Text or Form Document](#).)
2. At the assembly window, when a database table appears, click a row in the table.
3. 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.)
4. Click **Next**. HotDocs merges the answers from the database record into the assembled document.

## To select two or more records from a database table

1. At the assembly window, when a database table appears, click the first record you want to select and then perform one of the following options:
  - To select multiple, adjacent records, press and hold the **Shift** key while clicking on the last record you want included.
  - To select multiple, nonadjacent records, press and hold the **Ctrl** key while clicking on each subsequent record you want included.
2. 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.)
3. Optionally, select a record and click the **Up** (moves the record up to the next row in the list of records) button or the **Down** (moves the record down to the next row in the list of records) button. 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.)
4. Click **Next**. HotDocs merges the answers from the database into the assembled document.

### Notes:

- 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 **Clear**.
- 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** (moves the record up to the next row in the list of records) button or the **Down** (moves the record down to the next row in the list of records) button. You can also click the column headings in the **Selected Items** list to automatically sort the selected items by any field.

**Note:** 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.

**Note:** 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.
6. Once the list is filtered, you can select a record or records (see [Select Records During Document Assembly](#)) and continue with the interview.

### Notes:

- 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](#).)

**Warning:** 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.

# Working with Assembled Documents

## Print an Assembled Text or Form Document

Many projects require you to print copies of assembled documents.

**Warning:** Certain printers have difficulty printing form documents. See [Print a Form Document](#) for details.

### To print an assembled text document

1. At the HotDocs library window, select a template and begin assembly. (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.

### To print an assembled form document

1. At the HotDocs library window, select a template and begin assembly. (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. (You can also press **Ctrl+P**.)
4. Make your selection, depending on whether you are printing a PDF form document or a HotDocs form document.
  - For documents assembled from PDF-based templates (.HPTs), the **Print Document** dialog box appears, where you can choose to have Adobe Acrobat handle the printing.
  - If you're printing a form document, HotDocs Filler allows you to specify other options:

To	Do This
Print the form and answers	Select <b>Form with answers</b> from the <b>Output</b> group.
Print only the static text of the form template	Select <b>Form only (blank form)</b> from the Output group.
Print the answers onto a pre-printed copy of the form	Select <b>Answers only (use preprinted form)</b> from the Output group.
Change the double-sided printing settings	Select the desired option in the <b>Print this form</b> option group.

**Notes:**

- Some forms are different sizes and require settings different from the default printer settings. Using command-line options, you can make sure that the necessary paper size and paper source will be used. (See [Overview: Command-Line Options](#), [Paper Size](#), and [Paper Tray](#).)
- When printing with Adobe Acrobat, the form and the answers are always printed. Acrobat uses the current printer to determine the position for line breaks, answer overflows, and the addendum layout. All answer overflows are automatically sent to the addendum.

# 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.

## To save an assembled text or form document

1. At the HotDocs library window, select a template and begin assembly. (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** box, enter a name for the document and click **Save**.

**Warning:** 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.

# View an Assembled Document in Markup View

**Warning:** Markup View is supported only when assembling Microsoft Word documents.

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. (See [Specify How Documents Should Be Marked Up](#).)
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. Choose **Markup View (View menu)**. The document changes to show the marked up document.

### Notes:

- 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.
- If the template developer has used merge fields in question prompts, those merge fields can be displayed as markup fields. (Choose **Markup View (View menu)** while at 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)**.

# Save a Document as a PDF File

With PDF Advantage, you can use the PDF printer driver to save assembled text and form documents as PDF documents. Likewise, you can use the printer driver to save other documents not associated with HotDocs as PDF documents. Creating PDF files allows you to take advantage of features available in Adobe Acrobat.

## To save assembled text documents as PDF files

1. At the HotDocs library window, select a text template and click  **Assemble**. (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.

**Note:** Information you type in these boxes 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.

## To save an assembled PDF-based form document as a PDF file

1. At the HotDocs library window, select a PDF template file (.HPT) and begin assembly. (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.

**Note:** Information you type in these boxes 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.

### To save other documents as PDF files

1. Open the document in a Windows program with printing capability.
2. Select the **Print** command for that program.
3. At the printing dialog box, select **HotDocs PDF Driver** as the printer.
4. Optionally, click **Properties** and assign any properties to the document you may need.
5. Click the **Print** or **OK** button. The **Save As** dialog box appears.
6. Select a location and type a name for the new file.
7. At the **Save as type** box, select **PDF File (\*.pdf)**.
8. Optionally, select **Document Settings** and click the **Edit** button to add information about the document's title, subject, author, or search keywords.
9. Optionally, select **View Resulting PDF** to open the PDF once it has been created.
10. Click the **Save** button. The PDF file is created and saved to the location you specified.

**Warning:** If the file you're using to create a PDF document is a form document, the answer fields, components, 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.

**Note:** If you need to protect the PDF document, you can assign different security options to it when you save it. See [Assign Security Options to a PDF Document](#) for details.

# 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.

You can assign security at one of two places:

- At the **Security** tab of the **HotDocs PDF Driver Properties** dialog box.
- At the HotDocs **Save As** dialog box (when saving an assembled document as a PDF).

## To assign security options at the HotDocs PDF Driver Properties dialog box

1. Open the document in a Windows program with printing capability.
2. Select the  **Print** command for that program.
3. At the printing dialog box, select **HotDocs PDF Driver** as the printer.
4. Click **Properties**. The **HotDocs PDF Driver Properties** dialog box appears.
5. Click the **PDF Settings** tab. The window changes to show the different options you can assign to a PDF.
6. Click the **Security** drop-down button and assign any of the options available. (Click **Advanced** to review additional options.)

## 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 information in the following table (for additional information about each of these options, see the Adobe Help file):

To	Do This
Leave the document unprotected	Select <b>None</b> . No security options will be applied and the user can edit the document (using Adobe Acrobat Professional).
Apply less-stringent security options to the document (and make the document compatible with all versions of Adobe starting Adobe 3.x)	Select <b>40-bit RC4 encryption</b> . Selecting this option affects which options are available in the rest of the dialog box.
Apply more stringent security options to the document (and restrict use of the document to Acrobat 5.x and later users)	Select <b>128-bit RC4 encryption</b> . Selecting this option affects which options are available in the rest of the dialog box.

Require the user to enter a password when changing security options for the document	Select <b>Master password</b> and enter (and confirm) the required password.
Require the user to enter a password when attempting to open the document	Select <b>User password</b> and enter (and confirm) the required password.
Allow varying levels of printing	Select one of the printing options: <b>Fully allowed</b> , <b>Low resolution only</b> , and <b>Not allowed</b> .
Keep users from copying the text in the document and pasting it into other applications	Clear <b>Allow content copying and extraction</b> .
Allow the document to be read out loud by Acrobat	Select <b>Allow access for visually impaired</b> .
Let users make changes to the document	Select <b>Allow content editing</b> . When selected, users can modify the pages of the document, including adding or removing pages, or rotating pages.
Let users make changes to page properties of the document (such as inserting pages)	Select <b>Allow page-level editing (insert, delete, rotate)</b> .
Let users make comments or other notes in the document	Select <b>Allow annotations</b> .
Let users enter information in Adobe fields on the form	Select <b>Allow form filling</b> .

# 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.

**Warning:** 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 assembly. (See [Assemble a Text or Form Document](#).)
2. 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.

# 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.

## Warnings:

- 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.](#))
2. 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 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, perform any of the following actions:

To	Do This
Save the assembled text document to disk	Click the  <b>Save Document</b> button. (See <a href="#">Save an Assembled Text or Form Document.</a> )
Send the assembled text document to the word processor so you can see it in its final format.	Click the  <b>Send Document</b> button. (See <a href="#">Send the Assembled Text Document to the Word Processor.</a> )
Print the assembled text document	Click the  <b>Print Document</b> button. (See <a href="#">Print an Assembled Text or Form Document.</a> )
Attach the assembled text document to an e-	Click <b>Send Document To &gt; Mail Recipient</b> ( <b>File</b> menu). (See <a href="#">Attach a Text or Form</a>

mail message	<u>Document to an E-mail Message.</u> )
Close the assembly window	Click <b>Close</b> ( <b>File</b> menu).

**Note:** 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

**Warning:** 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.

**Warning:** 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 assembly. (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 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. Complete any of the following optional tasks:

To	Do This
<p>Jump to the place in the interview where the question is asked</p> <p><b>Warning:</b> Because of scripting in the template, sometimes you will not be able to move between the <b>Interview</b> and <b>Document</b> tabs. HotDocs will warn you when this is the case.</p>	<p>Place your cursor in the answer field and click the  <b>Go to Answer in Interview</b> button.</p> <p>While viewing the <b>Interview</b> 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  <b>Go to Answer in Document</b> button.</p>
<p>Move between answers in the document</p>	<p>Click  <b>Next</b> or  <b>Previous</b> to move to the next or previous answer in the document.</p> <p>Click the  <b>Next Unanswered</b> or</p>

	<p> <b>Previous Unanswered</b> button to move to the next unanswered or previous unanswered question in the document.</p> <p>Press <b>Ctrl+M</b> or <b>Ctrl+Shift+M</b> to move between matching answers in the document (for example, those answers in the document that are exact matches). (You can also press <b>Shift</b> as you click the  <b>Next</b> or  <b>Previous</b> button.)</p> <p>Click  <b>End</b> or  <b>Beginning</b> to go to end of the document or the beginning of the document.</p>
--	---

**Notes:**

- 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 **Instant 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  **Instant 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.

# Edit Document Text at the Document Preview Tab

**Warning:** 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.

**Warning:** 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 assembly. (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. 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.

## Notes:

- To move between sections of editable text, click the  **Next Editable Text** or the  **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.

# 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 assembly. (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.

### Notes:

- 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.)

# 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 (.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 Form Document](#) for instructions on completing these actions.

## To open a saved text document using Windows Explorer

1. Start Windows Explorer.
2. Locate the document file and double-click it. Your word processor appears, showing the saved document.

## To open a saved text document using a word processor

1. Start your word processor.
2. Using that program's commands, locate and open the desired document file.

**Warning:** 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.

# Comparing Assembled MS Word Documents

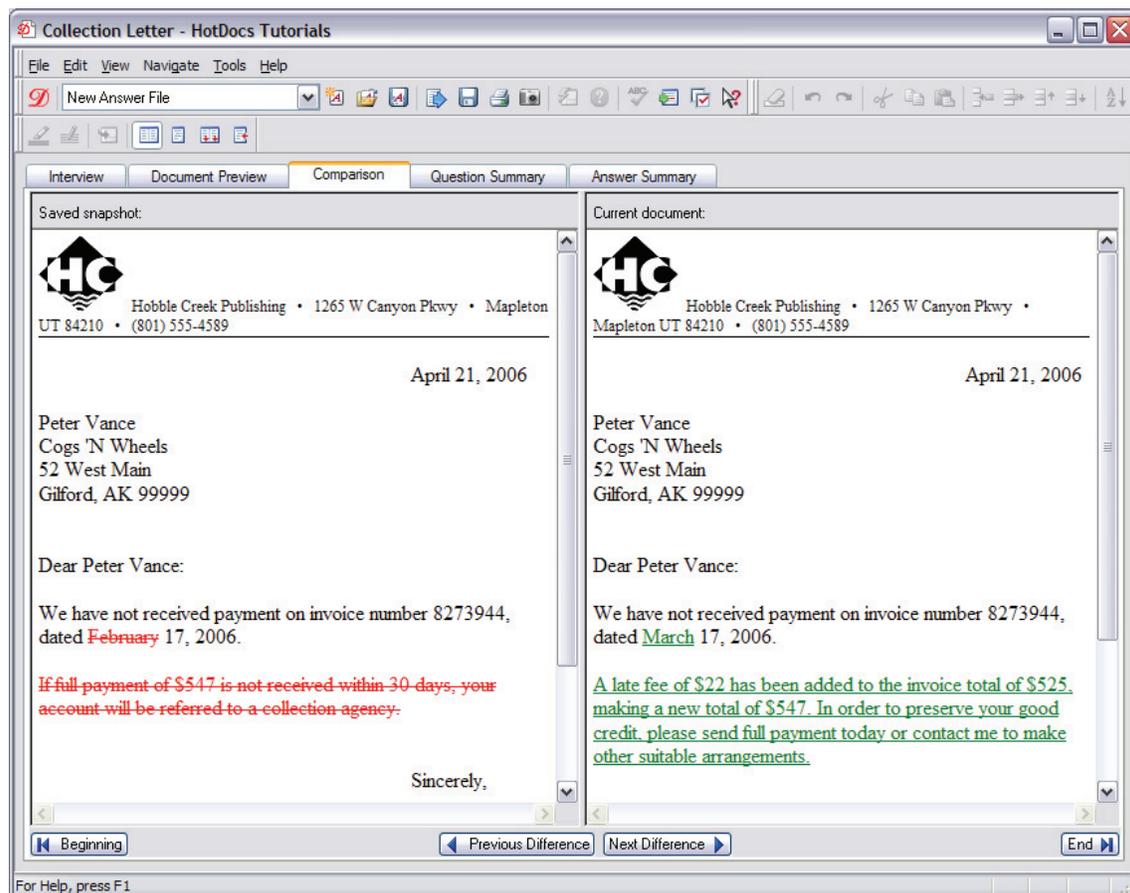
## Overview: HotDocs® Compare

**Note:** HotDocs Compare is only available for Word users.

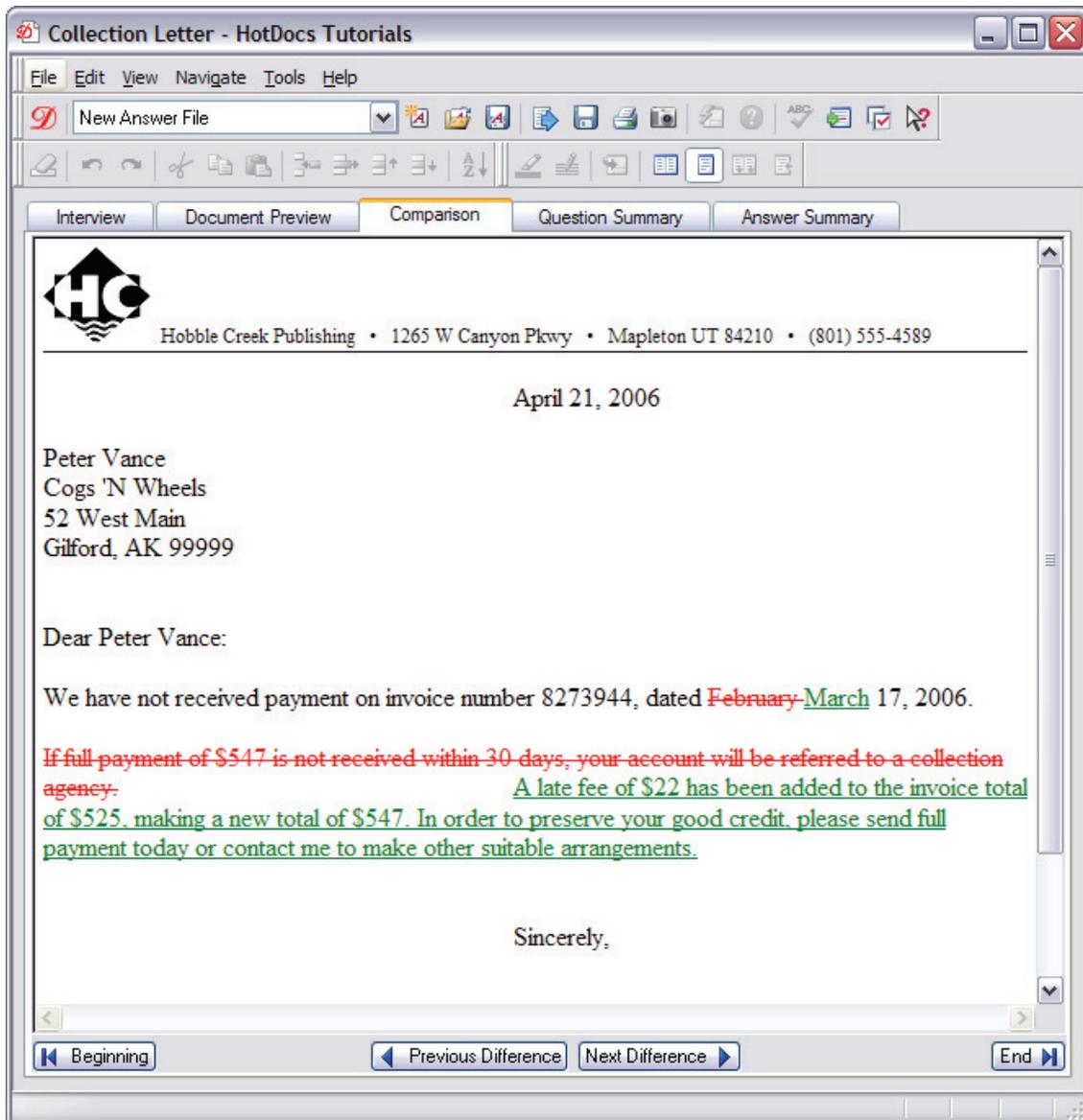
At times, you may want to compare different versions of a document you are assembling. For example, perhaps you want to see how answers you enter during the interview affect how the document is assembled. You can answer the questions in the interview one way, save a snapshot of the document, change your answers, and then compare the snapshot and the document.

As you are viewing the compared documents, you can view the comparison in either one pane or two.

In this first example, the comparison is shown in two different panes. The left pane, or snapshot, shows the document at the time the snapshot was saved. The right pane shows the document with the current answers. Differences in the two documents are marked using strike-throughs and underlines:



In this example, the comparison is displayed in a single pane:



As you are viewing the comparison, you can use the **Comparison Navigation Bar** to move between the sections of document that have changed. When viewing a side-by-side comparison, you can have HotDocs keep the two panes synchronized.

Finally, you can save different snapshots of the document. This lets you track the progression of an assembled document and allows you to retrieve older comparisons you have created.

**Note:** Information in this topic applies to **HotDocs Compare** users only. For details on purchasing a license for HotDocs Compare, contact your HotDocs sales representative at (800) 500-3627.

# Compare Different Versions of a Document

**Note:** HotDocs Compare is only available for Word users.

When you want to compare different versions of a document, you must first save a snapshot of the document you want to compare. You can then answer additional questions in the interview or change existing answers and then view the **Comparison** tab where the current (or the most up-to-date) document is shown along with the snapshot. Changed text in the snapshot is crossed out, while new text in the current document is underlined.

You can save a snapshot of a document anytime you want HotDocs to store a record of the document for comparison purposes. For example, if you are assembling a document where a single answer can change the outcome of the entire assembled document, you can answer the question one way and save a snapshot of it. Then you can return to the interview, answer the question another way, and then view how the two documents compare.

Additionally, when comparing documents, you may want to save multiple snapshots so that you can compare each one against the current document. For example, perhaps you want to see how a series of changes (not just a single change) affects the outcome of the document. Each time you click the  **Save Document Snapshot** button, you are prompted to save the document to disk. These saved documents are then listed in the **Saved document** drop-down list. At any time during the comparison process, you can click this drop-down list and choose a snapshot so you can compare it to the current document.

## To save a snapshot of the document for comparison purposes

1. At the HotDocs library window, select a template and begin assembling the document. (See [Assemble a Text or Form Document](#).)
2. Answer questions in the interview.
3. At the point in the interview where you want to save a snapshot of the document, either click the  **Save Document Snapshot** button, or click the **Comparison** tab. (HotDocs will only create a snapshot the first time you click the **Comparison** tab and you haven't yet saved a snapshot. In all other instances, you must click the  **Save Document Snapshot** button.)

At this point, a copy of the document with those answers is created and displayed in the **Comparison** tab. (If viewing the comparison in two panes, the snapshot appears in the **Saved snapshot** pane.)

4. Click the **Interview** tab and make whatever changes you need to answers in the dialogs.
5. Click the **Comparison** tab. Text in the snapshot that has changed is crossed out, while the new text is added to the current document and underlined.
6. If you want to save multiple snapshots, make sure **Allow saving and comparing of multiple snapshots** is selected at **HotDocs Options** (see [Set Preferences for Saving Comparison Snapshots](#)). Each time you save a snapshot, it is added to the **Saved snapshot** drop-down list so you can select it and display it as you need. These files are actually saved to disk.
7. Optionally, complete any of the following tasks:

To	Do This
View the comparison in a single pane	Click the  <b>Merged Comparison</b> button. To view the comparison in two panes again, click the  <b>Side-by-side Comparison</b>

	button.
Synchronize the views once you have scrolled or changed the view in one of the panes	<p>Click the  <b>Synchronize Views</b> button. The other pane is scrolled to the same position in the document as the pane you are currently viewing.</p> <p>To keep the panes synchronized at all times, click the  <b>Keep Views Synchronized</b> button. When you scroll in one pane, the other pane is likewise scrolled.</p>

**Notes:**

- If you've edited the document text (see [Edit Document Text at the Document Preview Tab](#)), you can choose whether to save the original document text in the snapshot, or to save the edited text in the snapshot. See [Save Edited Document Text in Comparison Snapshots](#).)
- Information in this topic applies to **HotDocs Compare** users only. For details on purchasing a license for HotDocs Compare, contact your HotDocs sales representative at (800) 500-3627.

# Navigate Through Changes in Compared Documents

**Note:** HotDocs Compare is only available for Word users.

As you view compared documents at the **Comparison** tab of the assembly window, you can move between changes in the document. You do this by using the Comparison Navigation Bar. As you navigate through the comparison, HotDocs highlights the differences between the document snapshot and the current document.

## To view specific changes in a comparison

1. While viewing the **Comparison** tab, click the  **Next Difference** button or the  **Previous Difference** button. HotDocs moves you through the document, showing you changes that have been made.
2. Optionally, click the  **Beginning** or  **End** buttons to go to the beginning or end of the document.

### Notes:

- You can change the colors HotDocs uses when marking changes in the document. See [Specify Colors for Marking Changes in Comparisons](#) for details.
- Information in this topic applies to **HotDocs Compare** users only. For details on purchasing a license for HotDocs Compare, contact your HotDocs sales representative at (800) 500-3627.

# Working with an Assembled Form Document

---

## Overview: 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.

# 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:

To	Do This
Create a new field on the document	Click the  <b>Select Tool</b> button and draw the field. (See <a href="#">Create a Form Field</a> .)  To enter text in the field click on the  <b>Fill Tool</b> and then click in the field.
Add additional answers to a table	If the template developer has included it, click the  <b>Table Wizard</b> 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  <b>Answer Wizard</b> button next to the field.
Show or hide the colored fields	Click the  <b>Show Fields</b> button.
Quickly browse through the pages of a form document	Click the  <b>Show Thumbnails</b> 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.

<p>Jump to a specific page of the assembled document</p>	<p>Choose <b>Go To</b> (<b>Edit</b> menu) and enter the page number of the page you want to view.</p>
<p>Move through the answers in a document</p>	<p>Place your cursor in a field and click one of the following navigation buttons: (If the <b>Form Navigation Bar</b> isn't showing, choose it at the <b>View</b> menu.)</p> <ul style="list-style-type: none"> <li>■ Clicking  <b>Next</b> or  <b>Previous</b> moves you to the next or previous answer in the document.</li> <li>■ Clicking the  <b>Next Unanswered</b> or  <b>Previous Unanswered</b> button moves you to the next unanswered question in the document</li> <li>■ Pressing <b>Ctrl+M</b> or <b>Shift+Ctrl+M</b> moves between matching answers in the document (for example, those answers in the document that are exact matches). (You can also press <b>Shift</b> as you click the  <b>Next</b> or  <b>Previous</b> button.)</li> <li>■ Clicking  <b>End</b> or  <b>Beginning</b> to go to the end of the document or the beginning of the document.</li> </ul>

**Notes:**

- 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 Options** dialog box. See [Check Fields for Answer Overflow](#). Similarly, you can specify when this warning should appear. See [Set Rules for Handling Answer Overflow in Form Fields](#).

# 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 form template and begin assembly. (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 [Type Answers Directly in a Form Document](#).)
4. Once you have finished entering answers, you can perform any of the following tasks using toolbar buttons and menus:

To	Do This
Save the assembled form document to disk	Click the  <b>Save Document</b> button. (See <a href="#">Save an Assembled Text or Form Document</a> .)
Print the assembled form document	Click the  <b>Print Document</b> button. (See <a href="#">Print an Assembled Text or Form Document</a> .)
Attach the assembled form document to an e-mail message	Click <b>Send Document To &gt; Mail Recipient (File menu)</b> . (See <a href="#">Attach a Text or Form Document to an E-mail Message</a> .)
Close the assembly window	Click <b>Close (File menu)</b> . HotDocs prompts you to save any changed answers.

**Note:** See [Overview: Assemble a Form Document](#) for a list of changes you can make to an assembled form document.

# Check 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.
4. 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:

To	Do This
Reduce the answer's font size to the minimum size allowed for the field	Select <b>Shrink the answer to fit in the field</b> . 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 <b>Send the answer to the addendum</b> . 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 <b>Split answer</b> if you want only the part of the answer that doesn't fit in the field to be sent to the addendum.
Change the text of the answer, such as reduce the number of words in the answer	Click <b>Edit</b> . HotDocs highlights the field that contains the answer so you can edit the text.  Once you edit your answer, click back on the <b>Overflow Status</b> dialog box to have HotDocs recheck the overflow.

<p>Make the answer field larger</p>	<p>Click <b>Resize</b>. HotDocs selects the field so you can click one of its handles to resize it.</p> <p>Once you resize the answer field, click back on the <b>Overflow Status</b> dialog box to have HotDocs recheck the overflow.</p>
<p>Override any settings or formats of the field and enter any text in the field</p>	<p>Click <b>Override</b> and enter the text you choose in the field.</p> <p>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.</p> <p>Once you override the answer field, click back on the <b>Overflow Status</b> dialog box to have HotDocs recheck the overflow.</p>
<p>Resolve the overflow at a later time</p>	<p>Click <b>Ignore</b>. The <b>Overflow Status</b> 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.</p>
<p>Resolve the overflow on your own</p>	<p>Choose <b>Close</b>. HotDocs highlights the overflowing field so you can change the answer in whatever way you choose.</p>

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.

**Note:** You can prevent HotDocs from checking for field overflow during assembly. See [Set Rules for Handling Answer Overflow in Form Fields](#) for details.

# 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 Fields for Answer Overflow](#) for details.

# Print a 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 HotDocs form document (.HFD)

1. At the HotDocs library window, select a form template and begin assembly. (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. The **Print** dialog box appears.
4. Make your printing selections, based on the information in the following table:

To	Do This
Print the form and answers	In the <b>Output</b> group, select <b>Form with Answers</b> .
Print only the static text of the form template	In the <b>Output</b> group, select <b>Form Only (Blank Form)</b> .
Print the answers onto a pre-printed copy of the form	In the <b>Output</b> group, select <b>Answers Only (Use Preprinted Form)</b> .
Change the double-sided printing options	Select the desired option in the <b>Print this form</b> group: <ul style="list-style-type: none"><li>■ <b>Single-Sided</b> prints the document using one side of each sheet of paper (default setting).</li><li>■ <b>Double-Sided, Side-to-Side</b> 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.)</li><li>■ <b>Double-Sided, Top-to-Bottom</b> 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.)</li></ul>

## To print a PDF form document (.HPD)

1. At the HotDocs library window, select a form template and begin assembly. (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. The **Print Document** dialog box appears, asking whether you want to print using the functionality of HotDocs or Adobe Acrobat.
4. Make your selection based on the information in the following:
  - 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.

**Notes:**

- Certain printers have difficulty printing form documents. See [Troubleshoot Form Printer Problems](#) for details.
- Some forms are different sizes and require settings different from the default printer settings. Using command-line options, you can make sure that the necessary paper size and paper source will be used. (See [Overview: Command-Line Options](#), [Paper Size](#), and [Paper Tray](#).)

# 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.

### Notes:

- You can specify the print setting for all forms you create at the **HotDocs Options** dialog box. (See [Overview: Customize HotDocs](#).)
- You can use command-line options to control the paper size and paper source. (See [Paper Size](#) and [Paper Tray](#).)

# Troubleshooting Form Printer Problems

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.

# Edit a Saved Form Document

After assembling a form document, you often need to save the document to disk. Once saved, you can re-open 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.
2. Optionally, you can perform any of the following tasks:

To	Do This
Change an answer in a field	Click on a field. The field becomes active, allowing you to enter an answer.  <b>Warning:</b> 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  <b>Select Tool</b> button selected, press the left mouse button and move the mouse pointer to draw a rectangle. To type in this field, select the  <b>Fill Tool</b> button.  The fields you create in Filler have no variable associated with them. They are simply text boxes where you can type information.
Print the document	Click the  <b>Print</b> button.
Open a different HotDocs form file	Click the  <b>Open</b> button, and then locate the desired file. (You can also press <b>Ctrl+O</b> .)

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.
- Change the size of a field.
- Change the position or margins of a field.
- Create a new answer field.

- Rotate text in a field.
- Convert answers on the form to bar code format.
- Insert a graphic into a field.
- Cross out static text on the form.
- Circle static text on a form.

# Create and Edit 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.

### 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 a Field](#).)
  - 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](#).)

#### Notes:

- 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:
  - Press the **Ctrl** key while clicking each field.
  - 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.

HotDocs Automator provides several other methods for selecting fields or canceling the selection, as described in the following table:

To	Do This
Select all the fields on the current page	Click <b>Select All</b> ( <b>Edit</b> menu). (You can also press <b>Ctrl+A</b> .)
Cancel the selection of all selected fields	Click outside the fields, or press the <b>Esc</b> key.
Cancel the selection of only one of a group of selected fields	Hold down the <b>Ctrl</b> key and click that field.
Add fields to your group of selected fields without canceling the selection of those already selected	Hold down the <b>Ctrl</b> key and click the new fields.

### Notes:

- Click the  **Show Fields** button to show and hide field colors.
- 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 Fields](#).)

# 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.

## To create a check-box field

1. Edit a form template. (See [Edit a Form Template](#).)
2. Double-click in the field margins, 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**.

**Note:** When assigning a True/False variable, if you want to merge a character other than an **X**, indicate the character in the **Format** box. To change the font for the field, see [Adjust Appearance of Check-Box Fields](#). When assigning a Multiple Choice variable, you should group the fields before attaching the variable. See [Group 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](#).)

### Notes:

- 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](#).)
- You may need to change other properties of the field, as well, such as horizontal and vertical alignment, borders and margins, and so forth.
- 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**.

# 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  **Detect** button. This button 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.

# 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 template, 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 **Order/Size** 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** boxes. (You can also click the up or down arrows for each field to change the distance.)

### Notes:

- If a field you want to move is part of a group, you must first ungroup the fields. (See [Ungroup 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.)
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** 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.

# 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.)

# Copy One or More 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 [Copy Components from One File to Another](#) 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](#).)
3. 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.)

### Notes:

- 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](#).)

# Resize a Field

*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 template. 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 the 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 **Order/Size** 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.

# Group 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 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.

## 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](#).)

### Notes:

- When fields are grouped, you cannot change properties for individual fields. You must first ungroup the fields. (See [Ungroup 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](#).)

# Ungroup 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.

**Warning:** 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.

## 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 following information:

Type of Field	Default Color	Description
Edit field	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 field	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 <b>X</b> or other character. (See <a href="#">Create a Check-Box Field</a> .)
Resource field	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 <a href="#">Create a Resource Link on a Form</a> .)  Resource fields provide help for the form in general, instead of for a specific variable or dialog.
Control field	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.

**Note:** You can change the default colors used for form fields at the **HotDocs Options** dialog box. (See [Change Colors in HotDocs Automator.](#))

# 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 your users' computers.

## 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.

**Note:** You can specify default font properties for all new fields you create at HotDocs Options. See [Adjust Appearance of Edit Fields](#) for more details.

# Format Lines and Paragraphs of Text in a 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:

To	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 <b>First line indent</b> 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 <b>Lines per inch</b> box.
Indicate how many lines can appear in the field	Enter the number of lines in the <b>Maximum lines</b> 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 <b>Max chars/line</b> box. When the number of characters exceeds this limit, text will wrap to the next line.

# Rotate Answers in a Field

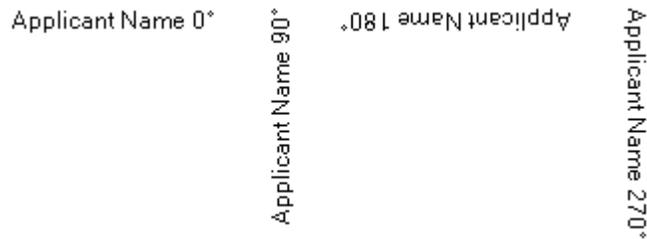
*These instructions can also be used to rotate answers in an answer field both at the **Form Document** tab of the assembly window and in HotDocs Filler.*

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:

Applicant Name 0°      Applicant Name 90°      Applicant Name 180°      Applicant Name 270°



### Notes:

- 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 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.

# Keep Contents of Fields from Printing

*These instructions can also be used to keep field text from printing 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 during direct-fill assembly.
3. Click the  **Field Properties** button. The **Field Properties** dialog box appears.
4. At the **Type** tab, select **Non-printing field**.

# 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 template 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](#).)

**Note:** You can use the alignment controls on the **Field Properties** dialog box to position the strike-through text more accurately.

# 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](#).)
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.

**Note:** To preview the circle (see [Preview the Formatting of Answers in a 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](#).)
  - 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.
  - 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). (Tip: With a True/False variable, you can display one graphic if the answer is *true* and another graphic if it's *false*. To do this, enter two file names separated by a forward slash (/) in the **Format** box.)
3. Click **OK** at the variable editor and at the **Variable Field** dialog box.
4. With the field still selected, click the  **Field Properties** button. The **Field Properties** dialog box appears.
5. At the **Type** tab, select **Image** from the **Display type** group.
6. Optionally, change the **Field type** to **Control** to restrict the user's ability to modify the field. (See [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. (See [Change the Field Type](#).)
8. Optionally, click the  **Variables** button in the Automator toolbar to view the image.

The image is visible when a user assembles a document and views the **Form Document** tab.

# 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 a Field](#) or [Change the Borders and Margins of a Field](#).)

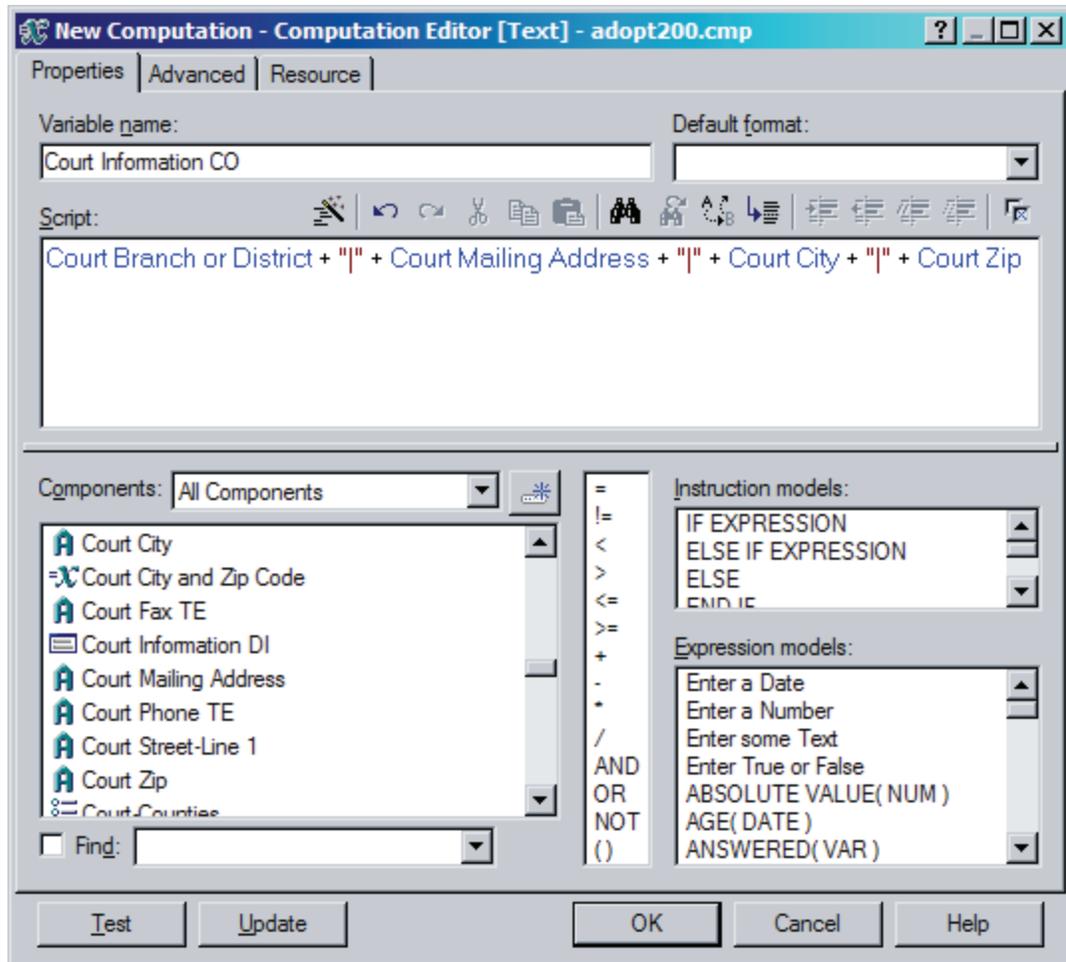
There are two methods for creating a bar code:

- **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](#).)
3. 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 [Overview: Instruction and Expression Models](#).)

For example, using a distinct delimiter character in the script makes it easier to distinguish separate answers when the bar code is scanned into a database.



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 (|).)

**Notes:**

- 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](#).)
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** 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 steps 4 and 5 for each field in the group, using the same group name as you assigned to the first field.

Now, when the user answers the question, text that does not fit in the first field will overflow to the next field named in the group.

**Note:** To span text across multiple fields on the same page, see [Group Fields So Answers Can Flow From One Field to Another](#).

# Customize Form Addenda

## Define Overflow Properties for a 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. (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. For example, in an addendum, grouped fields and tables are handled a little differently from regular fields. Grouped fields are shown in the addendum as just one answer. Answers from tables, by default, are also sent to the addendum as one item. That is, if one answer overflows, then all of the answers in the table are sent to the addendum. 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 the 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:

To	Do This
Automatically shrink an answer that overflows to a specific point size	Select <b>Shrink answer as needed</b> and enter a point size in the <b>points</b> box.
Automatically send an answer that overflows to the addendum	Select <b>Send answer to addendum</b> . To send only part of a multi-line answer to the addendum, select <b>Split answer</b> . (See <a href="#">Split a Multi-Line Answer Between the Form and the Addendum.</a> )
Define the text that appears in the answer field when an answer is sent to the	Enter the text in the <b>Cross-reference text</b> box.

addendum	
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the <b>Addendum label text</b> box.
Make an answer the first item on its addendum page	Select <b>Begin addendum entry on a new page</b> .
Make an answer the last item on its addendum page	Select <b>Begin new page following addendum entry</b> .

### To define the overflow properties for run-on fields

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:

To	Do This
Automatically shrink an answer that overflows to a specific point size	Select <b>Shrink answer as needed</b> and enter a point size in the <b>points</b> box.
Automatically send an answer that overflows to the addendum	Select <b>Send answer to addendum</b> . To send only part of a multi-line answer to the addendum, select <b>Split answer</b> .
Use a check box in the grouped fields to indicate the answer overflows  Sometimes forms include a check box that indicates when an answer is continued elsewhere, such as in the addendum or in an attachment. Choosing one of these options selects that check box when the answer overflows.	Select either <b>Indicator check box is first in field group</b> or <b>Indicator check box is last field in group</b> , depending on where the check box that indicates overflow is positioned.  Select <b>No addendum entry indicator check box</b> if there is no check box to indicate an answer has been sent to an attachment or addendum.
Define the text that appears in the answer field that an answer is sent to the addendum  If you want the indicator check box to be the only designation that a field overflows, enter	Enter the text in the <b>Cross-reference text</b> box.

NONE in this box.	
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the <b>Addendum label text</b> box.
Make an answer the first item on its addendum page	Select <b>Begin addendum entry on a new page</b> .
Make an answer the last item on its addendum page	Select <b>Begin new page following addendum entry</b> .

### To define the overflow properties for tables

1. Edit a form template. (See [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:

To	Do This
Automatically send an answer (including extra rows in the table) that overflow to the addendum	Select <b>Send answers to addendum</b> .
Define how much of the table is sent to the addendum when there is overflow	<p>In the <b>When sending answers to addendum</b> group, select one of the following options:</p> <ul style="list-style-type: none"> <li>■ Select <b>Send entire table</b> 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, <i>See 1 in Addendum</i>) is merged in the top row of the table and no other rows contain any answers.</li> <li>■ Select <b>Send complete rows</b> to send only rows that overflow to the addendum. For example, if an answer in a row overflows, the entire row is sent to the addendum. The overflow cross-reference text (for example, <i>See 1 in Addendum</i>) is merged in the affected row only.</li> <li>■ Select <b>Send individual cells</b> to send only the contents of a single overflowing field to the addendum.</li> </ul>

	The overflow cross-reference text (for example, <i>See 1 in Addendum</i> ) 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 <b>Cross-reference text</b> box.
Define the text that identifies the answer once it has been sent to the addendum	Enter the text in the <b>Addendum label text</b> box.
Make an answer the first item on its addendum page	Select <b>Begin addendum entry on a new page</b> .
Make an answer the last item on its addendum page	Select <b>Begin new page following addendum entry</b> .
Indent the answer a specific amount of space	Enter a number in the <b>Answer column indent</b> box (or click the up and down arrows to select a number).

# 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, a user enters an answer that is too long for the form field. One option the user has 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. As the template developer, 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 <b>Page margins</b> group.
Define the text that appears at the top of each addendum page	Enter the text in the <b>Header text</b> box.  To learn how to merge page numbers in the header, see <a href="#">Use Answer Overflow and Addendum Text Codes</a> .
Define the text that appears at the bottom of each addendum page	Enter the text in the <b>Footer text</b> box.  To learn how to merge page numbers in the footer, see <a href="#">Use Answer Overflow and Addendum Text Codes</a> .
Change the font properties (including font face, size, and style) of the header or footer text	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, click the <b>Font</b> button and make the changes at the <b>Font</b> dialog box.
Define how much vertical space the header or footer text requires	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, enter a number in the <b>Height</b> box.
Change the alignment of the header or footer text	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, click the <b>Alignment</b> button and choose your alignment

	option.
Insert a variable in the header or footer text	In the respective <b>Header properties</b> and <b>Footer properties</b> groups, click the  <b>Variable Field</b> button. The <b>Variable Field</b> 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 <b>Addendum entries</b> group, click the <b>Font</b> button and make your changes at the <b>Font</b> dialog box.
Define how much space there should be between the addendum label and the answer	In the <b>Addendum entries</b> group, enter a number in the <b>Indentation</b> box.
Define how much space there should be between each answer in the addendum	In the <b>Addendum entries</b> group, enter a number in the <b>Space between</b> box.
Make the addendum appear as a pleading paper	Select <b>Number lines to format as pleading paper</b> .

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.

# 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 <PAGES>* 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>	Current page in the document For example: <b>Header/footer Text:</b> Page <PAGE> <b>Inserted text:</b> Page 5
<PAGES>	Total number of pages in the document, including addendum pages For example: <b>Header/footer Text:</b> Total Pages: <PAGES> <b>Inserted text:</b> Total Pages: 5
<DOCPAGES>	Total number of pages in the document, without including the addendum For example: <b>Header/footer Text:</b> Total Document Pages: <DOCPAGES> <b>Inserted text:</b> Total Document Pages: 3
<PAGEINFO>	Current page in the document / Total pages in the document For example: <b>Header/footer Text:</b> Page <PAGEINFO> <b>Inserted text:</b> Page 4/5

<ADMPAGE>	<p>Current page in the addendum</p> <p>For example:</p> <p><b>Header/footer Text:</b> Addendum Page &lt;PAGE&gt;  <b>Inserted text:</b> Addendum Page 2</p>
<ADMPAGES>	<p>Total number of pages in the addendum</p> <p>For example:</p> <p><b>Header/footer Text:</b> Total number of addendum pages: &lt;ADMPAGES&gt;  <b>Inserted text:</b> Total number of addendum pages: 3</p>
<ADMPAGEINFO>	<p>Current page in addendum / Total pages in addendum.</p> <p>For example:</p> <p><b>Header/footer Text:</b> Addendum Page &lt;ADMPAGEINFO&gt;  <b>Inserted text:</b> Addendum Page 2/4</p> <p><b>Note:</b> This code inserts nothing if there is only one page in the addendum (instead of inserting <b>1/1</b>).</p>

## 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 the Assembled Document
<REFNR>	<p>Addendum item number</p> <p>For example:</p> <p><b>Cross-Reference Text:</b> See Addendum Item &lt;REFNR&gt;  <b>Inserted text:</b> See Addendum Item 1</p>
<VARNAME>	<p>The name of the variable</p> <p>For example:</p> <p><b>Cross-Reference Text:</b> See '&lt;VARNAME&gt;' in Addendum  <b>Inserted text:</b> See 'Case Description' in Addendum</p>

<p>&lt;ROW&gt;</p>	<p>Table row number (merged in addendum only)</p> <p>For example:</p> <p><b>Addendum Label Text:</b> Row &lt;ROW&gt;, Column &lt;COLUMN&gt;  <b>Inserted text:</b></p> <p>1. Row 1, Column 1:  Bentley Jones</p> <p>Row 1, Column 2:  Father</p> <p>Row 2, Column 1:  Mariah Jones</p> <p>Row 2, Column 2:  Mother</p>
<p>&lt;COLUMN&gt;</p>	<p>Table column number (merged in addendum only)</p> <p>For example:</p> <p><b>Addendum Label Text:</b> Row &lt;ROW&gt;, Column &lt;COLUMN&gt;  <b>Inserted text:</b></p> <p>1. Row 1, Column 1:  Bentley Jones</p> <p>Row 1, Column 2:  Father</p> <p>Row 2, Column 1:  Mariah Jones</p> <p>Row 2, Column 2:  Mother</p>
<p>&lt;MEMBER&gt;</p>	<p>Table cell sequence number (merged in addendum only)</p> <p>For example:</p> <p><b>Addendum Label Text:</b> Cell Number: &lt;MEMBER&gt;  <b>Inserted text:</b> Cell Number: 5</p>

## 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	<p>Arabic numerals (default)</p> <p>For example:</p> <p><b>Code:</b> Page &lt;PAGE:1&gt;  <b>Inserted Text:</b> Page 2</p>
A	<p>Uppercase letters</p> <p>For example:</p> <p><b>Code:</b> Page &lt;PAGE:A&gt;  <b>Inserted Text:</b> Page B</p>
a	<p>Lowercase letters</p> <p>For example:</p> <p><b>Code:</b> Page &lt;PAGE:a&gt;  <b>Inserted Text:</b> Page b</p>
I	<p>Uppercase Roman numerals</p> <p>For example:</p> <p><b>Code:</b> Page &lt;PAGE:I&gt;  <b>Inserted Text:</b> Page II</p>
i	<p>Lowercase Roman numerals</p> <p>For example:</p> <p><b>Code:</b> Page &lt;PAGE:i&gt;  <b>Inserted Text:</b> Page ii</p>

# Using Answer Files

---

## Overview: 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 can be saved as standard answer files (.ANS), or as XML answer files (.ANX).

You can organize your answer files with the Answer File Manager, or you can use Windows Explorer to manage your answer files. (See [Manage Your Answer Files](#).)

# Create a New Answer File

Often, you want to save answers you enter during an interview so they can be used with other interviews. HotDocs provides several options for doing this. Specifically, you can save a new answer file after completing an interview; save a new answer file based on answers in an existing answer file; or create a new, empty answer file using Answer File Manager.

## 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 **Type** drop-down list, select an answer file format. Your options include **.ANS** (native HotDocs format) and **.ANX** (XML-based format). (Saving answers in XML format allows you to integrate your answer files with other programs more easily.)
4. In the **Filename** box, enter a file name. (To save the answer file in a folder other than the default *Answers* folder, click  **Browse** and select a location.)
5. In the **Title** box, enter a name for the new answer file. This name will identify the answer file in Answer File Manager.
6. Optionally, type a note in the **Description** box 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.)

## To create a new answer file using answers from another answer file

1. At the HotDocs library window, select a template and click  **Assemble**. The **Answer File** dialog box appears. (See [Assemble a Text or Form Document](#).)
2. Click the  **Open Answer File** button. The **Open Answer File** dialog box appears.
3. In the answer file list, select an answer file, then click  **Select**. The **Answer File** dialog box appears again. (If the desired answer file isn't showing in the left pane, click  **Add**, then browse and select the desired answer file. See [Add an Answer File to Answer File Manager](#).)
4. Click **OK**. The assembly window appears, with answers from the selected answer file showing in the answer fields.
5. Complete the interview and close the assembly window when you are finished. HotDocs prompts you to save your answers.
6. Choose **Save As**. The **Save Answer File** dialog box appears.
7. In the **Type** drop-down list, select an answer file format. Your options include **.ANS** (native HotDocs format) and **.ANX** (XML-based format). (Saving answers in XML format allows you to integrate your answer files with other programs more easily.)
8. In the **Filename** box, enter a file name. (To save the answer file in a folder other than the *Answers* folder, click the  **Browse** button and select a location.)
9. In the **Title** box, type a name for the new answer file. This name will identify the answer file in Answer File Manager.
10. Optionally, type a note in the **Description** box to help identify the answer file. (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.)

## To create a new, empty answer file using Answer File Manager

1. At the HotDocs library, click  **Answer File Manager**. The **Answer File Manager** appears. (See [Overview: Use Answer Management](#).)
2. Click  **New**. The **New Answer File** dialog box appears.
3. In the **Type** drop-down list, select an answer file format. Your options include **.ANS** (native HotDocs format) and **.ANX** (XML-based format). (Saving answers in XML format allows you to integrate your answer files with other programs more easily.)
4. In the **Filename** box, enter a file name. (To save the answer file in a folder other than the *Answers* folder, click the  **Browse** button and select a location.)
5. In the **Title** box, type a name for the new answer file. This name will identify the answer file in Answer File Manager.
6. Optionally, type a note in the **Description** box to help identify the answer file. (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.)

**Note:** 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 an XML Answer File

At times, you may need to save answers to a database, a third-party program, or an Internet site using HotDocs Server. HotDocs lets you save answer files in XML format so you can integrate your answers with any of these types of programs. (See [What is XML?](#))

## To save an answer file in XML format

1. At the HotDocs library, select a template and assemble the document. (See [Assemble a Text or Form Document](#).)
2. After completing the interview, click the  **Save Answers** button. The **Save Answer File** dialog box appears.
3. At the **Type** drop-down box, select **HotDocs XML Answer File (.anx)**.
4. Enter a file name in the **File name** box. If you want to save the answer file in a folder other than the *Answers* folder, click the  **Browse** button and select a location.
5. Enter a title in the **Title** text box, and a description in the **Description** box. (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.)
6. Click **OK**. The answers are saved with the .ANX file name extension.

**Warning:** Only people who have a solid understanding of XML should attempt to manually modify the contents of an XML answer file. That said, XML answer files are text based, so they can be created and edited with any text editor. If you manually create an XML answer file, most answers can appear in any given order. However, answers that are part of a repeated dialog must appear in the same order as they appear in the dialog.

# 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 *.ANS* or *.ANX*. 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 on the **Properties** tab of the HotDocs library window.

## 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** box, click the  **Browse** button and locate the folder that contains the component file. Then type the same file name, followed by an answer file name extension (either *.ANS* or *.ANX*). For example, if the component file is named *Invoice.cmp*, the answer file name would be *Invoice.ans*.
5. In the **Title** box, type a name for the default answer file.
6. Optionally, in the **Description** box, type a description for the answer file.

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:

To	Do This
Choose the format for the answer file	Click the <b>Type</b> drop-down button and select an answer file format. Your options include <b>.ANS</b> (native HotDocs format) and <b>.ANX</b> (XML-based format).
Enter the file name	At the <b>File name</b> box, enter a name. When you click <b>OK</b> , 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  <b>Browse</b> button and navigate to the location.
Enter the title that identifies the answer file in Answer File Manager	At the <b>Title</b> box, enter a name or accept the suggestion HotDocs makes.
Add information to help identify the file	At the <b>Description</b> box, type notes about the answer file's purpose or contents. (The description will appear at the <b>Properties</b> tab of the Answer File Manager. When searching for specific answer files, you can search for text in the description.)

### Notes:

- See [Overview: 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](#).)
2. 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](#).)

The assembly window appears again, using the newly selected answer file.

**Note:** When viewing the answer library, you can sort the answer files. To do this, select a folder and click  **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 box.

# 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.

**Warning:** 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** box, 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 answers to be in XML format, select **XML Format**.
5. Click **OK**. HotDocs opens an e-mail message and attaches the answer file.

# 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 assembly. (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** box, 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 information in the following table:

To	Do This
Ignore the current instance of the word and continue spell checking	Click <b>Ignore</b> .
Ignore all instances of the word and continue spell checking	Click <b>Ignore All</b> .
Correct only the current instance of the word and continue spell checking	Select an existing replacement from the <b>Change to</b> list (or type the replacement in the <b>Change to</b> box) and click <b>Change</b> .
Correct all instances of the misspelled word and continue spell checking	Select an existing replacement from the <b>Change to</b> list (or type the replacement in the <b>Change to</b> box) and click <b>Change All</b> .
Add the word to your personal dictionary so that the spelling checker will not question the word again	Click <b>Add</b> .
Display additional spelling alternatives for the unrecognized word	Click <b>Suggest</b> . 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 <b>Options</b> . (You can also change your options at the <b>HotDocs Options</b> dialog box. See <a href="#">Change Your Spell Checking Options</a> .)

# Managing Answer Files

---

## Overview: 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 [Overview: 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.)

# 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 the Answer File Manager

- At the HotDocs library, click the  **Answer File Manager** button. Answer File Manager appears.

## To close the Answer File Manager

- At the Answer File Manager, click the **X** in the upper-right corner of the dialog box.

**Note:** 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.

## To add an answer file to Answer File Manager

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** box, 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** box, enter a name for the answer file. This title will identify the answer file in **Answer File Manager**.
5. Optionally, in the **Description** box, 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.)

### Notes:

- If the answer file is located in 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. Once added, you can organize the files and modify the properties of the files. In addition, you can 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.

## To display the 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:

To	Do This
Create a new, empty answer file	Click the  <b>New</b> button. (See <a href="#">Create a New Answer File</a> .)
Add an existing answer file to the library	Click the  <b>Add</b> button. (See <a href="#">Add an Answer File to Answer File Manager</a> .)  To add multiple files one at a time, use <b>Ctrl</b> or <b>Shift</b> 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  <b>Remove</b> .
Simultaneously copy an answer file you receive to the <i>Answers</i> folder and add it to the answer library	Click  <b>Import</b> . The <b>Import Answer File Name</b> dialog box appears, where you can browse to the answer file and select it. When you click <b>OK</b> , 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  <b>Sort</b> button. The contents of the folder are reorganized.
Search for a specific answer file	Select <b>Find</b> , then type the text from the answer file's title or description for which you are searching in the text box. Only files that have matching text in their title or description are shown.

View the answers currently saved in an answer file	<p>Select the answer file, then click the <b>Contents</b> tab.</p> <p>In the <b>Contents</b> tab, you can sort the columns by clicking the column headings.</p>
View the history of when the selected answer file was used to assemble various templates	Select the answer file, then click the <b>Properties</b> tab.
Change the title, path, file name, history, or description of an answer file	<p>Select the answer file, then click the  <b>Properties</b> button and make the necessary changes.</p> <p><b>Warning:</b> Changing the file path does <i>not</i> change the location of the actual answer file. If you change the file path without using Windows Explorer to move the actual file to the new location, that answer file will not be accessible.</p>
Add a folder so that you can better organize groups of answer files	Click the  <b>Add</b> button, then select <b>Folder</b> at the <b>Type</b> drop-down box. (See <a href="#">Add, Modify, or Delete Folders in Answer File Manager.</a> )
Modify a folder's title or description	Select the folder, then click  <b>Properties</b> . (See <a href="#">Add, Modify, or Delete Folders in Answer File Manager.</a> )
Delete a folder	Select the folder, then click  <b>Remove</b> . HotDocs asks what you want to do with any answer files within the folder. (See <a href="#">Add, Modify, or Delete Folders in Answer File Manager.</a> )
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.
Use Windows Explorer to access answer files instead of Answer File Manager	Change the setting at the <b>HotDocs Options</b> dialog box. (See <a href="#">Manage Answer Files and Assembled Documents.</a> )

**Notes:**

■ Answer files are saved to the default *Answers* folder that is 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](#).

■ Answer files can have one of two file name extensions: .ANS (standard answer file) and .ANX (XML answer file).

# 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 as described in the following table:

To	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  <b>Properties</b> button. The <b>Item Properties</b> 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  <b>Sort</b> . The answer files are rearranged based on title.
Remove a folder	Select a folder and click  <b>Remove</b> . If you want to remove the items in the folder as well, select <b>Remove the unselected items contained in the selected folders</b> . Otherwise, any files referenced in the folder will be moved to the next folder level.

# Creating Template and Document Summaries

## Overview: Create a Summary

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.

A third kind of summary, the Variable Sheet, shows variables, their current answers, and other information about the variable. This information appears in spreadsheet format.

The following example is a question summary, which you can print out and complete away from the computer. Note that different variable types show a different type of blank space for an answer.

**Question Summary**

**Template:** Invoice  
**Answer File:** New Answer File

**Invoice Information**

Invoice Number

\_\_\_\_\_

Salesperson

Paul Kent  
 Lisa Kelly  
 Rachel Winthrop  
 James Nichols

**Customer Information**

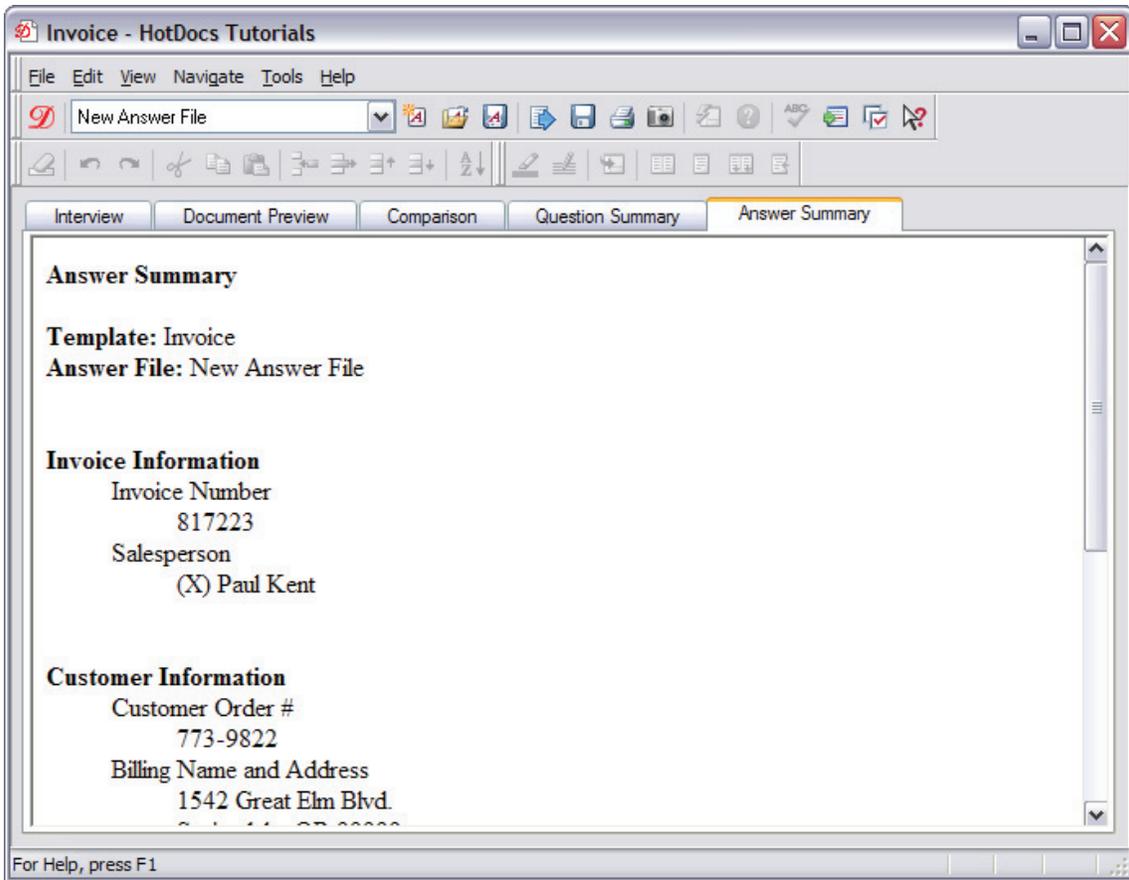
Customer Order #

\_\_\_\_\_

Billing Name and Address

For Help, press F1

The following example is an answer summary, which you can print out. An answer summary shows the questions that were asked and the answers that were given.



This example shows a record of the variable names, variable types, and answers currently used in the interview:

Final Judgment of Dissolution of Marriage with Property but No Children (Uncontested) (12.9...

File Edit View Navigate Tools Help

New Answer File

Interview Document Preview Comparison Question Summary Answer Summary Variable Sheet

Variable	Prompt	Answer	Type
<b>TF</b> Bold caption heading TF	Do you want to bold the caption ...	true	True/False
<b>TF</b> Print family law division TF	Do you want to print "Family Law...	true	True/False
<b>TF</b> Underline form title TF	The default format for the form ...	false	True/False
<b>A</b> Court judicial circuit TE	Judicial Circuit	Second	Text
<b>MC</b> Court county MC	County	Baker	Multiple Choice
<b>TF</b> Uniform case number TF	Have a reference number and u...	true	True/False
<b>A</b> Reference number TE	Reference Number	7772877	Text
<b>A</b> Uniform case number TE	Uniform Case Number	88377	Text
<b>TF</b> Include husband wife in caption TF	Should a designation such as "Wi...	true	True/False
<b>MC</b> Petitioner marital designation MC	Petitioner Designation	A	Multiple Choice
<b>MC</b> Respondent marital designation MC	Respondent Designation	B	Multiple Choice
<b>A</b> Petitioner name TE	Name	Jeffrey Walker	Text
<b>A</b> Petitioner address TE	Street Address	287 Candide Way	Text
<b>A</b> Petitioner city TE	City	Reveldale	Text
<b>MC</b> Petitioner state MC	State	Florida	Multiple Choice
<b>A</b> Petitioner ZIP TE	ZIP Code	00000	Text
<b>A</b> Petitioner phone TE	Telephone	(555) 555-5555	Text
<b>A</b> Petitioner fax TE	Fax	(555) 555-5554	Text
<b>A</b> Respondent name TE	Name	Kelly Walker	Text
<b>A</b> Respondent address TE	Street Address	1422 Evictus Lane	Text

For Help, press F1

#### Notes:

- 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, perform any of the following tasks:

To	Do This
Print a hard copy of the summary	Click the  <b>Print Question Summary</b> button.
Attach the summary (in HTML format) to an e-mail message to send to another person	Select <b>Send Question Summary To &gt; Mail Recipient</b> ( <b>File</b> menu).
Send the summary to the word processor so you can edit or format the text in it	Click the  <b>Send Question Summary to Word Processor</b> button. Once it is in the word processor, you can make changes to the summary. However, the changes you make are <i>not</i> reflected in the original template or question summary.
Save the summary as an HTML document	Click the  <b>Save Question Summary</b> 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 assemble 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:

To	Do This
Print a hard copy of the summary	Click the  <b>Print Answer Summary</b> button.
Attach the summary (in HTML format) to an e-mail message to send to another person	Select <b>Send Answer Summary To &gt; Mail Recipient</b> ( <b>File</b> menu).
Send the summary to the word processor so you can edit or format the text in it	Click the  <b>Send Answer Summary to Word Processor</b> button. Once it is in the word processor, you can make changes to the summary. However, the changes you make are <i>not</i> reflected in the original template or question summary.
Save the summary as an HTML document	Click the  <b>Save Answer Summary</b> 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, perform any of the following tasks using toolbar buttons and menus:

To	Do This
Sort the different lists in alphanumeric order	Click one of the column headings.
Save the variable sheet as an HTML document	Click the  <b>Save Variable Sheet</b> button. (You can also press <b>F12</b> .)
Send the information in the Variable Sheet to the word processor	Click <b>File &gt; Send Variable Sheet To &gt; Word Processor</b> . (You can also press <b>F11</b> .)
Attach the Variable Sheet (in HTML format) to an e-mail message	Click <b>File &gt; Send Variable Sheet To &gt; Mail Recipient</b> .
Copy the contents of the Variable Sheet into a spreadsheet program, such as Microsoft Excel  When sending Variable Sheet data to a spreadsheet, HotDocs automatically looks for Microsoft Excel. If you don't have Excel installed, you will be prompted to copy the contents from the Clipboard to another spreadsheet program.	Click <b>File &gt; Send Variable Sheet To &gt; Spreadsheet</b> .  <b>Warning:</b> Once the Variable Sheet is sent to a word processor or spreadsheet program, you can make changes to the information and save the document. However, the changes you make are not reflected in the original template, the assembled document, or the answer file.

# Using Published and Web-based Files

---

## Overview: Use Published Files on the Internet

When you install HotDocs, the installation program sets up HotDocs as a helper program for your browser if you use Netscape Navigator or 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 the installation is unable to set HotDocs up as a helper program for your browser, you can 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 auto-install (.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 and Netscape Navigator.

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 or Netscape Navigator)

1. 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. Make two new entries in the MIME types list using the following settings:

For HotDocs Auto-Assemble (.HDA) Files	
Name (optional)	HotDocs Auto-Assemble File
MIME type	application/x-hotdocs-auto
Extension	hda
Launch Application	hotdocs6.exe /ha=
<b>Warning:</b> By default, the launch application executable is found in <i>C:\Program Files\HotDocs 6</i> .	
For HotDocs Auto-Install (.HDI) Files	
Name (optional)	HotDocs Auto-Install File
MIME type	application/x-hotdocs

Extension	hdi
Launch Application <b>Warning:</b> By default, the launch application executable is found in <i>C:\Program Files\HotDocs 6</i> .	hotdocs6.exe /hi=

Save these settings and restart your browser.

**Note:** 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** box, 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**.

**Note:** 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** box, 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** box. The description appears at the **Properties** tab.

**Note:** 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).

# 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.
2. Locate and select the auto-install (.HDI) file you want to install, then click **Install**. The **Install Template Set** dialog box appears.
3. Click the **Choose where the library file should be installed** drop-down button and choose whether to save the library to the *Template Sets* folder or the *Libraries* folder.
4. At the **Specify where the templates should be installed**, accept the suggested *Template Sets* folder location, or click the  **Browse** button to choose a different location.
5. Click **OK**. HotDocs installs the template set (including the library) to the locations you specified.

# 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:

To	Do This
Install specific update items	Select <b>Install</b> for each item you want to install.
Discard an update item to prevent HotDocs from prompting you to install the update	Select <b>Discard</b> for the item. If this box is disabled, the template provider has marked it as a <i>required</i> 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 <b>Discarded updates</b> . The list changes to include discarded items.
View a list of all installed updates	Select <b>Installed updates</b> . The list changes to include installed updates.
Reinstall an update	Select <b>Installed updates</b> , then select the update you want to reinstall from the list.

4. Click **Continue**. HotDocs installs the selected updates.

**Note:** 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

1. 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.)
2. 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).

### Notes:

- 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.
2. Click the  **Add** button. The **Add Item** dialog box appears.
3. At the **Type** drop-down list, select **Web Address**.
4. In the **URL** box, enter the URL. Or, click the  **Browse** button to locate the address. Then close the browser window to save the URL to the **Add Item** dialog box.

**Note:** 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.

5. In the **Title** box, enter a title for the URL. The title identifies the item in the template library.
6. Optionally, type a description in the **Description** box. The description appears at the **Properties** tab.

# Creating Model Documents

---

## Overview: Create a Model Document

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.

Rather than search through the document for this text each time you need to customize the document for a new client, you can create a *model document*. A model document is an exemplar document used as the basis for drafting documents of the same type for specific clients. The Model Document Markup Language (MDML) defines markup used in model documents to identify text that varies from one client to another. MDML 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 to generate documents for your clients. If you are using HotDocs Professional, you can also convert your model documents 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 model document, you keep all of the different text variations in a single document, rather than storing the text in separate documents. Then, using MDML, you can mark up the document so that only the correct versions are used, depending on answers the user gives.

You mark up model documents directly in Microsoft Word. Your documents must be in DOC or RTF format.

**Note:** For full instructions on creating a model document, please visit <http://support.hotdocs.com/modeldocuments>.

For instructions on using model documents with HotDocs, see the following topics:

- [Convert Model Documents to HotDocs Template Format](#)
- [Convert HotDocs Templates to Model Documents](#)

# Convert Model Documents to HotDocs Template Format

**Note:** See <http://support.hotdocs.com/modeldocuments> for full instructions on creating model documents.

Once you have marked up a document, you can convert the model document to template format. Markup fields in the model are converted to variable and instruction fields. Once converted, you can edit the template or assemble it.

## To convert the document to template format

1. At the HotDocs library, choose **Convert Model to Template** (**Template** menu). The **Convert Model Document to HotDocs Template** dialog box appears.
2. Click the  **Browse** button next to the **Model document file name** box and locate the model document.
3. Enter a file path and name for the newly created template in the **HotDocs template file name** box. (HotDocs automatically suggests the same name as the model, with the text **.template** appended to it.)
4. Optionally, select **Add converted HotDocs template to library**. This places a reference to the template in the HotDocs library.
5. Click **OK**. The template is created from the marked up model document.

# Convert HotDocs Templates to Model Documents

**Note:** See <http://support.hotdocs.com/modeldocuments> for full instructions on creating model documents.

You can convert existing HotDocs templates to model documents. Saving templates as model documents allows you share your templates with subject matter experts unfamiliar with HotDocs. As long as those experts understand (and use) the markup language specified in this help file, they can further automate the template and use it with HotDocs Player to assemble documents from it.

When you convert a template to a model, 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.

**Note:** Some template development features are not supported in model documents. This includes pointed component files, clause libraries, INSERT, and ASSEMBLE instructions.

## To convert a template to a model document

1. At the template library, select the template you want to convert.
2. Click **Convert Template to Model** (Template menu). The **Convert HotDocs Template to Model Document** dialog box appears.
3. Optionally, change the name of the model document in the **Model document file name** box. (By default, HotDocs appends the text **.model** to the file name. This helps distinguish between it and the original template file.)
4. Optionally, select **Add converted model document to library** to add a reference to the model to the library. This allows you to more easily assemble the model document using the  **Assemble** command.
5. Click **OK**. HotDocs converts the template. You can further edit the model or distribute copies of it to others for review or additional markup.

# Troubleshooting: Template Development

---

## '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 2008-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 2008 Files with Earlier Versions of HotDocs](#) for details.

# 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 is 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**.)
6. 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 [View the Contents of a Template Manager Window](#).

# 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 a Single Template to a New File Format](#) and [Convert Templates and Clauses to Microsoft RTF](#) for details.)

Once converted, you should review each template carefully to make sure it still looks and works correctly.

# 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 [Overview: 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** box.
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** box.

# 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.

The following table lists each warning, followed by a description of what it means:

Warning	What it Means
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	<p>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.</p> <p>For example:</p> <p>An example of this would be the following script:</p> <pre>SET TF Variable TO TRUE  ... (later in the template)  ASK Dialog</pre> <p>(where <i>Dialog</i> uses the variable, <i>TF Variable</i>.)</p> <p>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.</p> <p>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.</p>
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.

<p>Variables that are referred to in a dialog script and are set to a value later</p>	<p>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.</p> <p>For example:</p> <p>An example of this would be the following script:</p> <pre> SET Client TO "Husband"  ASK Husband Information  SET Client TO "Wife"  ASK Wife Information </pre> <p>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:</p> <p>The following information applies to the «Husband or Wife».</p> <p><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>.</p> <p>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.</p>
<p>Variables that are not asked when referred to but are asked later</p>	<p>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 <b>Ask automatically</b> 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.</p>

**Notes:**

- You should understand the circumstances under which a variable or dialog is asked. 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.

# 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  **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.)

### Notes:

- Two other tools can help you match opening instructions with their closing instructions. These are the  **Match Fields** tool and the  **Label Fields** tool. See [Match Instructions with END 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 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.

# HotDocs is Prompting to Convert the Template to HotDocs 2008

## Problem

I've just installed the latest version of HotDocs. Now, when I edit a template, it prompts me to convert it to HotDocs 2008 format. What does this message mean? What will happen if I click Yes?

## Solution

HotDocs 2006/2007/2008 uses a component file format different from earlier versions of HotDocs. When you edit a template, HotDocs must convert the template to the latest format.

Once converted, if you need to use the template with an earlier version of HotDocs, you must manually save the component file in that version. See [Compatibility of HotDocs 6 Files with Earlier Versions of HotDocs](#) and [Change Component File Properties](#) for details.

# HotDocs Showing Unanswered Result When Repeating a Number Variable

## Problem

I have a computation script that repeats a Number variable. I want to merge the total from the repeated list, but HotDocs simply merges an unanswered variable placeholder.

## Solution

When repeating a list of numbers to create a calculated total, you must first set the starting value of the result. To do this, enter **0** at the start of the script—before the opening REPEAT instruction. For example:

```
0  
REPEAT Product Information  
RESULT + Unit Price  
END REPEAT
```

**Note:** If there is another number you need to use as the starting value, enter it instead of **0**.

# 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 a Template or a Template Set as an Auto-Assemble File \(HDA\)](#).
- 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 Ltd. and publish the files for use with HotDocs Player. See [Register Published 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** box, enter the number of the rows you want to be visible.
3. Click **OK**.

**Note:** To limit the number of answers a user can enter, use the LIMIT instruction in a dialog script. See [LIMIT NUM](#) for details.

# Matching Pairs of Instructions in a Computation Script

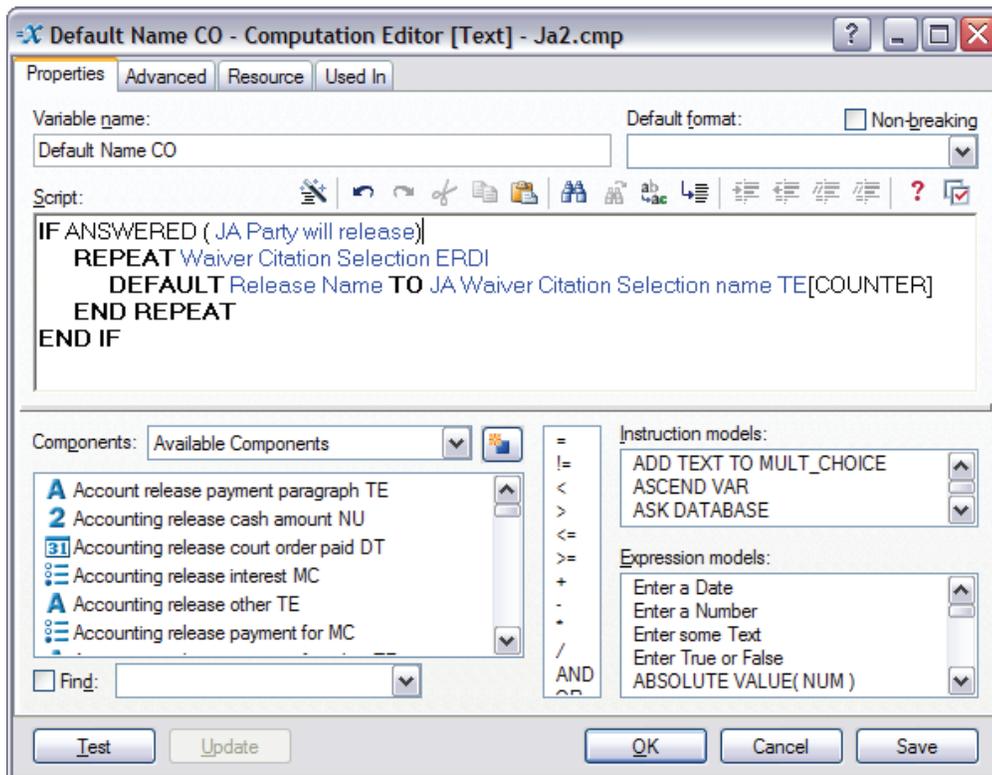
## Problem

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. (Following is an example of a script that has been auto formatted.)



- 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 Instructions with END 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** box.
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](#).)
2. At the **Format** box (either at the **Variable Field** dialog box or at the **Date Variable Editor**), enter **1990**.
3. 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:

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 be used as a keyword, which will cause HotDocs to read your variable name incorrectly. (See [Overview: Instruction and Expression Models](#).)
- Do not use any of the following characters when naming your components:

- . (period)
- \$ (dollar sign)
- " (quotation mark)
- :
- [] (brackets)
- ,
- () (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 [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 RTF 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.](#))
2. Create a dialog for the repeated variable. 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.
4. 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've added a REPEAT instruction to that cell, but when I assemble the document, it creates new rows for each repetition. How can I create the list in a single cell and not repeat the entire table?

## Solution

When you place a REPEAT instruction directly in a table, it will repeat each row of the table. To repeat the contents of a single cell, you must create a REPEAT instruction in a computation script and then insert that Computation variable in the cell. See [Create a REPEAT Instruction Using a Computation Variable](#).

# 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.

# 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  **Comment 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  **Uncomment Block** button. This enables the script once more.

**Note:** 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](#) and [Work with Multiple Choice Options](#) for details.

# 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 inserted template's headers will overwrite the main document'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 [Use Headers, Footers, and Margins 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 box 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. See [Work with Variables in Headers, Footers, Footnotes, and Text Boxes](#).

# Troubleshooting: 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.

# 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 Variable Answers](#).

# 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](#).

**Note:** 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- 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).

# Increasing the Font Size of Dialog Text

## 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](#).

**Note:** 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 **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](#).

**Note: 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](#).

# 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.

**Note:** 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 PDF-Based Documents Dialog Box](#) 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 PDF-Based Documents Dialog Box](#) 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 *hd6dispatch.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 *hd6dispatch.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 6*. However, if you performed a custom installation and changed the program folder for HotDocs to some location other than *C:\Program Files\HotDocs 6*, 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 *hd6dispatch.exe*.

### To update the registry with the correct information

**Warnings:** 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** box, type **regedit**. The Registry Editor appears.
3. Open **HKEY\_CLASSES\_ROOT > hdafile**. (To update the information for an HDI file, navigate to **HKEY\_CLASSES\_ROOT > hdi file**.)
4. Expand the folder so you can see its subfolders.
5. Click the **Default Icon** folder.
6. In the right pane, double-click **(Default)**. The **Edit String** dialog box appears.
7. Change the folder path in the **Value data** box 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.

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.

**Warnings:** 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 restore default settings

1. At the **Start** menu, select **Run**. The **Run** dialog box appears.
2. In the **Open** box, type **regedit** and click **OK**. The **Registry Editor** appears.
3. Navigate to **HKEY\_CURRENT\_USER > Software > LexisNexis > HotDocs 6**.
4. Select the **HotDocs 6** subfolder and press **Delete**. Click **Yes** to confirm the deletion.
5. Close the Registry Editor.

**Note:** To understand HotDocs installation and the changes that are made to the registry, please see [Understand HotDocs Installation](#).

# 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.

# 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.

**Note: 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**.

**Note:** 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 box. 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.

# Troubleshooting: 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 box.
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 Fields So Answers Can Flow From One Field to Another](#) and [Ungroup Fields](#).

**Warning:** 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](#).

**Note:** If you want all new templates you create to use a default interview, you can specify a HotDocs option. See [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 **Order/Size** 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 Actual Check Mark

## 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 an example using the *Symbol* font.

### To merge a check mark in a check box

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** box: **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.)
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 (✓) instead of the **X**.

# Addendum

---

## Names of Repeated Answers

HotDocs templates can repeatedly ask a question using a REPEAT instruction, which generates a list of answers. Templates created using HotDocs Professional can also contain nested REPEAT instructions up to four levels deep.

In addition to indicating the value type and variable name, repeated answer names must indicate the position of the answer in the list of answers for that variable. Position is indicated by appending a period and a number to the end of the 31 characters that indicate the value type and variable name, for example:

```
Ttenant .1
```

For answers that result from nested REPEAT instructions, one period and number will be appended for every repeat level, for example:

```
Ttenant .1.3.2
```

The first number represents the first repeat level and the last number represents the last (deepest) repeat level.

When the template developer gives you the answer names, he or she will also need to tell you which answers are repeated and at what level.

# Custom Library Properties Page Markup

You can customize the information displayed in the **Properties** tab using the following SPAN tags:

- **ID\_FILE:** Any information contained between this span tag and its close will be omitted when displaying the properties for a folder.
- **ID\_FOLDER:** Any information contained between this span tag and its close will be omitted when displaying the properties for a file.
- **ID\_HEADING:** The default heading for the library item type. For files this is *TEMPLATE PROPERTIES*. For folders, this is *FOLDER PROPERTIES*.
- **ID\_TITLE:** The title of the item (used for both files and folders in the library)
- **ID\_TYPE:** The type of item.
- **ID\_FILENAME** or **ID\_FILENAME\_LINKED:** The file name of the item. (If you append the text *LINKED* to the tag, HotDocs converts the file name to a hyperlink, which, when clicked, will browse to and select the file.)
- **ID\_FILEPATH** or **ID\_FILEPATH\_LINKED:** The file path for the item. (If you append the text *LINKED* to the tag, HotDocs converts the folder path to a hyperlink, which, when clicked, will browse to and open the folder.)
- **ID\_DESCRIPTION:** The description of the item (used for both files and folders in the library)

## Sample Code for Uploading an Answer File Using HotDocs Server

This script fragment receives the uploaded answers from the client, decodes the answers, and populates an AnswerCollection object with the answers. Once the answers are in the AnswerCollection object, they can be used like any other AnswerCollection object in HotDocs Server. This example creates an XML representation of the answers, which can easily be stored in a database.

```
<%  
    ' Create the AnswerCollection Object  
    dim ac  
    Set ac = Server.CreateObject ("HotDocs_Online.AnswerCollection")  
  
    ' Get the amount of data  
    dim count  
    count = Request.TotalBytes  
  
    ' Read the POSTed data  
    dim data  
    data = Request.BinaryRead (count)  
  
    ' Process the uploaded answers  
    ac.SetUploadedAnswers data, Cint(count)  
  
    ' Now that you have an AnswerCollection object with the uploaded  
    answers,  
    ' you can do anything you want with them. In this example, we get  
    an  
    ' XML representation of the answer set which could easily be stored  
    ' in a database.  
  
    ' Get a copy of the answers in XML format  
    dim answers  
    answers = ac.GetXMLAnswers ()  
    ' Now answers is a string containing the XML-format answers  
%>
```

## Publishing Destination Page Sample Code

The following is sample code for the Publishing Destination Page.

```
<%  
    dim dund  
    Set dund = Server.CreateObject("Dundas.Upload.2")  
    dund.SaveToMemory  
  
    response.Write "Data from form<hr>"  
    dim objUploadedForm  
    For Each objUploadedForm in dund.Form  
        Response.Write objUploadedForm.tagname & "="  
    Response.Write objUploadedForm.Value & "<br>"  
    Next  
    response.Write "<br>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\" & filen  
        objFile.SaveAs filename  
        Response.Write "<a ref='http://www.company.com/temp/"  
    Reponse.Write filen & "'>"  
        Response.Write objFile.TagName  
    Response.Write "=" & objFile.OriginalPath  
    Response.Write " : Size = " & objFile.Size  
    Response.Write "</a><br>"  
    Next  
    dund.ImpersonationTerminate  
    set dund = nothing  
%>
```

## Publishing Form Page Sample Code

The following is sample code for creating Publishing Form Page:

```
<form method="POST"
name="HotDocsPubForm"
action="http://www.company.com/hotdocsuploadproc.asp" enc-
type="multipart/form-data">
Template Name: <input type="text" name="textinput" size="65"><br>
Template Desc: <textarea rows="2" name="textareaDesc" cols="55">
Template Description
</textarea><br>
<select size="1" name="dropdownbox">
<option value="choice1">choice1</option>
<option value="choice2">choice2</option>
</select>
<input type="submit" value="Submit">
</form>
```

## URL-Encoded Text

In URL-encoded text, alphanumeric characters and the dollar sign, minus sign, underscore, period, exclamation mark, asterisk, left and right parentheses, and comma are transmitted as normal text.

All other characters (such as quotation marks and spaces) are encoded to facilitate their transmission over the Internet. In the encoded text, these characters are replaced with a percent sign followed by a two-digit hexadecimal number representing the ASCII value of the original character. (For example, a space character would be replaced with %20 since 20 is the hexadecimal number for 32, which is the ASCII code for a space.)

# 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*.
2. 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.)

**Warning:** When a user opens a library by clicking the HotDocs button in the word processor toolbar, no splash screens are displayed.

# What is XML?

XML (eXtensible Markup Language) 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.

An XML document can reference what is called a DTD (document type definition). A DTD validates the XML to ensure proper use of tags and the correct exchange of data with HotDocs. When you save a HotDocs answer file in XML format, the DTD is also written to the answer file so the contents of the file can be validated when read later.

Several resources are available on the Internet and in print that contain information on XML, including the official documentation of the World Wide Web Consortium. The W3C is a non-profit organization responsible for setting Internet standards. Their Web address is <http://www.w3c.org>, and they provide a useful index that references XML technology.

## **To view a copy of the HotDocs XML DTD**

- Click [http://support.hotdocs.com/dtd/hotdocs2005\\_ans.dtd](http://support.hotdocs.com/dtd/hotdocs2005_ans.dtd).

# Document Text Editor Toolbar

While editing the text of the assembled document at the Document Text Editor, you can use any of the following options:

Button Name	What it Does
 <b>Save and Close</b>	Saves your changes, closes the Document Text Editor, and merges the new text in the assembled document.
 <b>Restore Original Text</b>	Discards any changes you've made to the text and restores the original document text.
 <b>Print</b>	Prints a copy of the text you are editing.
 <b>Undo</b>	Undoes the most recent change to the text you are editing.
 <b>Redo</b>	Reapplies the most recent change to the text you are editing
 <b>Cut</b>	Removes the selected text and saves it to the Windows Clipboard. You can then paste the text using the <b>Paste</b> command.
 <b>Copy</b>	Copies the selected text and saves it to the Windows Clipboard. You can then paste the text using the <b>Paste</b> command.
 <b>Paste</b>	Pastes any text saved to the Windows Clipboard at the cursor position.
 <b>Find</b>	Displays a dialog box where you can specify the text for which you are searching.
 <b>Find Next</b>	Searches the document for the next instance of text for which you are searching.

 <p><b>Replace</b></p>	<p>Displays a dialog box where you can specify a text string for which you are searching, as well as text you'd like to use as a replacement.</p>
 <p><b>Show Ruler</b></p>	<p>Displays a ruler in the Document Text Editor toolbar.</p>
 <p><b>Show Codes</b></p>	<p>Displays editing codes, such as paragraph marks, in the document.</p>
 <p><b>Spell Check</b></p>	<p>Checks the spelling of text in the document.</p>
 <p><b>Close Without Saving Changes</b></p>	<p>Closes the Document Text Editor without saving the changes you have made.</p>

## Assembly Queue Statuses

The following table describes the different statuses of templates listed in the assembly queue:

Template Status	Description
Confirmed	The assembly queue has confirmed the location of the template and is ready to assemble it.
Assembling	HotDocs is currently assembling this document.
Pending	This template is in the queue because of instructions in the template currently being assembled. After the current interview is finished, the <b>Pending</b> status changes to <b>Confirmed</b> .
Complete	Assembly is finished. <i>Complete</i> status doesn't necessarily mean that all the questions in a template have been answered. For example, when creating a question summary, even though the assembly window opens, no questions are answered, but HotDocs still considers the assembly <i>complete</i> .
Error	An error has occurred, preventing the interview from being completed.

# Troubleshooting Form Printer Problems

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.

# Understand Bar Code Settings

When you convert an answer to a PDF417 bar code format, you can control some of the settings that are used to generate the bar code. These specifications are frequently determined by the agency to which you are submitting your form. Please check with the agency to determine which settings to specify.

- **X-Dimension (module width):** The X-dimension is the width of the narrowest bar in a printed bar code. The smaller the number, the stricter the requirements are for printing the bar code. (For example, the bar code must be printed on higher quality paper, etc.)
- **Module aspect ratio:** This determines the range of angles with which the bar code can be scanned. For example, when placed under the scanner, must the paper be flat and facing a single direction, or will the scanner be able to read the data regardless of how it's placed? The higher the number, the less restrictive the placement of the paper under the scanner will be.
- **Number of rows and Number of columns:** These numbers control the number of rows and columns that appear in the bar code. These settings determine the width and/or height of the bar code.
- **Error level:** This determines the redundancy of information in the bar code to compensate for any damage that may be done to the bar code (for example, if the paper is torn or if other marks have been made on the bar code). The higher the number, the less data can be stored in the bar code. (However, the greater the chances are that the bar code will be readable.)

Additional information about the PDF417 bar code specification can be found on the Internet.

## Custom Command Sample

The following is an example of a custom command:

```
SELECT Client.Company, Client.Contact, Client.Address1,  
Client.Address2, Client.City, Client.State_Prov, Client.Country,  
Client.PostalCode, Invoice.Inv_Number, Invoice.Inv_Date,  
Invoice.Ship_Date, Invoice.Inv_Total, Invoice.Date_Paid FROM Client  
INNER JOIN Invoice ON Client.ClientID = Invoice.ClientID ORDER BY  
Client.Company;
```

# 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**.

# Component Properties

When working with literal text in components, you can search, search and replace, print, or spell check text used in the different properties of a given component. Specifically:

Using both the  **Find in Components** and  **Find and Replace in Components** buttons, these properties are:

- **Display text (titles, prompts, etc.):** Searches for text in dialog and variable titles, as well as variable prompts.
- **Merge text, options, patterns, and formats:** Searches for the specified text string in Multiple Choice merge text groups, options, Text variable patterns, and variable formats.
- **Plain text resources:** Searches for the specified text string inside any plain text resources for variables or dialogs.
- **Scripts:** Searches for the specified text string inside quotation marks in computation and dialog scripts.
- **Variable references (Find option only):** Searches for variables that are referenced in the selected variables' prompts, merge texts, and scripts.

Using the  **Spell Check Components** button, these properties are:

- **Display text (titles, prompts, etc.):** Checks the spelling of text in dialog titles and variable prompts and titles.
- **Merge text:** Checks the spelling of merge text in Multiple Choice variables.
- **Plain text resources:** Checks the spelling of plain text resources for variables and dialogs.
- **Scripts:** Checks the spelling of text strings (inside quotation marks) in computation scripts.

Using the  **Print Components** button, these properties are:

- **Component types:** Prints the type of component for the variable you are printing (for example, Text, Date, Number, True/False, Multiple Choice, Computation, and Personal Information)
- **Titles, prompts, and dialog elements:** Prints each component's prompt or title, if one has been assigned. Prints dialog element text in a dialog.
- **Scripts:** Prints any scripts associated with the variable or dialog.
- **Plain text resources:** Prints any plain-text resources that have been assigned to the variable.
- **All other component type-specific properties:** Prints other component-specific data such as the maximum number of characters for a Text variable or the contents and script for a custom dialog.

# Tips for Working With Answer Sources

You should 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 .ANS 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** box.
- 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. (See [Share Answers Between Two Dialogs](#).) 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.)

# Differences Between ASSEMBLE and INSERT Instructions

The way HotDocs processes an ASSEMBLE instruction is different from the way it processes an INSERT instruction.

- With ASSEMBLE, HotDocs finishes assembling the main document and waits until you close the assembly window before starting the next assembly. 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.
- The INSERT instruction interrupts assembly of the main document (if the instruction is placed before the end of the template) to assemble the inserted template. HotDocs then inserts the assembled text where the instruction is, and then finishes assembling the main document.

# 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
<p>Sets the value of a variable each time the instruction is processed, even if the variable is already answered.</p> <p><b>Warning:</b> 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.</p>	<p>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.)</p>
<p>When the instruction is processed, causes HotDocs to prompt the user to save the answer file. (To avoid warnings like this, clear the <b>Save in answer file</b> option for the variable (<b>Advanced</b> tab of the <b>Variable Editor</b>.)</p>	<p>When the instruction is processed, does not cause HotDocs to prompt the user to save the answer file.</p>

# Foreign Language Number and Date Formats

The following is a list of formats you can customize for foreign language Number and Date variable formats:

- Alphabetic cardinal numbers (nine, one thousand nine hundred and ninety)
- Ordinals both in dates and numbers (9th, ninth)
- Month names and abbreviations (June, Jun)
- Weekday names and abbreviations (Sunday, Sun)
- The keyword percent used in a number format
- The keyword minus used in a number format
- A comma, period, or spaces used as a thousands separator
- A comma or period used as a decimal point

You should be familiar enough with the foreign-language equivalent of these words and phrases to ensure proper spelling and punctuation.

## View the Component's Status

The following icons are used to describe how components are being used in the component file. Template Manager can overlay these icons on components to help you know the status of the component—whether it is being used correctly or incorrectly:

 Components that have a red R next to them are components that are referenced in the template (either directly or indirectly, as in the case of variables used in dialogs or Computation variables), but do not appear in the component file. Their status is **Missing**.

 Components that are grayed out with an overlay of a grayed question mark are components that appear in a component file but are not referenced in the template. Their status is **Unused**.

Components that are used in the component file and referenced in the template file have the **OK** status assigned to them. They are not represented by any special icons.

## View the Template Status

The following icons are used to describe how templates are being used in the library. Template Manager overlays these icons on items in the file list to help you know the status of the file.

 This symbol means HotDocs has detected a problem with the file. Usually, this means the file refers to another file that doesn't exist.

 This symbol means the file listed doesn't exist.

 This symbol means the file is referred to in the template using an ASSEMBLE instruction. In such cases, Template Manager adds the file to the main level of the file list so you can work with it.

# Currency Symbols

Currency symbols are not merged into the document—they only appear in the answer field to help the user know what type of information to enter. If you want a currency symbol to merge in the document, create an example format that contains the currency symbol, or make sure the symbol is included in the template text.

If the currency symbol you want is not available, you can create it by typing it in the **Currency box**. Currency symbols can be up to three characters long.

# Additional Support Information for Folio Views 4.2

## Differences Between Versions

HotDocs provides the same level of support for Folio Views 4.2 that was provided for Views 3.x in earlier versions of HotDocs. Developers who create templates with help links to Folio infobases should note the following:

- With Views 4.x versions prior to 4.2, most notably 4.11, a new instance of Views is started each time an infobase is accessed from within HotDocs. This is due to an error in Views that has been corrected in version 4.2.
- Views 4.x infobases cannot be opened in Views 3.x.
- Version 3.x infobases can be used with Views 4.2. However, Views 4.2 does not handle jump destinations passed by HotDocs correctly with 3.x infobases. The 3.x infobase comes up in Views 4.2, but Views does not move to the jump destination.

## Command-Line Options

Two command-line options help you identify the version of infobase files distributed with HotDocs template sets. These options ensure that HotDocs looks for the correct version of Views to display the help files. The options operate as follows:

**/v3** - identifies a Views version 3.x infobase

**/v4** - identifies a Views version 4.x infobase

If a user may have both versions of Views installed, or if you want to make sure HotDocs does not attempt to open an infobase in the wrong version of Views, specify the infobase version by appending the correct command-line option following the **Infobase file** entry in the **Resource** tab. For example:

```
TCAHelp.nfo /v3
```

This file name identifies the infobase *TCAHelp.nfo* as a Views version 3.x infobase.

## Tips on Naming True/False Variables

If you are using the True/False variable's name as the prompt in the answer-gathering dialog, you may want to consider naming a True/False variable after the section of the template it affects—for example, *Marriage Clause*. This makes sense if the user is going to be selecting from a list of such sections—*Marriage Clause*, *Children Clause*, *Out-of-State Residence Clause*, and so forth.

Or, you can consider naming the variable after the condition itself—*Client Is Married*. This makes sense if the user will not be selecting from a list of sections—especially if the same True/False variable might be used as the condition for more than one part of the template.

# Tips on Using Unanswered Variable Placeholders

You can specify unanswered text placeholders four different places in HotDocs:

**Variable field:** When specified at the **Variable Field** dialog box, the placeholder is used only for that specific instance of the variable. It overrides any placeholders you specify at the **Variable Editor**. (You can enter whatever placeholder text you like here.)

**Variable component:** When specified at the **Variable Editor**, the placeholder becomes the default for the variable and will be used when no field-specific placeholder has been specified. (You can enter whatever placeholder text you like here.)

**Component file:** When selected at the **Component File Properties** dialog box, the placeholder is applied to all variables within the component file, except those that have a default or a field-specific placeholder assigned. If **Default** is selected, whatever options the user has specified at **HotDocs Options** will be used for those not already assigned a specific placeholder.

**HotDocs Options:** When users assemble documents, they have some control over which placeholder will be used if they leave variables unanswered. (They make their selection at the **HotDocs Options** dialog box.) However, if you have assigned placeholders using any of the methods above, your selection will override theirs.

## Inserted Template Locations

Templates can be INSERTed or ASSEMBLEd from any location, as long as you specify the correct folder path information. For example, templates can be inserted using the following folder structure:

Location of Template	Example
Same folder as parent template	«INSERT "Template.rtf"» «ASSEMBLE "Template.rtf"»
Subfolder of parent template	«INSERT "Subfolder\Template.rtf"» «ASSEMBLE "Subfolder\Template.rtf"»
Full file path	«INSERT "C:\My Documents\Template.rtf"» «ASSEMBLE "C:\My Documents\Template.rtf"»
Reference path	«INSERT "^referencePath\Template.rtf"» «ASSEMBLE "^referencePath\Template.rtf"»
Default <i>Templates</i> folder	«INSERT "\Template.rtf"» «ASSEMBLE "\Template.rtf"»

# HotDocs Glossary

---

## **.ANS**

File name extension that designates that the file is a HotDocs answer file.

## **.ANX**

File name extension that designates that the file is an XML answer file.

## **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.)

## **answer**

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.

## **answer file**

A saved file that contains the answers entered during an interview. Often users save their answers in a file so they can use them to assemble other similar documents.

## **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

this 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.

#### **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, assembly**

*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 its button () 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, automation**

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.

#### **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.)

#### **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*.

**.CMP**

File name extension that designates that the file is a component file.

**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.

**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**

An element in a HotDocs template that displays or stores information about the answers that are merged. 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. 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.

**.DOC**

File name extension that designates that the file is a Microsoft Word document. (*See* text document.)

**.DOT**

File name extension that designates that the file is a Microsoft Word DOT template. (*See* text template.)

**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**

1. In template development, represents the component in which the developer groups variables and other components.
2. 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.

**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. 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 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.

**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.) Users now must use HotDocs PDF Advantage to create PDF-based form templates. (HotDocs still supports automation of HFT files—just not creation of these files.)

**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.

**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.

**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.

**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.

**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.)

**.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.

**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 Compare**

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 Filler**

The application used to view assembled form documents.

**HotDocs Markup Tool**

An application used to mark up documents so they can then be automated in HotDocs. Markup is usually performed by a subject matter expert. During the process, the expert identifies (or marks) sections of the document that will change depending on the user. Information about these sections is then saved in a markup database so that it can be used to generate templates from the documents (using the HotDocs Template Set Generator.)

**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 tool that allows the creation and automation of PDF-based form templates. PDF Advantage can also be used to save most types of documents as PDF (assembled or otherwise).

**HotDocs Player Edition**

A version of HotDocs that is used for assembling published (and registered) templates.

**HotDocs Professional Edition**

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 Standard Edition.)

**HotDocs Server**

The Web-based version of HotDocs. When using HotDocs Server, interviews are presented in a user's Web browser. Answers are then sent back to a server where the document can be assembled. HotDocs Server allows users to create documents and answer files without requiring them to have HotDocs installed on their desktop.

**HotDocs Standard Edition**

A version of HotDocs that contains the tools necessary to automate a simple to moderately complicated set of text templates. HotDocs Standard can also be used to assemble text and form documents. (*See also* HotDocs Professional Edition.)

**HotDocs Template Set Generator**

The tool used convert marked up documents to template format. (*See also* HotDocs Markup Tool.)

**IF instruction (*also* ELSE IF, ELSE, END IF)**

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**

1. 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.
2. 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 that contains that text can be created and inserted in each template that requires it (via an INSERT instruction). This way, if changes need to be made to the text, the change has to be made in only one template.

**inserted dialog**

*See* child dialog.

**inserted template**

A template that is inserted into another template 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 presentation of questions that must be answered in order to create an assembled document. The interview is viewable by clicking the **Interview** tab of the assembly window.

**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 other entity. 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.

**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.

**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.

**label**

1. 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.

2. 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.)

**list**

Two or more answers to one question merged in the document.

**manual index**

*See explicit index.*

**map, mapping**

*See variable mapping.*

**mark up, markup**

1. The process a subject matter expert goes through to identify and mark changeable information in a document so that a template developer can look at the document and know how to automate it. The HotDocs Markup Tool expedites this process by automating the markup process. (Once a document is marked up, it can be converted to template format using the Template Set Generator.) (*See also* HotDocs Markup Tool.)

2. 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 Tool**

*See* HotDocs Markup Tool.

**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 field**

1. During template development, the place in the template where a variable is inserted.
2. 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.

**navigation bar**

1. In an interview (at the **Interview** tab), the toolbar used to move from dialog to dialog.
2. 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.

**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.

**.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 (with HotDocs PDF Advantage) installed. With PDF Advantage, template developers can also create PDF-based form templates. They can also create PDFs from assembled documents.

**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.

**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*).

**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.

**.RTF**

File name extension that designates that the template file is a Word RTF file. (*See* text template.)

**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 two or more answers can be entered. 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.

**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.

**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 script-writing 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 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.)

**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 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).

**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 the templates in the set. 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* 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.

**Template Set Generator**

*See* HotDocs Template Set Generator.

**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 variable**

A type of component that determines a true/false status of some condition and then merges the appropriate answer or text.

**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.*)

**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.*)

**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.)

**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 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.

**.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.)

**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.

**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. HotDocs can save answer files in XML format so the answer file can be integrated with other applications.

# Third-Party Software Acknowledgments

---

## Apache Software Foundation

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). The following information applies to the Apache software:

Copyright © 1999-2000 The Apache Software Foundation. All rights reserved.

## 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.

# End-User License Agreement

---

This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and HotDocs, a division of HotDocs Limited, ("HotDocs") for the HotDocs software, which includes computer software and may include associated media, printed materials, and "online" or electronic documentation (collectively the "Product"). By installing, copying, downloading, accessing, or otherwise using the Product, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, do not install or use the Product; you may, however, return it to your place of purchase for a full refund.

1. The Product is protected by copyright laws and international copyright treaties and other intellectual property laws and treaties. The Product is licensed, not sold.
2. As long as you comply with the terms of this Agreement, HotDocs grants to you a non-exclusive license to use the Product on a single computer or workstation.
3. Any software provided along with the Product that is associated with a separate end-user license agreement is licensed to you under the terms of that license agreement.
4. You may also store or install a copy of the Product on a network server, used only to run the Product on your other computers over an internal network; however, you must first acquire and dedicate a license for each separate computer on which the Product is run from the storage device. Notwithstanding the foregoing, the product may be used on one additional personal computer such as a home personal computer or other personal computer owned or controlled by you provided that your use of the product is only on one personal computer at a time. A license for the Product may not be shared or used concurrently on different computers.
5. You may make one backup copy of the Product, provided your backup copy is not installed or used on any computer. You may not transfer the rights to a backup copy unless you transfer all rights in the Product.
6. The Product is licensed as a single product. Its component parts may not be separated for use on more than one computer.
7. You may not rent, lease, or lend the Product or use the Product as part of a service bureau.
8. If you are using an evaluation version of the Product, your use must only be in conjunction with evaluating the Product and for no other purposes.
9. All rights not expressly granted are reserved by HotDocs.
10. HOTDOCS AND ITS SUPPLIERS' ENTIRE LIABILITY AND YOUR EXCLUSIVE REMEDY SHALL BE, AT HOTDOCS' OPTION, EITHER (A) RETURN OF THE PRICE PAID, IF ANY, OR (B) REPAIR OR REPLACEMENT OF THE PRODUCT THAT DOES NOT MEET HOTDOCS' LIMITED WARRANTY AND WHICH IS RETURNED TO HOTDOCS WITH A COPY OF YOUR RECEIPT. THIS LIMITED WARRANTY IS VOID IF FAILURE OF THE PRODUCT HAS RESULTED FROM ACCIDENT, ABUSE, OR MISAPPLICATION. ANY REPLACEMENT PRODUCT WILL BE WARRANTED FOR THE REMAINDER OF THE ORIGINAL WARRANTY PERIOD OR THIRTY (30) DAYS, WHICHEVER IS LONGER.
11. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, HOTDOCS AND ITS SUPPLIERS DISCLAIM ALL OTHER WARRANTIES AND CONDITIONS, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, AND NON-INFRINGEMENT, WITH REGARD TO THE PRODUCT, AND THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY HAVE OTHERS, WHICH VARY FROM STATE/JURISDICTION TO STATE/JURISDICTION.
12. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL HOTDOCS OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION,

DAMAGES FOR LOSS OF BUSINESS PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, OR ANY OTHER PECUNIARY LOSS) ARISING OUT OF THE USE OF OR INABILITY TO USE THE PRODUCT OR THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT SERVICES, EVEN IF HOTDOCS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. IN ANY CASE, HOTDOCS' ENTIRE LIABILITY UNDER ANY PROVISION OF THIS EULA SHALL BE LIMITED TO THE AMOUNT ACTUALLY PAID BY YOU FOR THE PRODUCT. BECAUSE SOME STATES AND JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY, THE ABOVE LIMITATION MAY NOT APPLY TO YOU.

13. Maintenance and Support Services are provided under a separate agreement.
14. Upon request by HotDocs, not more than once annually, you will report the number of copies in use by you.
15. If you acquired this Product in the United States, this EULA is governed by the laws of the State of New York.