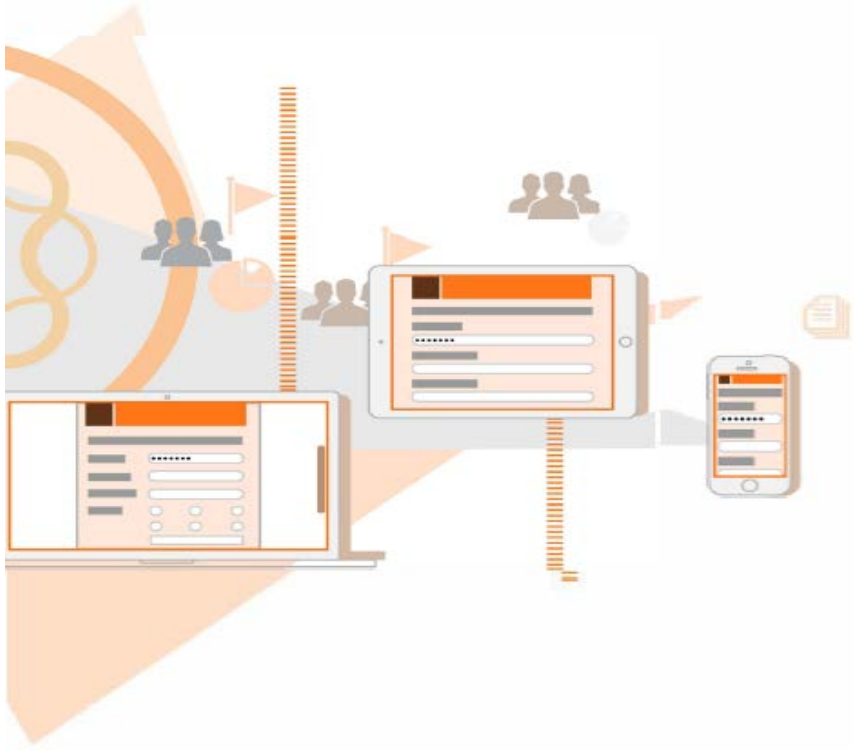
Using Designer



AEM Forms 6.5

Legal notices

For legal notices, see http://help.adobe.com/en_US/legalnotices/index.html.

Contents

Chapter: 1 **Welcome to Designer 1**

About Designer 1
New features in Designer 1
Preview your XDP forms in HTML in Designer 1
Preview your XDP forms in HTML in forms manager 1
Accessibility Checker 2
Add document title to the PDF title bar 2
Where to find documentation, samples, and tutorials 2
Get started with samples and tutorials 2
Add calculations to forms 3
Enhance forms by using scripts 3

Chapter: 2 **Getting Started 4**

About forms 4
Interactive forms 4
Non-interactive forms 5
Print and fill forms 5
About form designs 5
Form design layouts 6
Fixed layout 6
Flowable layout 6
Parts of a form design 7
Master pages 7
Pages 8
Content areas 8
Subforms 8
Tables 8
Field objects 9
Boilerplate or static objects 9
Installing and configuring Designer 10
Configuring JVM settings 10
Looking at the workspace 10

About editors 11
Layout Editor 11
Script Editor 12
About palettes 13
Hierarchy palette 14
Data View palette 14
Tab Order palette 14
PDF Structure palette 15
Object Library palette 16
Fragment Library palette 16
Style Catalog palette 16
Layout palette 16
Border palette 17
Object palette 17
Accessibility palette 17
Font palette 17
Paragraph palette 17
Drawing Aids palette 18
Info palette 18
Report palette 18
How To palette 18
Customizing the workspace 19
To customize the Layout Editor 19
To customize the Script Editor 19
To customize the palettes 20
To use the drawing aids 20
To customize the toolbars 21
To customize the tool tips for objects 23
Managing library palettes 23
Object Library palette 23
Fragment Library palette 27
Fragment Library palette 27
Steps to creating a form design 28
Basic steps to creating a form design 28
Plan the form design 28
Create the form design 29
Start with a sample form 29
Considerations when creating form designs 29
Save the form design 30
Preview and test the form design 30
Publish the form 31
Distribute the form 31

Chapter: 3 Create Form Designs 32

Create, open, and save 32
Creating forms 32

Creating forms with the stand-alone version of Designer 32
Creating forms from Workbench 33
Creating forms based on a template, sample, or existing form . 33
Using forms as a model for Guides 33
Creating forms by importing a PDF 34
Considerations for creating forms for use with Acrobat and Adobe Reader 34
Adding details about the form 35
To add information about a form 35
To add custom properties 35
To remove custom properties 36
To add document title to the PDF title bar 36
To set user permissions on a PDF form 36
Saving forms 36
To save a form design 37
To check in a form in Workbench 38
To save a copy of a form design 38
Opening forms 39
Selecting the Acrobat and Adobe Reader target version 39
Saving forms for Acrobat and Adobe Reader 40
Distributing forms 40
To distribute a form using Designer 41
Publishing forms 41
Publishing to the AEM forms repository 41
To publish a form to a repository 41
To publish a form to the repository 42
To set save options prior to publishing a form: 42
Publishing a form to a repository 42
Creating and managing templates 42
Creating a template 43
Managing templates 44
Importing a template 46
Creating and customizing a form based on a template 47
Page layout 47
Setting up pages 47
To show or hide a page 47
To add a page 48
To delete a page 48
To move between multiple pages 48
To reorder pages 48
To rename a page 48
Using master pages 49
To use master pages with repeating subforms 49
When to add a master page 49
To show or hide a master page 50
To add a master page 50

To delete a master page 50
To move between multiple master pages 50
To reorder master pages 50
To rename a master page 51
To add page numbering 51
To specify the size of a master page 51
To specify the orientation of a master page 51
Specifying white space around the edges of a form 51
Headers, footers, and other background items 53
Assigning additional master pages to complex form designs . . .53
To assign a master page to a specific page 53
Applying master pages to forms whose number of pages vary .55
To use the first master page one time only 56
To use the first master page one time only by specifying minimum and maximum page-occurrence values 56
To use the first master page one time only by specifying the master page's placement in the page set 56
Controlling the order and visibility of pages 56
To move between multiple page sets 57
To add or delete a page set 57
To reorder page sets 57
To rename a page set 57
To manage page sets by controlling the number of occurrences of each page 57
Setting page break controls 57
To control widow and orphan lines for new forms 58
To control widow and orphan lines for a form 58
To remove widow and orphan control from a form 59
Styles 59
To sort the style sheets in the Style Catalog 60
To select a default style sheet for new forms 60
To add a style sheet to a form 60
To create a new style sheet 61
To embed a style sheet 61
To extract styles to a new style sheet 61
To remove a style sheet 61
To replace a style sheet 62
To edit a style sheet file 62
Create a new style from an object 62
Creating a new style for an object type 62
To create a new style in the internal style sheet 63
To create a new style in an external style sheet file 63
Creating a common style 63
To create a common style in the internal style sheet 63
To create a common style in an external style sheet file 64
To apply a style to an object 64

To remove a style from an object 64
Editing a style 64
To edit a style in the internal style sheet 65
To edit a style in an external style sheet file 65
Deleting a style form a style sheet 65
To delete a style from the internal style sheet 65
To delete a style from an external style sheet 65
To set a default style for new objects 66
To apply default styles to new objects 66
To edit default styles for new objects 66
Printing forms 66
Setting up forms for printing 67
Specifying page placement and printing options in existing form designs 67
To specify page placement and printing options in existing form designs 68
To specify double-sided printing for an existing form design . . . 68
To prepare a form design to use with preprinted paper 69
To use master pages in a form intended for double-sided printing 71
To specify the master page placement in a page set 71
Inserting a blank page after the first page in a form intended for double-sided printing 73
To insert a blank page after the first page in a form intended for double-sided printing 74
Using page sets to control single-sided and double-sided printing in a form 74
To print the contents of the Design View or Master Pages tab . .76
To print a form with sample data 76
To create sample data to print with your form 77
To automatically generate sample data to print with your form .77
To specify settings for printing a PDF form 78
To automatically print a PDF form when it is opened 79
Tabbing order 79
How the default tabbing order works 79
Viewing the default tabbing order 80
To hide the tabbing order and return to normal form-editing mode 81
Using the Tab Order palette 81
Changing the tabbing order 82
To change the tabbing order for a single object using the mouse 83
To change the order using drag-and-drop 84
To change the order using the arrow buttons 84
To change the order using the menu 84
To change the order by editing the number 84
To change the order using copy and paste 84
To change the tabbing order for a single object using keyboard shortcuts 84

To change the order using the menu 85
To change the order by editing the number 85
To change the order using copy and paste 85
Changing the tabbing order for a group of objects 85
To select a sequential group of objects using the mouse 85
To select a sequential group of objects using the keyboard 85
To change a group of individually selected objects to a sequential group 85
To change the order to geographic (left-to-right) 86
To change the geographic order to right-to-left 86
Changing the tabbing order on master pages 86
To change the tabbing order in fragments 86
To return to the default tabbing order 87
Excluding text and image objects from the tabbing order 87
To exclude text and image objects 87
To view the tabbing order using visual aids 88
Building actions in forms 88
Combining conditions 89
Build an action 90
Rename an action 91
Revert to the default name 91
Edit an action 91
Remove an action 91
Build a custom result 91
Setting the zoom level in a PDF form 92
Spell checking in forms 92
To select the default locale 93
To check spelling while you type 93
To set up Designer to identify spelling errors as you type 94
To check the spelling in a form 94
To add a word to all languages in the custom dictionary 95
To add a word to selected languages in the custom dictionary .96
To edit the custom dictionary 96
To open the Edit Dictionary dialog box 96
Adding a word to the list of ignored words 96
To add a word to the list of ignored words while typing 97
To add a word to the list of ignored words while spell checking a form 97
To add a dictionary 97
To remove a dictionary 98
Hyphenate text 98
Considerations for setting hyphenation 99
Hyphenate option in the Paragraph palette 100
Currently Editing palette menu 100
Expand to Fit options 100
Allow Multiple Lines option 100
Setting and removing hyphenation in forms 101

To set hyphenation for a new form 101
To set hyphenation for an individual form 102
To remove hyphenation from a form 102
Hyperlinks 103
To insert a hyperlink 103
To remove hyperlink 104
To edit a hyperlink 104
To test a hyperlink 104
Prepare for translation 104
To create translation identifiers 105
Extracting the translation identifiers and translatable text . . . 105
extractstrings.xslt optional parameters 105
Creating the form design with the translated text 106
Macros 106
To set up macros for use in Designer 107
About macro.xml configuration files 107
Organizing the macros subfolder 107
Localizing macros 109
Macro Logging 109

Chapter: 4 Test and troubleshoot 110

Previewing and testing forms 110
To preview and test forms in the Preview HTML tab 110
To preview and test forms in the Preview PDF tab 111
To set preview options for an interactive form 111
To set preview options for a non-interactive form 112
To preview a form using sample data 112
To automatically generate sample data to preview your form . 113
To create sample data to preview your form 113
Considerations for testing form designs with data 114
Creating a sample data file 115
To test for accessibility 115
Addressing warning messages in the Report palette 116
Target version warning messages 117
Action warning messages 124
Warning marker messages 124
Scripting error messages 127
JavaScript Scripting Errors 127
FormCalc Scripting Errors 133
Displaying validation errors in Acrobat 134
Recommendations for using form validation with static PDF forms 135
To configure how Acrobat displays validation error message boxes 135
To color fields that fail validation 136
To color mandatory fields that are not filled 136
To set the focus to the first field that fails validation 136
Guidelines for forms 137

Rendering PDF forms 137
Characteristics of static PDF forms 137
Characteristics of dynamic PDF forms 138
Choosing the type of PDF form 139
Applying formatting by using client-side scripts 139
Changing the field border color 139
Applying field shading 140
Hiding objects 140
Saving object formatting 141
Unavailable commands for dynamic forms in Adobe Reader . 142
Font behavior 142
Fonts can be subset in static PDF forms 142
Fonts are embedded in dynamic PDF forms 142
Objects and properties for static PDF forms 142
Import documents 143
Importing PDF files 144
To import a PDF file by using the New Form Assistant in the stand-alone version of Designer 145
To import a PDF file without the New Form Assistant in the stand-alone version of Designer 145
To use File > New to import a PDF file with Designer and Workbench 146
To use File > Open to import a PDF file with Designer and Workbench 146
Reviewing the results of a PDF file import 147
How Designer converts PDF objects 148
About reducing PDF conversion problems 150
Importing tagged PDF files 151
Matching unavailable fonts 151
To make the font substitution permanent 152
Adding a new mapping to the font-mapping table in Designer 152
Syntax 152
Parameters 152
Example 153
Font-mapping table in ConvertPDF_FontMap.txt 153
Syntax 153
Example 154
Character-mapping table in ConvertPDF_CharMap.txt 154
Syntax 154
Example 155
Importing PDF documents as artwork 155
Keep in mind the following factors when importing a PDF document as artwork: 155
To import a PDF file as artwork in the stand-alone version of Designer 156
To use File > New to import a PDF file as artwork with Designer and Workbench 156
To use File > Open to import a PDF file as artwork with Designer and Workbench 156

To replace PDF artwork 157
About importing structured PDF documents as artwork 157
Importing Word files 159
How Designer converts Word objects and settings 160
About reducing Word conversion problems 162
Copying spreadsheet data from Microsoft Excel 162
Creating a form from the content of a Microsoft Excel spreadsheet 162
Importing InfoPath files 163
How Designer converts Microsoft InfoPath objects 164
Differences between Designer and InfoPath objects 165
Importing Adobe Output Designer Form files 165
Importing XForms Model files 166

Chapter: 5 Guidelines for forms 168

Rendering PDF forms 168
Characteristics of static PDF forms 168
Characteristics of dynamic PDF forms 169
Choosing the type of PDF form 169
Applying formatting by using client-side scripts 170
Changing the field border color 170
Applying field shading 170
Hiding objects 171
Saving object formatting 171
Unavailable commands for dynamic forms in Adobe Reader 172
Font behavior 173
Fonts can be subset in static PDF forms 173
Fonts are embedded in dynamic PDF forms 173
Objects and properties for static PDF forms 173

Chapter: 6 Import documents 175

Importing PDF files 175
To import a PDF file by using the New Form Assistant in the stand-alone version of Designer 176
To import a PDF file without the New Form Assistant in the stand-alone version of Designer 177
To use File > New to import a PDF file with Designer and Workbench 177
To use File > Open to import a PDF file with Designer and Workbench 178
Reviewing the results of a PDF file import 178
How Designer converts PDF objects 179
About reducing PDF conversion problems 182
Importing tagged PDF files 183
Matching unavailable fonts 183
To make the font substitution permanent 183
Adding a new mapping to the font-mapping table in Designer 184
Syntax 184

Parameters . . . 184
Example . . . 184
Font-mapping table in ConvertPDF_FontMap.txt . . . 185
Syntax . . . 185
Example . . . 185
Character-mapping table in ConvertPDF_CharMap.txt . . . 186
Syntax . . . 186
Example . . . 186
Importing PDF documents as artwork . . . 187
Keep in mind the following factors when importing a PDF document as artwork: . . . 187
To import a PDF file as artwork in the stand-alone version of Designer 188
To use File > New to import a PDF file as artwork with Designer and Workbench . . . 188
To use File > Open to import a PDF file as artwork with Designer and Workbench . . . 188
To replace PDF artwork . . . 189
About importing structured PDF documents as artwork . . . 189
Replacing PDF artwork affects document structure . . . 190
To add a field tag to the document structure . . . 190
To remove a field tag from the document structure . . . 191
To move a field tag up or down in the document structure . . . 191
To find untagged fields in the document structure . . . 191
To show the context tags in the document structure . . . 191
Importing Word files . . . 191
How Designer converts Word objects and settings . . . 192
About reducing Word conversion problems . . . 193
Copying spreadsheet data from Microsoft Excel . . . 194
Creating a form from the content of a Microsoft Excel spreadsheet . . . 194
Importing InfoPath files . . . 195
How Designer converts Microsoft InfoPath objects . . . 196
Differences between Designer and InfoPath objects . . . 197
Importing Adobe Output Designer Form files . . . 197
Importing XForms Model files . . . 198

Chapter: 7 Using tables . . . 200

About tables . . . 200
Types of tables . . . 201
Ways to create tables . . . 201
To create a simple table . . . 202
To create a table using the Table Assistant . . . 202
To create a table from existing objects . . . 204
To create a table within a table . . . 205
To import a table from Microsoft Word . . . 205
To create a table using subforms . . . 206
To create a table containing variable-width cells . . . 207

To create a nested table using subforms 207
To create a table that grows by using the Subform Instance Controls . 210
To create a table that grows by using the Subform Instance Controls 210
To add the Subform Instance Control buttons 210
To create a table that grows using the Button object 212
To create a table that grows using the Button object 212
To add the buttons 212
Selecting, copying, moving, and navigating 214
To select a table, row, column, cell, or section 214
To select the entire table 214
To select the cells in a row 214
To select a row 215
To select the cells in a column 215
To select a cell 215
To select multiple rows, columns, cells, or sections 215
To select a section 216
To copy a table, row, column, cell, or section 216
To move a table 216
To go to a specific row 216
Inserting and deleting 217
To insert a row, column, or section 217
To delete a table, row, column, or section 217
To delete a table 217
To delete a row, column, or section 217
Formatting a table 217
To add a table title 217
Adding space 217
To add space around a table 217
To add space around cells 218
To add space around rows 218
Applying borders and shading to a table 219
To apply and remove borders and shading 219
To remove the border from around the entire table 219
To distribute rows and columns evenly 219
To format the corners of a table or cells 220
To show captions in rows 221
To create a table style for the custom library 222
To create a calendar using a table 223
Aligning, resizing, and arranging tables 224
To align a table with other objects 224
To align the contents of a cell 224
To make another object the same size as a table 225
To resize an entire table 225
To resize rows and columns 225
To center a table 225
To arrange a table with other objects 225

Working with data in tables 226
To create a table from a data source 226
To set the rows in a table to adjust to accommodate data . . . 227
To create a table that groups data 228
To create the main table 230
To create the first nested table 231
To create the last nested table inside the first nested table . . 231
To show data in the cells 231
To make a table optional 232
To perform calculations in a table 233
To use the TableCalcs custom object 234
Working with pagination in tables 235
To set up a table to span multiple pages 235
To set up a master page that includes the Table continued tag 236
To set the pagination for the table 237
To connect to a data source 237
To select a data file 238
To repeat a header or footer row on subsequent pages 238
Working with header and footer rows in tables 238
To change a row to a header, body, or footer row 238
To control table, header row, body row, footer row, and section breaks using conditional statements 238
Working with cells and cell contents 240
To clear a cell 240
To merge and split cells 240
To merge cells into one cell 240
To split a merged cell 241
To orient text vertically in cells 241
To change a cell to another object type 241
To change a cell to another object type by using the Type list in the Cell tab of the Object palette 242
To change a cell to another object type by dragging an object from the Object Library palette 243
To change a cell to another object type by dragging an object from the form design 243
Working with table sections 243
To insert a table section 243
To group rows into a section 244
To ungroup a section into rows 244
To make a table section optional 244
Creating choice sections in tables 244
To create a choice section from an existing section 245
To create a choice section from a row 245
To add a header, body, or footer row to a choice section . . . 245
To remove a header, body, or footer row from a choice section 245
Creating data bindings and conditional statements for choice table sections 246

To create data bindings and conditional statements for choice table sections 246

Chapter: 8 Using subforms 248

About subforms 248
Subforms that position content 250
Subforms that flow content 251
Creating and configuring subforms 251
To add a subform 251
To add a subform from the menu 252
To add a subform from the Library palette 252
To name a subform 252
To name or rename a subform using the Hierarchy palette . . . 252
To name or rename a subform using the Binding tab of the Object palette 252
To make a subform visible, invisible, or hidden 252
To specify a locale (language and country or region) for a subform 253
To wrap and unwrap objects in a subform 253
To add objects to an existing subform 253
To unwrap the objects in a subform 253
To specify how a subform manages content 253
To specify how a subform positions content 254
To specify a flow direction for the objects in a subform that flows content 254
To change a subform that positions content into a subform that flows content 254
To position subforms 254
To specify where to position a subform 254
To keep the subform within the same content area or page as the previous or next subform 255
To specify how to merge data between subforms 255
To define custom data-binding properties for a subform 256
Using subform sets 256
To insert a subform set 257
Creating repeating and multipage subforms 257
To create a repeating subform 258
To specify a subform to span multiple pages 258
To enable a subform to span multiple pages 259
About overflow leaders and trailers 259
To create and assign overflow leaders and trailers 260
To create an overflow leader or overflow trailer subform 260
To apply an overflow leader or overflow trailer to a subform . 261
To control subform and subform set breaks by using conditional statements 261
Using choice subform sets 262
To create choice subform sets 262
To create a choice subform set from an existing subform set . . 263

To create a choice subform set from a subform object 263
To add and remove subforms in choice subform sets 263
To add a subform to a choice subform set 263
To remove a subform from a choice subform set 263
Working with forms that have a flowable layout 264
Differences at design time and run time 264
Controlling the placement of objects in forms that have a flowable layout 264
Content areas in a flowable layout 265
Subforms in a flowable layout 265
Creating form designs that have a flowable layout 266
Start with a fixed layout 266
Tips for starting with a fixed layout 267
Creating interactive forms that have a flowable layout 268
Position the header and footer on the master page 269
Wrap form objects in subforms and setting occurrence values 269
Modify margins to accommodate subforms 270
Use overflow leader and overflow trailer subforms 270
Create subforms with flowable elements 270
Create a button to add and remove a section 271
Create add and delete buttons 272
Add and remove subform instances using scripting 273
To add a subform instance to a form 274
Add and remove tables and table rows by using scripting . . . 275
To add a table or table row instance to a form 275
FormCalc 275
To remove a table or table row instance from your form 276
FormCalc 276
Creating non-interactive forms that have a flowable layout . . 277
How non-interactive forms that have a flowable layout work . 277
Design tips for forms that have flowable layout 278
General 278
Subforms 279
Accessibility 280
Scripting 280
Data binding 280
Common mistakes 280

Chapter: 9 Using fragments 282

About fragments 282
Fragments and subforms 282
Script fragments 283
When to use fragments 283
Content reuse 283
Global updates 283
Shared form creation 284

Security 284
Naming and organizing and fragments 284
Fragment references 285
Fragment references in the Layout Editor 285
Fragment references in the Hierarchy palette 285
How fragment references are resolved 286
Binding fragments to a data source 287
Creating and inserting fragments 287
Create a fragment 288
Inserting fragment references 288
To insert a fragment from the Fragment Library palette 289
To insert a fragment from the Insert menu 289
Editing and embedding fragments 289
To edit a fragment from the Fragment Library palette 290
To edit a fragment from a fragment reference 290
To view fragment information 290
To convert a fragment reference to an embedded object . . . 290
Viewing and embedding fragments 291
To view fragment information 291
To convert a fragment reference to an embedded object . . . 291
Fixing overrides and broken fragment references 291
To remove fragment reference overrides 292
To resolve broken fragment references 292
Fragments tips 292

Chapter: 10 Working with Objects 294

About Objects 294
Categories of objects 294
To change an object type 297
To change an object type using the Type list in the Object palette tabs 298
To change an object type using the Type list in the Object Editor 298
To name and rename objects 298
To name or rename an object by using the Hierarchy palette . 298
To name or rename an object by using the Binding tab of the Object palette 298
To name or rename an object by using the Object Editor 299
To set the border style 299
To specify a locale (language and country or region) for an object 299
Objects that support scripting and calculations 299
Using objects in interactive forms 300
Using objects 300
Using barcodes 300
Barcode formats 301
Barcode position and appearance 301
Barcode properties 301

To define barcode field properties 302
To set the default value for the barcode 302
To control how a barcode obtains data 302
To define custom data-binding properties for a barcode 303
Encoding non-printing characters in barcode data 303
Using buttons 304
Email submit buttons 305
HTTP submit buttons 306
Print buttons 307
Reset buttons 307
To insert a standard button that runs a calculation or script . . 307
Processing options for a button 308
To execute a web service operation or database query using a button 308
To attach a database script to a button 309
Using a button to populate fields from a database 310
Submitting data using a button 311
To add a button that submits an XML Data Package (XDP) to a URL 312
To add a button that submits an embedded PDF to a URL 313
To add an email submit button that submits an embedded PDF to an email address 314
To add a button that submits XML data to a URL 314
To add an email submit button that submits XML data to an email address 315
To add a button that submits XML data to an e-mail address . . 316
To add a button that submits a text stream using the POST method 316
To add an email submit button that submits encrypted form content to an e-email 317
To add a button that submits encrypted and encoded form content to an e-mail address 317
To add an HTTP button that submits encrypted form content to a URL 318
Using check boxes 318
To specify the size for the check box 319
To specify the check style 319
To specify the states for the check box 319
To define the behavior of the check box 319
To specify the default state of the check box 320
To change the values assigned to check box states 320
To define custom data-binding properties for a check box 320
Using circles, lines, and rectangles 321
To change the line style 321
To change the thickness of the line 321
To select a line color 321
To specify the shape of the circle 322
To specify the start and end points of an arc 322
To fill the circle or rectangle with color or a pattern 322
To change the slope of the line 322

To change the corners of the rectangle 322
Using content areas 322
Data flow between content areas 323
Copying, moving, and resizing content areas 324
Adding content areas 325
Using date/time fields 325
To add current date fields 326
To define the behavior of date/time fields 327
To define custom data-binding properties for date/time fields 327
To specify the data to display in date/time fields 327
To define the pattern for displaying date/time fields 328
To specify the comb format for date/time fields 328
To limit the numbers in date/time fields 328
Using decimal and numeric fields 328
About numeric fields 329
Value ranges of XSD simple types mapped to numeric fields. . . 330
To define a radix (decimal point) alignment for numeric values 330
To define the behavior of decimal or numeric fields 331
To define custom data-binding properties for decimal or numeric fields 331
To specify the data format for numeric fields 331
To limit the numbers entered in decimal and numeric fields . . 332
To perform a calculation with decimal fields by using JavaScript 332
To specify the comb format for decimal and numeric fields . . . 332
Using extended decimal fields 333
Using drop-down lists and list boxes 334
To specify the list of options for a drop-down list or list box . . 335
To add items from Excel or text editor 335
To add items individually 336
To define the behavior of a drop-down list or list box 336
To display a default selection in the drop-down list or list box . 337
To define custom data-binding properties for a drop-down list or list box 337
To specify list item values for a drop-down list or list box 337
Using flash fields 338
To specify the URL of a SWF file 338
To embed flash data in the form 338
To specify the location of a poster image 339
To embed a poster image 339
To display rich media content in a floating window 339
To add an additional asset 339
To unembed or delete an additional asset 339
To define custom data-binding properties for a flash field 339
Using image fields 340
To specify an initial (default) image to link 343
To embed image data in the form when the form is created . . 343

To size an image 343
To define custom data-binding properties for an image field . . 343
Using images 344
To specify the image to link 345
To embed an image into a form 345
To size the image 345
About paper forms barcodes 345
How the paper forms barcode works 346
Using paper forms barcodes 347
To specify a unique caption for the paper forms barcode 347
To generate a caption for the paper forms barcode 347
To specify the symbology for the paper forms barcode 347
To specify the paper forms barcode scanning method 347
To specify custom decoding settings for a paper forms barcode 348
To control how a paper forms barcode encodes data 348
To create a custom script for encoding data for a paper forms barcode 349
To use legacy encoding format 350
To specify a delimiter 350
To specify character encoding 351
Design tips for using a paper forms barcode 351
Using password fields 353
To specify the password display character 354
To define the behavior of the field 354
To define custom data-binding properties for a password field 355
To limit the characters entered in password fields 355
Using radio buttons 355
To set the size for a radio button 356
To specify the radio button style 356
To assign On values to radio buttons 357
To define custom data-binding properties for radio buttons . . 357
To create a new exclusion group 357
To move a radio button to a different exclusion group 357
To move an exclusion group 358
To merge exclusion groups 358
To define the behavior of the exclusion group 358
To specify the default selection for the exclusion group 358
Using signature fields 359
Using text 359
To type characters into a text object 360
To insert a floating text field into a text object 360
To insert the value of a run-time property into a text object . . 361
To merge text objects 361
To create a caption by merging a text object and a field object 361
To insert language-specific numbers in text or captions 362
To allow page breaks within a text object 363

To keep a text object with the next object in the form 364
Using text fields 364
To allow multiple lines of text in text fields 365
To limit the number of characters in text fields 365
To specify the number of characters in text fields 366
To limit the number of characters to the width of text fields . . 366
Allowing page breaks within a text field 366
To allow page breaks within a text field 366
To keep a text field with the next object in the form 366
To enable the field to be filled with rich-formatted text 366
To define the behavior of the field 366
To define custom data-binding properties for text fields 367
To specify the data format for text fields 367
To specify the comb format for text fields 367
Using form object collections 368
To create a form object collection 368
To create a form object collection 368
To copy a form object collection 369
To copy a form object collection 369
To rename a form object collection 369
To rename a form object collection 369
To modify a form object collection 370
To modify a form object collection 370
To delete a form object collection 370
To delete a form object collection 370
Using custom objects 371
About custom objects for the SAP environment 373
To create a custom object 376
Adding, copying, and deleting objects 377
To add objects to a form design 377
To add an object using the Insert menu 377
To add an object from the Object Library palette 377
To add several objects of the same type 377
To cut, copy, and paste objects 378
To cut or copy and paste an object 378
To copy a group of objects 378
To make multiple copies of one or more objects 378
To duplicate objects 378
To duplicate an object by dragging it 379
To delete objects 379
Selecting, grouping, and moving objects 379
To select objects 379
To select objects using the Hierarchy palette 379
To group and ungroup objects 380
To group interactive fields and set their properties 380
To move objects 381

To move or duplicate objects by pasting . . . 381
To move an object by dragging it . . . 381
To move an object by using the arrow keys . . . 382
To move objects by using x and y coordinates in the Layout palette 382
To stack objects . . . 382
To lock objects . . . 382
To manipulate objects in a form design by selecting them in the Hierarchy palette . . . 383
Aligning and sizing objects . . . 383
To align objects to each other . . . 383
To align the edges of objects . . . 384
To align the vertical centers of objects . . . 384
To align the horizontal centers of objects . . . 384
To align objects in subforms that flow content . . . 384
To align objects to a grid . . . 384
To center objects on the page . . . 385
To rotate objects . . . 385
To distribute objects . . . 385
To resize objects . . . 387
To make objects the same size . . . 387
To make objects expand to fit . . . 387
To position objects . . . 388
Using Snap to Grid . . . 388
Using to Snap to Guideline . . . 389
Using Snap to Object . . . 389
Using Snap to the Center of the Page . . . 390
Formatting . . . 390
Formatting text . . . 390
To apply a different font . . . 391
To change the font size . . . 391
To apply bold, italic, underline, or strikethrough formatting . . 391
To change the color of text . . . 392
To change the baseline shift of text . . . 392
To change the letter spacing of text . . . 393
To change the vertical or horizontal scale of text . . . 393
To auto kern text . . . 394
Formatting paragraphs . . . 394
To align paragraphs . . . 394
To create a bulleted list . . . 395
To create a numbered list . . . 395
To add compound tags to a numbered list . . . 395
To change the indentation of paragraphs . . . 396
To change the spacing of paragraphs . . . 396
Hyphenation in selected paragraphs . . . 397
To hyphenate the text in a selected paragraph . . . 397
To remove hyphenation from a selected paragraph . . . 398

Formatting objects 398
To add space around an object 398
To add a border around an object 398
To add a border around the fillable area 398
To change the default formatting for new objects 399
Using leaders in text 399
To add a leader 400
To edit a leader 400
To delete a leader 400
To enable a visual clue for No Leader 401
To delete tab stops 401
Creating an insertion point 401
Insert an insertion point object into a form 402
Insertion point into an existing subform 402
Define a subform as an insertion point 403
Add an insertion point placeholder (to a subform that is defined as an insertion point) 403
Formatting captions 403
To set a default font for captions in new forms 404
To set a default font for captions in an existing form 404
To rename a caption 404
To position a caption 404
To specify the amount of space for a caption 404
To set the caption area to auto-fit 405
To hide a caption 405
Making objects visible, invisible, or hidden 405
To make an object visible, invisible, or hidden 407
Formatting field values and using patterns 407
When to use patterns 408
Capturing and displaying user input 408
Retrieving and displaying bound data 409
Defaults for value formatting 409
To specify a default value 410
To specify a display pattern 411
To prompt users to enter data 411
Recommending that users enter data 411
Requiring that users enter data 411
To specify an edit pattern 412
To validate user input 412
To define a validation pattern and custom message 413
To display a message when an attached script detects unacceptable input 413
To specify a data pattern 414
Simple patterns 414
Locales 414
Epoch 419

Date formats 419
Time formats 420
Date and time patterns 421
Reserved symbols 424
Complex field patterns 425
Locale-specific patterns 425
Variable patterns 425
To set a default font for values in new forms 425
To set a default font for values in an existing form 425
Setting up an object for other languages 426
To set up an object for Arabic and Hebrew 426
To set up an object for Thai and Vietnamese 427
To set up a date field for other languages 428
Using special objects 429
To define a global field 429
To insert a floating field 430
To insert run-time properties 431
To add a run-time property to a text object 432
To add a run-time property to a text field or numeric field object 432
To add variables to a form 432
Adding repeating fields in a form design 432
About repeating text fields 433
To add repeating fields 433
Adding URL links to a PDF form 434
About URL links 434
Button 434
Hidden text field 434
To hard-code a URL link 435
To pass a URL from data 435
To bind the hidden field explicitly 435
To bind the hidden field implicitly 435
Setting a PDF form action upon submission of data 436
About setting a PDF form action 436
To set a PDF form to close after data submission 436
To set PDF form fields to read-only 437

Chapter: 11 Object Properties 438

Layout properties in the Layout palette 438
Border properties in the Border palette 439
Font properties in the Font palette 439
Paragraph properties in the Paragraph palette 440
Accessibility properties in the Accessibility palette 443
Master page properties in the Master Page tab 445
Presence 446
Master page properties in the Pagination tab 446
Page set properties in the Page Set tab 448

Presence 448

Chapter: 12 Properties in the Object palette 450

Barcode 450
Barcode properties in the Field tab 450
Location 450
Property/Value 450
Presence 450
Barcode properties in the Value tab 451
Type 451
Default 452
Empty Message 452
Validation Pattern 452
Validation Pattern Message and Error 452
Validation Script Message and Error 453
Form Level Validation Settings 453
Override Message 453
Barcode properties in the Binding tab 453
Name 453
Data Binding 453
Import/Export Bindings 454
Supported barcode formats 454
Valid barcode text characters 458
Button 459
Button properties in the Field tab 459
Type 459
Caption 459
Appearance 460
Highlighting 460
Rollover Caption 460
Down Caption 460
Control Type 461
Presence 461
Locale 462
Print button properties in the Field tab 462
Type 462
Caption 462
Appearance 462
Highlighting 463
Rollover Caption 463
Down Caption 463
Presence 463
Reset button properties in the Field tab 464
Caption 464
Appearance 465
Highlighting 465

Rollover Caption 465
Down Caption 465
Presence 466
Locale 466
Button properties in the Submit tab 467
Submit to URL 467
Submit As 467
Include 468
Data Encoding 468
Button properties in the Execute tab 469
Connection 469
Connection Info 469
Run At 469
Re-merge Form Data 469
Check box 470
Check box properties in the Field tab 470
Caption 470
Appearance 470
States 471
Size 471
Check Style 471
Presence 472
Locale 472
Check box properties in the Value tab 473
Type 473
Default 473
Validation Script Message and Error 474
Form Level Validation Settings 474
Override Message 474
Check box properties in the Binding tab 474
Name 474
Data Binding 474
On Value 475
Off Value 475
Neutral Value 475
Import/Export Bindings 475
Circle 475
Circle properties in the Draw tab 475
Appearance 475
Start 476
Sweep 476
Line Style 476
Color Picker 476
Fill 476
Color Picker 476
Presence 476

Content area 477
Content area properties in the Content Area tab 477
Name 477
Flow Direction 477
Presence 478
Date/time field 478
Date/time field properties in the Field tab 479
Caption 479
Appearance 479
Limit Length to Visible Area 479
Use Cells and Cells 479
Patterns 480
Presence 480
Locale 480
Date/time field properties in the Value tab 481
Type 481
Default 482
Empty Message 482
Validation Pattern 482
Validation Pattern Message and Error 482
Validation Script Message and Error 483
Form Level Validation Settings 483
Calculation Script 483
Runtime Property 483
Override Message 483
Date/time field properties in the Binding tab 483
Name 483
Data Binding 484
Data Format 484
Import/Export Bindings 484
Examples of date/time patterns 485
Predefined patterns 485
Custom patterns 485
Decimal field 486
Decimal Field properties in the Field tab 486
Type 486
Caption 486
Appearance 486
Limit Length to Visible Area 487
Comb of <x> characters 487
Patterns 487
Limit Leading Digits and Max 487
Limit Trailing Digits and Max 487
Presence 487
Locale 488
Drop-down list 489

Drop-down list properties in the Field tab 489
Type 489
Caption 489
Appearance 489
List Items 490
Allow Custom Text Entry 490
Commit On 490
Presence 490
Locale 491
Drop-down list properties in the Value tab 491
Type 491
Default 492
Empty Message 492
Validation Pattern 492
Validation Pattern Message and Error 493
Validation Script Message and Error 493
Form Level Validation Settings 493
Override Message 493
Drop-down list properties in the Binding tab 493
Name 493
Data Binding 493
Specify Item Values 494
Up, Down, Sort Ascending, Sort Descending 494
Import/Export Bindings 494
Email submit button 494
Email submit button properties in the Field tab 495
Type 495
Caption 495
Appearance 495
Highlighting 495
Rollover Caption 496
Down Caption 496
Email Address 496
Email Subject 496
Submit As 496
Presence 497
Locale 498
Extended Decimal Field 498
Extended Decimal Field properties in the Field tab 498
Type 498
Caption 498
Appearance 498
Limit Length to Visible Area 499
Comb of <x> characters 499
Patterns 499
Limit Leading Digits and Max 499

Limit Trailing Digits and Max 499
Presence 500
Locale 500
(Deprecated) Flash field 501
(Deprecated)Flash field properties in the Field tab 501
Property/Value 501
Property/Value 502
Presence 503
Locale 504
Flash field properties in the Binding tab 504
Name 504
Data Binding 504
HTTP submit button 505
HTTP submit button properties in the Field tab 505
Type 505
Caption 505
Appearance 505
Highlighting 506
Rollover Caption 506
Down Caption 506
URL 506
Presence 507
Locale 508
Image field 508
Image field properties in the Field tab 508
Type 508
Caption 508
URL 508
Embed Image Data 508
Presence 509
Locale 510
Image field properties in the Binding tab 510
Name 510
Data Binding 510
Import/Export Bindings 511
Image 511
Image properties in the Draw tab 511
Type 511
URL 511
Embed Image Data 511
Sizing 511
Presence 512
Line 512
Line properties in the Draw tab 513
Type 513
Appearance 513

Line Style 513
Color Picker 513
Presence 513
List box 514
List box properties in the Field tab 514
Type 514
Caption 514
Appearance 514
List Items 515
Allow Multiple Selection 515
Commit On 515
Presence 515
Locale 516
List box properties in the Value tab 516
Type 516
Default 517
Empty Message 517
Validation Pattern 517
Validation Pattern Message and Error 518
Validation Script Message and Error 518
Form Level Validation Settings 518
Override Message 518
List box properties in the Binding tab 518
Name 518
Data Binding 518
Specify Item Values 519
Up, Down, Sort Ascending, Sort Descending 519
Import/Export Bindings 519
Numeric field 519
Numeric field properties in the Field tab 520
Type 520
Caption 520
Appearance 520
Limit Length to Visible Area 520
Comb of <x> characters 520
Patterns 521
Presence 521
Locale 521
Numeric field properties in the Value tab 522
Type 522
Default 523
Empty Message 523
Validation Pattern 523
Validation Pattern Message and Error 523
Validation Script Message and Error 524
Form Level Validation Settings 524

Calculation Script 524
Runtime Property 524
Override Message 524
Numeric field properties in the Binding tab 525
Name 525
Data Binding 525
Data Format 525
Import/Export Bindings 526
Numeric patterns 526
Reserved symbols 528
Examples of numeric patterns 528
Paper Forms Barcode 530
Paper forms barcode properties in the Field tab 530
Type 530
Label 530
Generate Label Automatically 530
Symbology 530
Scanning Method 531
Property/Value 532
Presence 532
Paper forms barcode properties in the Value tab 533
Compress Data Before Encoding 533
Automatic Scripting 533
Format 534
Include Field Names 534
Include Label 534
Use Legacy Format 534
Delimiter 534
Character Encoding 534
Apply To 536
Collection 536
Modify 536
Password field 536
Password field properties in the Field tab 536
Type 537
Caption 537
Appearance 537
Password Display Character 537
Limit Length to Visible Area 537
Patterns 537
Presence 538
Locale 538
Password field properties in the Value tab 539
Type 539
Empty Message 539
Validation Pattern 539

Validation Pattern Message and Error 540
Validation Script Message and Error 540
Form Level Validation Settings 540
Password field properties in the Binding tab 540
Name 540
Data Binding 540
Import/Export Bindings 541
Password patterns 541
Reserved symbols 541
Examples of password patterns 542
Radio button 542
Radio button properties in the Field tab 542
Type 542
Appearance 543
Size 543
Check Style 543
Item 544
Presence 544
Locale 545
Radio button properties in the Value tab 545
Type 545
Default 546
Empty Message 546
Validation Script Message and Error 547
Form Level Validation Settings 547
Override Message 547
Radio button properties in the Binding tab 547
Name 547
Data Binding 547
Specify Item Values 548
Import/Export Bindings 548
Rectangle 548
Rectangle properties in the Draw tab 548
Type 548
Line Style 548
Color Picker 548
Corners 549
Radius 549
Fill 549
Color Picker 549
Presence 549
Signature field 550
Signature field properties in the Field tab 550
Type 550
Caption 550
Appearance 550

Presence 551
Locale 551
Signature field properties in the Signature tab 552
Lock Fields After Signing 552
All Fields in Collection 552
All Fields not in Collection 553
Subform 553
Subform properties in the Subform tab 553
Source File 553
Fragment Name 553
Content 553
Flow Direction 553
Allow Page Breaks Within Content 554
This Subform Is An Insertion Point 554
Name 554
Presence 554
Locale 555
Subform properties in the Pagination tab 555
Place 555
Keep With Previous 556
Keep With Next 556
After 556
Conditional Breaks 557
Overflow 557
Overflow Leader 557
Overflow Trailer 558
Subform properties in the Binding tab 558
Name 558
Data Binding 558
Import/Export Bindings (Execute WSDL) 559
Initial Count 559
Subform set 559
Subform set properties in the Subform Set tab 559
Type 559
Presence 559
Subform set properties in the Pagination tab 560
Place 560
After 561
Conditional Breaks 562
Overflow 562
Overflow Leader 562
Overflow Trailer 562
Subform set properties in the Binding tab 563
Name 563
Repeat Subform For Each Data Item 563
Min Count 563

Max 563
Import/Export Bindings (Execute WSDL) 563
Table 563
Table properties in the Table tab 563
Source File 564
Fragment Name 564
Body Rows 564
Header Rows 564
Columns 564
Footer Rows 564
Allow Page Breaks Within Content 564
Presence 564
Locale 565
Table properties in the Row Shading tab 565
Apply Alternating Row Shading 566
Color Picker 566
Next 566
Color Picker 566
Table properties in the Pagination tab 566
Place 566
Keep With Previous 567
Keep With Next 567
After 567
Conditional Breaks 568
Overflow 568
Table properties in the Binding tab 569
Name 569
Data Binding 569
Import/Export Bindings 569
Repeat Table For Each Data Item 569
Min Count 569
Max 569
Initial Count 570
Header row properties in the Row tab 570
Source File 570
Fragment Name 570
Type 570
Layout 571
Allow Page Breaks Within Content 571
Presence 571
Locale 572
Footer row properties in the Row tab 572
Source File 572
Fragment Name 572
Type 572
Layout 573

Allow Page Breaks Within Content 573
Presence 573
Locale 574
Header and footer row properties in the Pagination tab 574
Place 574
Keep With Previous 575
Keep With Next 575
After 575
Conditional Breaks 576
Include Header Row At Beginning Of Table 576
Include Header Row In Subsequent Pages 576
Include Footer Row In Subsequent Pages 576
Include Footer Row At End Of Table 576
Header row properties in the Binding tab 577
Name 577
Data Binding 577
Import/Export Bindings 577
Repeat Row For Each Data Item, Min Count, Max, and Initial Count 577
Footer row properties in the Binding tab 578
Name 578
Data Binding 578
Import/Export Bindings 578
Repeat Row For Each Data Item, Min Count, Max, and Initial Count 578
Body row properties in the Row tab 579
Source File 579
Fragment Name 579
Type 579
Layout 579
Allow Page Breaks Within Content 580
Presence 580
Locale 580
Body row properties in the Pagination tab 581
Place 581
Keep With Previous 582
Keep With Next 582
After 582
Conditional Breaks 583
Body row properties in the Binding tab 583
Name 583
Data Binding 583
Import/Export Bindings 583
Repeat Row For Each Data Item, Min Count, Max, and Initial Count 584
Section properties in the Section tab 584
Type 584
Section properties in the Row Shading tab 584
Apply Alternating Row Shading 584

First . . . 585
Color Picker . . . 585
Next . . . 585
Color Picker . . . 585
Section properties in the Pagination tab . . . 585
Place . . . 585
After . . . 586
Conditional Breaks . . . 587
Overflow . . . 587
Section properties in the Binding tab . . . 587
Name . . . 587
Repeat Section For Each Data Item, Min Count, Max, and Initial Count 588
Cell properties in the Cell tab . . . 588
Presence . . . 588
Cell properties in the Value tab . . . 589
Cell properties in the Binding tab . . . 589
Text . . . 590
Text properties in the Draw tab . . . 590
Type . . . 590
Allow Page Breaks Within Content . . . 590
Keep With Next . . . 590
Presence . . . 590
Locale . . . 591
Text field . . . 591
Text field properties in the Field tab . . . 592
Type . . . 592
Caption . . . 592
Appearance . . . 592
Allow Multiple Lines . . . 592
Limit Length and Max Chars . . . 592
Limit Length to Visible Area . . . 593
Allow Page Breaks Within Content . . . 593
Keep With Next . . . 593
Comb Of Characters . . . 593
Field Format . . . 593
Patterns . . . 593
Presence . . . 593
Locale . . . 594
Text field properties in the Value tab . . . 594
Type . . . 594
Default . . . 595
Empty Message . . . 595
Validation Pattern . . . 596
Validation Pattern Message and Error . . . 596
Validation Script Message and Error . . . 596
Form Level Validation Settings . . . 596

Calculation Script 596
Runtime Property 596
Override Message 597
Text field properties in the Binding tab 597
Name 597
Data Binding 597
Import/Export Bindings 598
Text field patterns 598
Reserved symbols 599
Examples of text patterns 599
Predefined patterns 599
Custom patterns 599

Chapter: 13 Working with Data Sources 601

About data connections 601
Updating data connections and the Data View palette 603
Updating the Data View palette for non-embedded XML schema data connections 603
To work with the Data View palette 603
Schema filtering 604
To apply a filter to the Data View palette 604
To remove a filter from the Data View palette 604
Connecting to a data source 604
Create a data connection to an XML schema 605
Create a data connection to an XML data file 606
Create a data connection to a model 606
Create a data connection to an OLE database 607
About creating a data connection using a WSDL file 608
Create a data connection using a WSDL file 610
Execute the connection to the web service 611
To execute the connection to the web service using a button . 611
To execute the connection to the web service from another event 611
To view a button's connection information 611
Modifying and deleting a data connection 611
Delete a data connection 611
Modify a data connection 612
Binding fields to a data source 612
Binding to data not defined by the default data connection . . 612
Saving or exporting undefined data in a separate namespace . 613
Identifying nodes not bound by the default data connection . . 613
To allow binding to data that is not defined by the default data connection 613
Ways to create bound fields or subforms 613
To create a bound field or subform 614
To bind referenced data nodes to fields or subforms 614
To bind a node to an existing field 616

To generate bound fields in a subform 616
Create data bindings and conditional statements for choice subform sets 617
Ways to show the same data in multiple places 618
Bind multiple fields to one node 621
Use a script to show the data that is entered into one field in another field 622
Set field generation options 622
Remove a binding 623
Refresh data connections 623
View a list of bound fields 623
Field types 624
Simple types 624
Built-in simple types 625
Restricted simple types 626
List and union types 627
Complex types 627
How a subform is generated from element content in a data source 627
Simple content 628
Mixed content 628
Empty content 628
Occurrence 628
Content models 628
Rich text objects 628
Images 629
Default/fixed values 630
Annotation information 630
Namespaces 630
Validation 630
Numeric type scripts 631
Inclusion/exclusion scripts 631
length, minLength, and maxLength scripts 632
totalDigits/fractionDigits scripts 632
Dynamically populating form object properties from a data source . . 632
Set up dynamic properties 634
To turn active labels on and off 634
To set the default color for active labels 635
Dynamically populate a drop-down list or list box 635
Dynamically populate a drop-down list or list box from an OLEDB data connection 638
Dynamically populate a drop-down list in the Web DynPro environment 638
To customize the default value in the drop-down list 639
Dynamically populate a caption 640
Dynamically populate a validation pattern message 641
Dynamically populate a validation script message 643

Dynamically populate a tool tip 645
Dynamically populate screen reader text 646
Dynamically populate a second field after populating the first 648
To set up the first field 649
To set up the script that dynamically populates a second field after populating the first 650
To set up the second field 651
Dynamically populate form object properties using the drag and drop method 651
Edit the dynamic data binding 651
Remove the dynamic data binding 652

Chapter: 14 Creating Accessible Forms 653

About accessible forms 653
Reading order versus tabbing order in accessible forms 653
Reading order restrictions 653
Reading order and subforms 654
Static text objects and tabbing order 654
Designing accessible forms 654
Tips for designing accessible forms 654
Forms for users with vision impairment or disabilities 656
Forms for users with reduced mobility 657
Color in accessible forms 657
Support for screen readers 658
Design tips for authoring forms for screen readers 658
Create a PDF form with accessibility tags 659
Create a PDF form with accessibility tags 659
Creating accessible PDF forms 659
Design tips for creating accessible PDF forms 659
Techniques for testing form accessibility 660
Making objects accessible 661
Specify custom screen reader text for an object 661
Specify a custom tool tip for an object 662
Specify a different search order for screen reader text 662
Turn off screen reader text for an object 662
Make radio buttons accessible 663
To make radio buttons accessible using the button captions . . 663
To make radio buttons accessible using a specified item value 663
To make Table of Content accessible using Accessible Palette . 664
Make lists accessible 664
Make images accessible 664
Make scripts accessible 665
Making tables accessible 666
Make simple tables accessible 666
Navigate tables in accessible forms 666
Make complex tables accessible 667

Add structural navigation in forms 667
Checking the accessibility of a form 668
To run the Accessibility checker 668
Accessibility checker rules 668

Chapter: 15 Setting Security 671

Importing digital IDs 671
To import a digital ID into the Windows Certificate Store . . . 672
To import a digital ID into the Digital ID Files store 673
Using XML encryption 673
To apply XML encryption to form content using a button . . . 674
To apply XML encryption to form content using an email submit button 675
To apply XML encryption to form content using an HTTP submit button object 675
Using digital signatures 675
Document signatures 676
Data signatures 676
To add a signature field 677
Considerations for adding multiple signature fields 678
To test a signature field 678
To add a data signature to a submit button 678
To use digital signatures in dynamic PDF forms 679
Using password fields 680
To add a password field in an interactive form 680
To change the password display character 681
To specify an input format and validation message 681
Using PDF security options 681
To set a password to open a PDF form 682
To restrict printing in a PDF form 682
To restrict editing in a PDF form 682
To enable text access for screen readers in secure forms . . . 683

Chapter: 16 Digital Signatures in PDF Workflows 684

Design considerations 684
Application design considerations 684
How PDF documents are signed 684
How document signature fields are added 685
How digitally signed PDF documents are submitted for validation . . . 685
Performing multiple operations on a PDF document 686
Using scripts in forms that have digital signatures 686
Obtaining digital certificates 686
Signing PDF documents 686
Signing PDF documents in Adobe document services 686
Signing PDF documents in Acrobat 686

Signing PDF documents in Adobe Reader 687
Validating a signature in Adobe document services 687
Comparing document versions 687
Single digital signature workflows 688
Multiple digital signature workflows 688
Locking fields 688
Signature workflow in Adobe document services 689
Certified documents 689
Legal warnings 689
GeoTrust certificates 690
Adobe Root Certificate Authority 690
Chain of trust 690
Creating and printing RFID labels 691
Before you start 691
Create and print RFID labels 691
Step 1 Add the RFID object to the form design 692
Step 2 Set the data length of the object 692
RFID Tag types 692
Step 3 Bind the object to the XML data 692
Step 4 Configure the XDC file 692
Step 5 Map the XDC file to an SAP output device 693
Step 6 Verify the printed RFID label 693

Chapter: 17 Creating and printing RFID labels 694

Before you start 694
Create and print RFID labels 694
Step 1 Add the RFID object to the form design 695
Step 2 Set the data length of the object 695
RFID Tag types 695
Step 3 Bind the object to the XML data 695
Step 4 Configure the XDC file 695
Step 5 Map the XDC file to an SAP output device 696
Step 6 Verify the printed RFID label 696

Chapter: 18 Creating Forms for Optimal Performance 697

General design considerations for performance 697
Optimizing performance for objects 698
Optimizing performance for images 698
Optimizing performance for discrete option choices 699
Optimizing performance for subforms 699
Optimizing performance for text fields 700
Performance of objects on master pages versus pages 700
Tips for reducing the number of objects 700
Tips for reducing the complexity of objects 701
Structuring data for performance 701

Data binding options for performance 701
Evaluating data structure for performance 702
Choosing fonts for performance 703
Design considerations for printing 704
Performance considerations for importing forms 704
Performance considerations for accessible forms 705
Performance considerations for scripts 705
Performance considerations for XML Form Object Model expressions 706
Resolving log messages for optimal performance 706

Chapter: 19 Considerations for Creating Forms for Server Processing 708

Creating forms for AEM forms 708
How Forms processes captured data 708
How Forms renders forms 709
How Forms prepopulates forms 709
Designing forms for Forms 709
Considerations for creating form designs for Forms 710
Additional requirements for submitting form designs to Forms 711
Specifying the format for submitting data 712
Form transformations 712
Copying form designs to the local network 713
Saving a form design: XDP or PDF 713
Optimizing and improving performance for Forms 713
Form caching 714
To specify form caching for a form that has a fixed layout . . . 714
Rendering a form design that has a flowable layout on the client 715
Prepopulating form fields with data 715
Verifying your XML data source for data merging 716
Creating HTML forms 716
Using subforms to create HTML pages 716
Layout considerations for HTML forms 719
Image guidelines for creating HTML forms 719
Working around web browser limitations 720
Previewing an HTML form 720
Where calculations and scripts are run 721
Client- and server-side processing 721
Client-side processing 722
Server-side processing 722
Events that you can reference in a script or calculation 722
Server-side scripts 722
Client-side scripts 723
Summary of supported properties, methods, and events . . . 724
Expressions supported by HTML clients 728
How to write a script to handle HTML pages 729
Optimizing and improving performance for Adobe document services 731
Form caching 731

To specify form caching for a form that has a fixed layout . . . 732
Rendering a form design that has a flowable layout on the client 732
Creating scripts for use with Adobe document services 733
Where calculations and scripts are run 733
Client- and server-side processing 733
Client-side processing 733
Server-side processing 734
Events that you can reference in a script or calculation 734
Summary of supported properties, methods, and events . . . 734

Chapter: 20 Creating Forms for forms workflow 740

Preparing a form for use in Workspace 740
Using the Form Bridge object with form designs that contain Artwork 742
To submit form designs that contain Artwork to Workspace: . . 742
How Workspace handles buttons in PDF forms 742
Fields in the Process Fields object 743

Chapter: 21 Working with the Keyboard 745

Default keyboard shortcuts 745
Keys for common actions 745
Keys for undoing and redoing an action 745
Keys for deleting text 746
Keys for selecting text 746
Keys for copying and moving text 747
Keys for applying character formats to selected text 747
Keys for special characters 748
Keys for setting line spacing 748
Keys for aligning paragraphs 748
Keys for controlling windows 749
Keys for controlling palettes 749
Keys for navigating a form design 750
Keys for controlling menus 751
Keys for the Script Editor or XML Source Tab 752
Keys for manipulating objects 752
Keys for changing the tabbing order 753
Keys for selecting and navigating within a table 754
Keys for navigating the Designer Help 755
Controlling the palettes with a keyboard 755
To size the Data View Info area 756
To size the Object Library Preview area 756
To size the Fragment Library Preview area 756
Customizing keyboard shortcuts 756
To create a customized keyboard shortcut set 756
To assign a keyboard shortcut to a command 757
To remove a keyboard shortcut from a command 758

To verify the current keyboard shortcuts for a command . . . 758

Chapter: 22 Menu, Command, Toolbar, and Dialog Box Reference 760

Menus and Commands 760
To open a menu using the keyboard 760
To display a context menu 760
Menu bar commands 760
File menu 760
Edit menu 762
View menu 764
Insert menu 766
Table menu 768
Layout menu 770
Tools menu 772
Window menu 773
Palettes menu 775
Help menu 777
Palette menus 778
Hierarchy palette menu 778
Data View palette menu 778
Tab Order palette menu 780
PDF Structure palette menu 781
Style Catalog palette menu 782
Layout palette menu 782
Border palette menu 783
Object palette menu 783
Accessibility palette menu 784
Font palette menu 784
Paragraph palette menu 785
Drawing Aids palette menu 785
Info palette menu 786
Report palette menu 786
How To palette menu 787
Object and Fragment Library palette menus 788
Object Library palette menu 788
Standard menu 788
Custom menu 789
Barcode menu 790
Fragment Library palette menu 791
Fragment Library panel menu 791
Template Manager menu 792
Toolbars 793
About toolbars 793
Standard toolbar 793
Text Formatting toolbar 795
Layout toolbar 796

Styles toolbar 797
Table toolbar 798
Tools toolbar 799
Dialog Box Reference 799
Action Builder dialog box 799
Additional Assets dialog box 800
Binding Properties dialog box 801
Check Spelling dialog box 802
Collection Editor dialog box 803
Collection List dialog box 804
Connection Properties dialog box 805
Create Fragment dialog box 805
Customize Toolbars dialog box 806
Toolbars (Customize Toolbars dialog box) 806
Commands (Customize Toolbars dialog box) 807
Data View Options dialog box 807
Delete Tab Stops dialog box 808
Digital IDs dialog box 808
Windows Certificate Store tab (Digital IDs dialog box) 809
Digital IDs Files tab (Digital IDs dialog box) 809
Dynamic Properties dialog box 809
Edit Conditional Breaks dialog box 811
Edit Data Nominated Subforms dialog box 814
Edit Dictionary dialog box 816
Edit Fragment Identifier dialog box 817
Edit Hyperlink dialog box 817
Encrypt Data and Submit Settings dialog box 817
Encrypt Data (Encrypt Data and Submit Settings dialog box) . . 818
Encryption Algorithm (Encrypt Data and Submit Settings dialog box) 818
Encryption Certificates (Encrypt Data and Submit Settings dialog box) 819
Issuers and Policies (Encrypt Data and Submit Settings dialog box) 820
Key Usage (Encrypt Data and Submit Settings dialog box) 821
Extract Style Sheet dialog box 822
File Import Options 822
Adobe Output Designer form (File Import Options dialog box) 822
Microsoft InfoPath (File Import Options dialog box) 823
Microsoft Word (File Import Options dialog box) 824
XForms Model (File Import Options dialog box) 824
Form Properties dialog box 825
Info (Form Properties dialog box) 825
Defaults (Form Properties dialog box) 826
Run-time (Form Properties dialog box) 827
Formatting (Form Properties dialog box) 828
Default Fonts (Form Properties dialog box) 830
Form Validation (Form Properties dialog box) 831
Server (Form Properties dialog box) 832

Variables (Form Properties dialog box) 832
PDF Security (Form Properties dialog box) 833
PDF Print Options (Form Properties dialog box) 834
Compatibility (Form Properties dialog box) 835
Preview (Form Properties dialog box) 836
Save Options (Form Properties dialog box) 837
Data Binding (Form Properties dialog box) 838
Fragment Information dialog box 838
Import Options dialog box 839
Insert Fragment dialog box 841
Insert Hyperlink dialog box 841
Insert Table dialog box 842
Keyboard Shortcuts dialog box 843
Missing Fonts dialog box 844
Change Font Substitutions dialog box 844
Network Connection Not Found dialog box 845
New Data Connection dialog box 845
Authentication (New Data Connection dialog box) 845
ADO Properties (New Data Connection dialog box) 846
OLEDB Connection (New Data Connection/Connection Properties dialog box) 847
New Connection (New Data Connection dialog box) 847
Sample XML Data (New Data Connection/Connection Properties dialog box) 848
Operation (New Data Connection dialog box) 848
WSDL File (New Data Connection dialog box) 848
Adobe Data Model (New Data Connection dialog box) 849
XML Schema (New Data Connection/Connection Properties dialog box) 849
New dialog box 850
Open dialog box 851
Options dialog box 851
Document Handling (Options dialog box) 851
Workspace (Options dialog box) 853
Spelling (Options dialog box) 854
Formatting (Options dialog box) 855
Default Fonts (Options dialog box) 856
Wizards and Tips (Options dialog box) 857
Bidirectional Text (Options dialog box) 859
Data Binding (Options dialog box) 859
International (Options dialog box) 860
Tab Order (Options dialog box) 861
Security (Options dialog box) 861
Patterns dialog box 861
Display (Patterns dialog box) 862
Edit (Patterns dialog box) 863

Validation (Patterns dialog box) 863
Data (Patterns dialog box) 864
Print dialog box 864
Publish To Repository dialog box 865
Remove Style dialog box 866
Report Warnings Options dialog box 866
Save As dialog box 867
Select Digital IDs dialog box 868
Signature Settings dialog box 868
Signature Handler (Signature Settings/Sign Data and Submit Settings dialog box) 869
Signing Certificates (Signature Settings/Sign Data and Submit Settings dialog box) 873
Issuers and Policies (Signature Settings/Sign Data and Submit Settings dialog box) 875
Signing Reasons (Signature Settings/Sign Data and Submit Settings dialog box) 877
Key Usage (Signature Settings/Sign Data and Submit Settings dialog box) 878
Signature Info (Signature Settings/Sign Data and Submit Settings dialog box) 879
Sign Data and Submit Settings dialog box 880
About the Sign Data and Submit Settings dialog box 880
Sign Data (Sign Data and Submit Settings dialog box) 881
Specify Substitute Font dialog box 881
Spreadsheet Paste Options dialog box 881
Style Editor dialog box 882
General (Style Editor dialog box) 882
Caption Text Formatting (Style Editor dialog box) 883
Value Text Formatting (Style Editor dialog box) 885
Border (Style Editor dialog box) 885
Layout (Style Editor dialog box) 886
Syntax Formatting dialog box 887
Table Assistant dialog box 887
Body Layout (Table Assistant dialog box) 887
Footer Row (Table Assistant dialog box) 888
Header Row (Table Assistant dialog box) 888
Row Shading (Table Assistant dialog box) 889
Sections (Table Assistant dialog box) 889
Template Manager dialog box 890
Template Options dialog box 891

Chapter: 23 Best Practices for Creating Adobe Forms Designs 892

Chapter: 24 Glossary 893

A . . . 893
accessible forms . . . 893
Acrobat form . . . 893
Adobe certified document . . . 893
application . . . 893
B . . . 893
binding data . . . 893
boilerplate . . . 893
boilerplate object . . . 894
C . . . 894
canonical format . . . 894
certificate . . . 895
client . . . 895
credential . . . 895
F . . . 895
fixed layout . . . 895
flowable layout . . . 895
form . . . 896
form authors . . . 896
FormCalc . . . 896
form design . . . 896
form developers . . . 896
form object . . . 896
I . . . 896
interactive form . . . 896
P . . . 896
PDF form . . . 896
permissions . . . 897
prepopulated form . . . 897
R . . . 897
render . . . 897
restricted document . . . 897
rights-enabled document . . . 897
run time . . . 897
S . . . 897
script fragment . . . 897
security handler . . . 898
static form . . . 898
subform . . . 898
T . . . 898
tagged PDF form . . . 898
trusted identity . . . 898
U . . . 898
usage rights . . . 898
X . . . 899
XML Forms Architecture . . . 899

XML form 899

1. Welcome to Designer

Welcome to Designer.

1.1. About Designer

NOTE: Effective March 10, 2012, Adobe is deprecating the Guides capabilities of Adobe® LiveCycle® ES. The Guides functionality is available for upgrade purposes only and will be removed from the product after two major releases.

Designer is a point-and-click graphical form design tool that simplifies the creation of forms. Form recipients can fill a form online, submit the data, and print it, or print and fill the form by hand. You can design a form, define its logic, and modify it to match paper counterparts or to meet strict legislative requirements. Form developers can use Designer to create applications that generate dynamic, data-driven documents and produce customized business documents for print, web, or archival. Using form designs, form developers can create, interactive data capture applications by leading users through a series of visually appealing and streamlined panels, improving usability and reducing data entry errors.

You can also build and maintain data capture solutions that read from, validate against, and add to corporate data sources. With Designer, you can integrate PDF documents into existing workflows by binding forms to XML schemas, XML sample files, databases, and web services.

Forms and documents that are created in Designer can be merged with business data and rendered as a number of file types, including Adobe PDF, HTML, SWF, and printing for PCL, Adobe PostScript® and Zebra (ZPL) printers.

1.2. New features in Designer

Preview your XDP forms in HTML in Designer

Besides previewing the PDF rendition of forms, now you can also preview the forms in HTML rendition. While designing the form in Designer, click on the new Preview HTML tab, to preview the form as it would appear in a browser.

See HTML Preview.

Preview your XDP forms in HTML in forms manager

You can now use forms manager to preview and manage your AEM forms. You can launch forms manager from within Designer. To configure forms manager to launch from the Designer, see Launch forms manager fromDesigner.

Accessibility Checker

An accessible form is one that a wide range of people can use, including those with disabilities that affect how they are able to interact with the form on a computer screen. You can now use the Accessibility Checker to test the accessibility of your Designer forms against a set of accessibility rules. For more details, see *Checking the accessibility of a form*.

Add document title to the PDF title bar

You can now add the document title to the PDF title bar. If a user opens the PDF, the document title will display on the title bar of the window. For more details, see *To add document title to the PDF title bar*.

1.3. Where to find documentation, samples, and tutorials

The following resources can help you start building forms and applications.

Learn about AEM forms

Learn about AEM forms and get important information about Designer that was not available when the product documentation was written.

Documentation	Description and location
Overview	Provides an overview of AEM forms. See Overview.
Workbench Help	Available from the Help menu when you install Workbench.
Creating Dynamic Forms with Designer by J.P. Terry	Shows you how to use Designer to create interactive and dynamic forms. Available at Adobe Press.

Get started with samples and tutorials

These tutorials take you through the process of creating forms for applications. Form samples help you learn about form design techniques and provide a starting point to creating a form design. Scripting samples demonstrate quick and simple form solutions.

Documentation	Description and location
Quick Start tutorials	Designed to get you started on creating forms using Designer. See any of the following tutorials: *Creating a purchase order form*, *Creating a purchase order form that has a flowable layout*, and *Creating and using fragments in form designs*.

Documentation	Description and location
Tutorial for creating a AEM forms application	See Creating Your First AEM forms Application.
Designer form fragment tutorial	Walks you through the steps of working with form fragments. See Designer form fragment tutorial.
Form samples	Designer includes a selection of complete samples, each one accompanied by a form design, sample data, and/or schema, and the final version of the form. The samples are installed with Designer in the default location installation directory under \...\EN\Samples.

Add calculations to forms

Use FormCalc to build calculations into Designer forms.

Documentation	Description and location
FormCalc Reference	See Introducing FormCalc.

Enhance forms by using scripts

Use scripts to build intelligence into forms.

Documentation	Description and location
Scripting Basics	Provides an overview of how you can use scripting to develop and enhance forms that are created in Designer. See Scripting Basics
Scripting samples	You can experiment with the sample forms and apply the techniques to your own work. Go to the Adobe Developer Connection to get the latest sample forms.
Scripting Reference	Describes the objects, properties, and methods that are available for scripting in the Designer environment. See Scripting Reference.

2. Getting Started

2.1. About forms

A *form* is the document that a user views or interacts with. It is derived from a form design that you create using Designer.

Forms typically collect or present structured data and are the front end to a business process. Designer creates forms and documents that can be merged with business data and rendered in a number of file types, including PDF documents, HTML, Flash (SWF file), and printing for PostScript and Zebra (ZPL) printers.

Forms can capture or present information in three different ways.

Interactive forms

Forms can be designed to capture data directly from end users. Users fill the form and select options online, and return the form data according to some prescribed process. These forms are known as interactive forms. You can author interactive forms that the user fills using Acrobat Professional or Adobe Reader, or you can author forms for Forms Generator that the user fills in a web browser.

Interactive forms have many benefits over paper-based forms:

- Although interactive forms may look like traditional paper-based forms, they eliminate cumbersome and time-consuming effort required to process paper forms. Using interactive forms to provide business solutions makes sense in the worlds of the Internet and enterprise-wide computing.
- You can deliver interactive forms through Internet, intranet, or email. You can automate the document exchange process, store forms in reliable formats, and protect document content and integrity.
- Interactive forms allow you to streamline your data collection process. An interactive form can collect and integrate data into your existing core data collection systems, thereby extending their value. The form might integrate data directly to your data collection system or use a program on the server, such as a CGI script, an ASP page, Java Server Pages (JSP), or servlet.
- Using interactive forms, you can also establish online forms-based workflow processes using built-in logic to route the form electronically from one user to the next. Interactive forms can also support assistive technologies, such as screen readers, so that you can extend the form to users with disabilities.

In the simplest scenario, end users only require Adobe Reader to electronically fill the form and send the form data to the originator of the form or print the form and send the paper copy of form and data to the originator. If you have purchased Forms, the interactive form can be in PDF or HTML. In this case, users open and fill the form using a web browser.

Interactive forms typically include data entry features such as selection lists, drop-down lists, check boxes, automatically generated calculations, validation messages, digital signatures, and Submit and Execute buttons. Form authors can use built-in FormCalc functions and custom scripting by using JavaScript™ to extend the functionality of interactive fields. Interactive forms can include command buttons so that users can save the data to a file or database or to send the data by email to a specified address. In addition, validations can be added to ensure the accuracy of user-entered data. The form can provide feedback such as messages to prompt for specific types of data.

Non-interactive forms

Forms can be designed to present information to end users. The data can come from a variety of data sources, such as databases, web services, or enterprise content management systems. The end user views the form already prepopulated with data. The end user cannot modify the data in the form or add new data to the form. These types of forms are known as *non-interactive forms*. A typical scenario for these types of forms involves Forms as part of the solution. Forms merges the form design with data and renders the form, prepopulated with data, to the end user. A classic example of a non-interactive form is a credit card statement or telephone bill.

In yet another scenario, a form might be designed to initially present information to the end user, and then provide the capability for the end user to supply additional information and send it to the initiator or server for further processing.

Print and fill forms

Another type of form is the print-and-fill form. The form author creates a form design in Designer and typically saves it as a PDF. The end user opens the form in Acrobat or Adobe Reader, prints the form, and fills the form manually. The form is then returned to its originator by fax or land mail.

RELATED LINKS:

Guidelines for forms

2.2. About form designs

Form design is the term that describes the form during its creation or design process. The form design specifies the layout, data capture, and final presentation rules for the form. The presentation rules are applied when the form is filled with data. How the form is filled with data depends on the purpose of the form, which in turn influences the type of form you create.

In Designer, you add functionality to your form by adding objects, such as a place to enter text, or a button for sending the form by email. Static objects are available that present fixed text or graphic information. These objects include circles, lines, rectangles, images, and text

2.3. Form design layouts

Fixed layout

The most common type of forms have a *fixed layout*; that is, they have a predetermined layout, always with a fixed number of pages regardless of the amount of data available to fill it. For example, a course registration form that an end user can either print and fill by hand, or fill in Acrobat or Adobe Reader. When filled, the form retains its original layout and number of pages. Fields that are not filled remain empty. Conversely, if the amount of data is more that the form can hold, the form cannot expand to accommodate excess data. For example, if a course registration form has 5 rows where end users list their course selections, and enough data is available to fill 10 rows, only 5 rows can be filled. Similarly, if an end user lists only 2 course selections, you will still see 5 rows, 2 that are filled and 3 that are empty.

This form can be interactive, where an end user fills the form typically in Acrobat or Adobe Reader, or it can be non-interactive, where a server process merges the form with data from a data source. Similarly, Forms typically renders non-interactive forms that have a fixed layout to present information from a data source.

Flowable layout

In addition to forms that have a fixed layout, you can define sections of the form that will expand and shrink in response to the amount of data that is merged when the form is rendered. You do this by wrapping various sections (groups of subforms) in subforms that are set to flow content. Then, at run time, only the subforms that are necessary for displaying the exact amount of data are instantiated.

This type of form has a *flowable layout* with a varying number of pages. The subforms adjust depending on the amount of data merged with the form when it is rendered, or the subforms expand when end users need to add more data. For example, you may decide to let end users add to the form the number of rows they need to list their selections, remove rows from the form, and then return the form data electronically. Depending on how many rows they add, the form may extend over two or more pages.

Interactive forms that have a flowable layout are sometimes referred to as client-side forms. Acrobat and Adobe Reader 7.0 and later support this type of interactive forms.

You can also create forms that have a flowable layout for use with Forms. In this scenario, Forms merges the form design with data. For example, such forms as a telephone bill or credit card statement are typically non-interactive forms and designed to present users with information from a data source. Users then print these forms or store them electronically. These forms are sometimes referred to as server-side forms because the merging of the form design and data occurs at the server.

RELATED LINKS:

About forms

Creating interactive forms that have a flowable layout

Parts of a form design

2.4. Parts of a form design

You create a form design by dragging objects from the Library palette onto one or more pages in the Layout Editor and working in the Object palette to modify the properties specific to the object you select. The object that you select determines which tabs are available in the Object palette. Many other palettes are available that contain specific properties that you may want to change. For example, you can use the Font palette to change the font family, size, and style of text in a selected object.

You place the objects you want in the form design on pages. If you want an object to appear on each page of the form design, you place it on a master page. For example, you can include a logo, watermark, or introductory information that always appears in the same location on each page of the form design.

The following key components make up a form design:

- Master pages
- Pages
- Content areas
- Subforms
- Fields
- Boilerplate objects

Form design elements are displayed in the Hierarchy palette.

Designer automatically generates XML source code for each object as you build the form design. Because Designer automatically generates the XML source code for you, you can create form designs without having any knowledge about XML.

Master pages

Every form design contains at least one master page that Designer creates automatically. *Master pages* are designated to format pages, and they help to facilitate design consistency because they can provide a background and layout format for more than one page in a form design.

You can use the supplied master page to format pages, edit the master page's settings, or add additional master pages if needed. If you are creating a simple interactive form, you would probably use the supplied master page without changing its settings.

At the very least, master pages define the orientation and dimensions of pages. You can use master pages to define these aspects of a form design:

- Page size and orientation
- Headers and footers
- Watermarks and company logos

Each master page is created with a default content area that covers the whole page. You can add text, images, and other boilerplate objects to a master page. These objects are displayed on all of the pages that the master page formats. (See *Using master pages*.)

NOTE: Text fields, numeric fields, and date/time fields on master pages will not be interactive on Acrobat 6-compatible forms. Users cannot modify the associated data in these fields.

You manipulate master pages in the Master Pages tab.

Pages

Pages represent the pages of a form. Each page derives its size and orientation from a master page and, by default, each page is associated with the default master page that Designer creates. Each page is created with a default subform that covers the whole page. (See *Setting up pages*.)

If your form design contains more than one master page, you can choose which master page to assign to a page. (See *Using master pages*.)

You work with pages in the Design View tab.

Content areas

*Content areas*define where objects can be placed on pages. When you design a form, you cannot place an object on a page unless it is inside the area bounded by a content area. You can add content areas to master pages only.

Whenever you create a new master page, Designer creates a default content area on the master page.

A form design that has a fixed layout will typically contain one content area. A form design that contains sections that adjust to accommodate data can have one or more content areas. You can specify whether the objects in each content area should be positioned from top to bottom, or from left to right and top to bottom. (See *Using content areas*.)

Subforms

Subforms are container objects that you can use to group form design objects, including fields, boilerplate objects, and other subforms. When they are grouped, you can control whether the subform and the grouped objects appear on your form based on data bindings that you configure for your form. You can also configure subform objects to be repeatable, which lets you have multiple instances of a single subform and its grouped objects appear on your form. This allows you to create more flexible and adaptable form designs.

Subforms are essential when creating forms that contain sections that expand to accommodate data because they provide the dynamic capabilities to be visible, to remain hidden, and to grow, all in response to data and user interaction at run time. (See *Subform*.)

Tables

Tables are essentially structured container objects that you can use to organize your form design content in meaningful, logical ways. Each cell of a table is a separate container capable of storing form design objects. (See *Using tables*.)

Tables are very similar to subforms in terms of functionality and behavior. Like subforms, tables can be dynamic, which means they can repeat and grow in response to data and user interaction at run time. (See *Subform*.)

Field objects

Designer provides a number of field objects that are capable of capturing, merging, and displaying data. A *field object* provides a data-entry region, and users can interact with field objects by entering or selecting an associated data value. (See *About Objects*.)

The following objects are field objects:

- Button
- Check box
- Date/time field
- Decimal field
- Signature Field
- Drop-down list
- Email Submit button
- HTTP Submit button
- Image field
- List box
- Numeric field
- Paper Forms Barcode
- Password field
- Print button
- Radio button
- Text field

Boilerplate or static objects

Boilerplate or *static objects* are read-only objects that improve the aesthetic appeal of a form and may provide context or assistance for users. They can be added to pages or master pages. (See *About Objects*.)

The following objects are boilerplate objects:

- Circle
- Image
- Line
- Rectangle
- Text

2.5. Installing and configuring Designer

Designer is available as a standalone installer and is also bundled with workbench. To know more about installing Designer as part of Workbench, see Installing and running Workbench in this document. If you are using a standalone installer for Designer, follow these steps:

1) Launch the Designer installer by double-clicking `setup.exe`.
2) Proceed and provide your details and the serial number on the Personalization screen.
3) If you accept the license agreement, click Next to proceed.
4) (Optional) change the default installation path, if you want to install Designer ES4 at a location of your choice. Click Next.
5) (Optional) If you have a previous version of LC Designer installed, the installer provides an option to migrate, delete, or ignore the user settings of the previous installation. Click Next.
6) Click Back to change any preferences. To install Designer, click Install.
7) Click Finish when the installation completes.

Configuring JVM settings

By default, a Java Virtual Machine (JVM) is initialized when Designer is launched. ***NOTE:*** *To initialized the JVM, install JDK 1.5 or later.*You can modify the JVM configuration and the default behavior using the supplied INI file. The file name is `JVMSettings.ini` and it is available at `[`*`Designer Install Directory`*`]\cfg\` location. The various configurable parameters are:

JVMOnDemand

specifies if the JVM is initialized with Designer (value `0`) or is initialized only when required (value `1`). By default, the value is `0`.

JVMHeapSizeCustomization

specifies if the heap size for JVM is determined by the JVM itself (value `0`) or by Designer (value `1`). By default, the value is `0`.

JVMInitialHeapSize

specifies the initial heap size in KB. This setting is relevant if `JVMHeapSizeCustomization` = `1`. If the value is <1025, it is set at 1025 (default value).

JVMMaxHeapSize

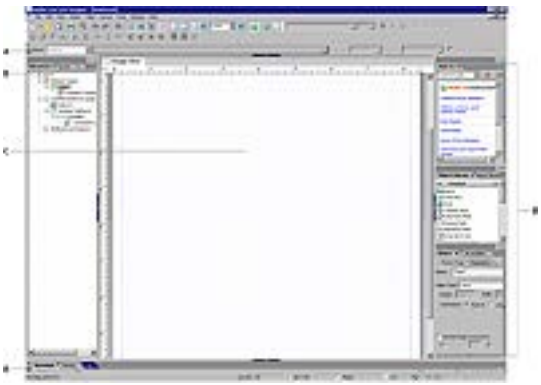
specifies the maximum heap size in KB that JVM can grow to. This setting is relevant if `JVMHeapSizeCustomization` = `1`. If the value is <2048, it is set as 4096 (default value).

2.6. Looking at the workspace

About editors

By default, the Designer workspace consists of an area called the Layout Editor where you create and lay out your form design and a Script Editor where you can write scripts to extend the capabilities of the form design.

As you become comfortable working with form designs, you can customize the workspace to suit your requirements.



A.

Script Editor

B.

Palettes

C.

Layout Editor

Layout Editor

The Layout Editor is the main area where you create and maintain the form design. It contains four tabs:

Design View tab

Displays the pages that make up the form design. The pages in Design View contain a form design's content. The first time you start Designer or create a new form design, the Design View tab displays a page ready for objects to be added. Objects in the master page appear in the page but cannot be selected.

Master Pages tab

Displays the master pages that can be applied to pages in Design View. Master pages specify the layout and the background for the form design. You add objects that will occur in the same position throughout the form design on a master page. The Master Pages tab is hidden by default. Objects in the Design View page do not appear in the master page.

XML Source tab

Displays the XML source code that describes the structure of the form design and its objects. It is recommended that you do not edit the XML source code directly.

Preview PDF tab

Displays a PDF form based on the current form design. As you work, if you have Acrobat or Adobe Reader installed, you can preview the form in the Preview PDF tab. Use the Preview PDF tab to view and test the operation of a form or template as if it were a PDF file. You can set options for previewing interactive forms or printable forms in PDF by using the Form Properties dialog box (Preview tab).

NOTE: To preview a form in the Preview PDF tab, in Acrobat, ensure that the Edit > Preferences > Internet > Display PDF In Browser option is selected.

By default, not all the tabs are displayed. The Design View tab is displayed; however, the Master Pages tab and the XML Source tab are not displayed. The Preview PDF tab is displayed only when Acrobat or Adobe Reader is installed.

Tabbed palettes are arranged around the Layout Editor to provide easy access to the tools without cluttering your workspace.

For more information see *To customize the Layout Editor*.

Script Editor

The Script Editor is where you create, modify, and view the calculations and scripts of a particular form. For example, you can use the Script Editor to write a simple calculation that adds two numeric fields or complex scripts that alter the appearance of the form based on end-user actions. Designer supports scripting either in its own scripting language called FormCalc or in JavaScript.

By default, the Script Editor appears at the top of the Designer workspace, but you can dock it anywhere. It has both a single-line view and a multiline view that you can switch between, depending on your needs. Single-line view is designed to maximize the amount of space dedicated to the Layout Editor and other palettes. Multiline view is designed to maximize the amount of space for writing script.

Show

Lists all form design events that support user-defined scripting. Any events that do not apply to a particular object appear dimmed. Events that contain a calculation or script display an asterisk (*) beside the name of the event.

Show Events for Child Objects

Displays the event you have currently selected in the Show list for the current object and all of its child objects. If you select the uppermost object in the Hierarchy palette, this option displays the event you have currently selected in the Show list for all objects on your form.

Functions

Displays a list of available built-in FormCalc or JavaScript functions, depending on the scripting language you currently have selected in the Language list.

To place a function onto your script editing field, select a function from the list and press Enter.

Check Script Syntax

Checks all of the scripts in a form for correct syntax and reports any errors on the Warnings tab in the Report palette.

Language

Specifies the scripting language you want to use for the current calculation or script. Two options are available:

- **FormCalc** FormCalc is a native Adobe calculation language typically used for shorter scripts, such as simple calculations.
- **JavaScript** JavaScript is the default scripting language for new forms.

  The scripting language that is displayed in the Language list matches the scripting language option you select as the default for new forms in the Workspace panel in the Options dialog box. However, if you change the scripting language setting for the current form on the Defaults tab in the Form Properties dialog box, the scripting language that is displayed in the Language list changes similarly for any new scripts on new events. Changing the scripting language option in the Form Properties dialog box does not change the scripting language for existing scripts. If an event already contains script and that script is deleted, the Script Editor continues to use that same scripting language for the duration of your Designer working session.

Run At

Specifies where the calculation or script will execute. Three options are available:

- **Client** Calculations and scripts execute while the client application (for example, Acrobat, Adobe Reader, or a web browser) processes the form.
- **Server** Calculations and scripts execute while the server application (for example, Forms) processes the form.
- **Client and server** Calculations and scripts execute while the server application (for example, Forms) processes the form, except in cases where the HTML client application supports client-side scripting. For example, a script that accesses a database to prefill data on a form.

For more information, see *Scripting Using Designer*.

About palettes

The palettes provide easy access to the tools without cluttering your workspace. Palettes can include one or more tabs, each containing common properties. For example, the Object palette can include one or more tabs.

You can arrange the palettes in the workspace to suit your work style. For example, you can hide the rarely used palettes and move the frequently used ones into one palette window.

As you work in the Layout Editor, the information that appears in certain palettes changes to reflect the selected object. For example, if you select an object, the information in the Layout palette changes to display information about the object's size and position.

Hierarchy palette

The Hierarchy palette is a graphical representation of the contents in the Design View and Master Pages tabs.

Whatever you select in the Hierarchy palette is also selected in the body or master page that it is associated with. See *Hierarchy palette menu*.

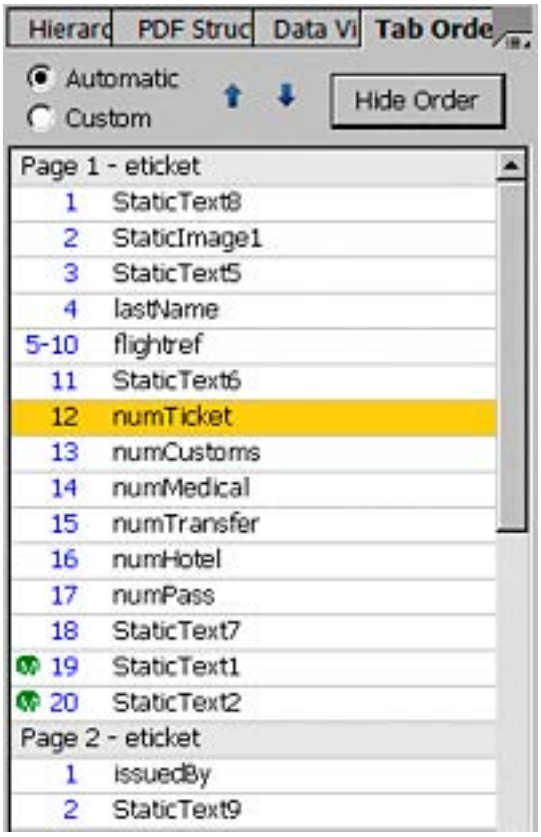
Data View palette

If a data connection exists, the Data View palette displays the hierarchy derived from the data connection. The top nodes in the hierarchy represent each data connection and display the name of the data connection. A data connection provides a link between the form and the data source.

When you design a form based on a data connection, Designer builds a data structure for your form based on that data source. You can filter the nodes to work with and quickly create a form using some or all of the data source. You then use binding to link a node from a data source to an object on the form. See *Connecting to a data source*.

Tab Order palette

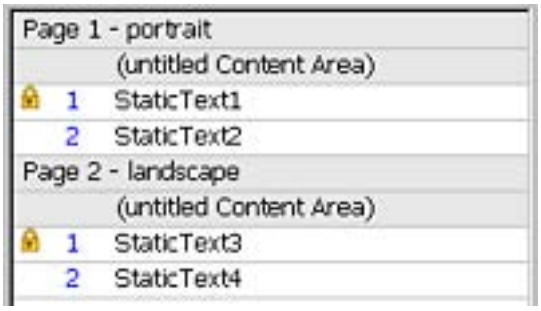
The Tab Order palette displays a numbered list of all the objects on the form, where each number represents the position of the object within the tabbing order.



The Tab Order palette may show the following visual markers in the list:

- A gray bar marks each page of the form. The tabbing order on each page starts with the number 1.
- The letter *M* inside a green circle indicates master page objects (visible only when viewing the form on the Design View tab).
- A range of numbers indicates objects within a fragment reference.
- A yellow background indicates the currently selected object.
- A lock icon beside the first object on the page indicates that the object cannot be moved within the order (visible only when viewing the form on the Master Pages tab).

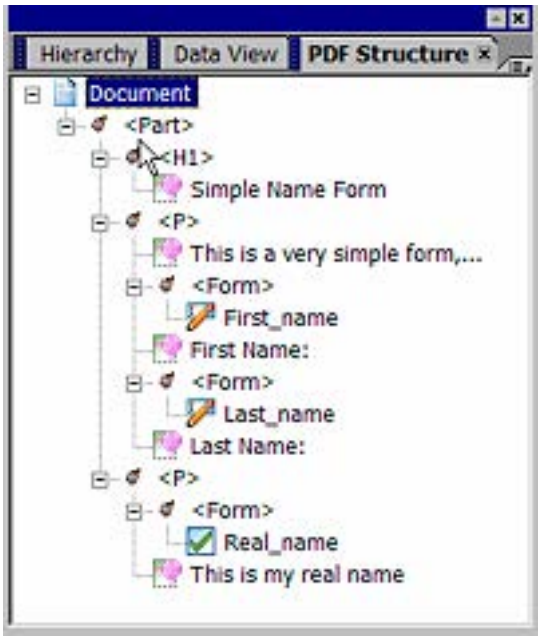
For more information see *Using the Tab Order palette*.



PDF Structure palette

The PDF Structure palette displays a view of the hierarchical structure of tagged PDF documents, which provide accessibility and a defined tabbing and reading order for assistive technologies, such as keyboard access and screen readers. See the *PDF Structure palette menu*.

For information about PDF documents as artwork, see *Importing PDF documents as artwork*.



Object Library palette

The Object Library palette contains all the objects that you can add to a form design. Objects are organized into categories:

Standard

Contains the most commonly used form objects, such as check boxes and text fields.

Barcodes

Contains a list of barcode objects.

Custom

Contains preformatted objects, such as address blocks and phone number fields.

For information about using the library palettes, see *Managing library palettes*.

Fragment Library palette

The Fragment Library palette contains the fragment libraries that are currently open. A fragment library corresponds to a folder in your file system that contains the fragment source XDP files.

Each fragment library has an expandable panel in the library that lists the available fragments.

My Fragments

A location for the fragments that you create. You can insert them in a form design or use them to create new fragments.

Style Catalog palette

Use the Style Catalog palette to manage styles sheets, and to edit and apply styles to objects in a form design. The Style Catalog palette lists the various style sheets available with a form and the styles included with each style sheet. The Style Catalog includes a different panel for each style sheet. Each panel lists the styles included with that style sheet. The first panel is the Internal Style Sheet panel. Below the internal style sheet panel are panels for each Designer Style Sheet file (XFS) file that you add to the Style Catalog. See *Styles* .

Layout palette

Use the Layout palette to set the following properties for the selected object::

- Size and position of the object.
- Whether the object should ignore the defined height and width, and expand to reveal all of its content.
- Position of the anchor (insertion) point. You can rotate an object around its anchor point in a 90°, 180°, or 270° increment.
- Align selected objects in subforms that flow content.
- Margins around the object.
- Caption position and width. You can also hide the caption.

For more information, see *Formatting objects*.

When you select an object, the Layout palette automatically displays the selected object's settings. You can edit most of an object's layout settings directly in the Layout Editor. For example, to change an object's position, you can drag it to the new location on the page.

Border palette

Use the Border palette to edit the border properties for objects in the form design. You can edit the borders individually (left, right, top, and bottom) or together. You can also specify the type of border corner and background color.

For more information, see *Border properties in the Border palette*.

Object palette

Use the Object palette to modify properties that are specific to the selected object. The object that is selected in the Layout Editor determines which tabs are available in this palette.

Accessibility palette

Use the Accessibility palette to specify custom text for an object that a Microsoft® Active Accessibility (MSAA)- compliant screen reader reads as it passes through the form. (See *Making objects accessible*.) If custom screen reader text is available for the object, the screen reader will read the custom text and not the tool tip.

You can also change the default order in which the screen reader searches for text to read on an object-by-object basis, and you can turn off screen reader text for any object.

For more information, see *Accessibility properties in the Accessibility palette*.

Font palette

Use the Font palette to change the font family, size, style, and scale, as well as the baseline shift, letter spacing, and kerning of the text in one or more selected objects. You can change the font properties of text in text objects, in the caption area of objects such as text fields, decimal fields, and numeric fields, and in the value area of text field objects.

For more information, see *Formatting text*.

Paragraph palette

Use the Paragraph palette to change the alignment, indentation, line spacing, and hyphenation of the selected text. You can also set the radix alignment for a Numeric Field object. The options that appear in the Paragraph palette depend on what is selected.

For more information, see *Formatting paragraphs*.

Use the Paragraph palette to create lists and to change the alignment, indentation, line spacing, and hyphenation of the selected text. You can also set the radix alignment for a Numeric Field object. The options that appear in the Paragraph palette depend on what is selected.

Drawing Aids palette

Use the Drawing Aids palette to specify the grid and ruler settings and drawing units. You can also show or hide object boundaries and specify boundary border style. The settings for snapping objects to other elements on the page and guideline definitions are also included on the Drawing Aids palette.

The horizontal and vertical rulers, the grid, and the long cross hairs help you position objects precisely across the width or length of a page. When visible, rulers appear along the top and left side of the active body or master page. Markers in the ruler display the pointer's position when you move it. Changing the ruler origin (the (0, 0) mark on the top and left rulers) lets you measure from a specific point on the form design.

You can also show or hide object boundaries on the form design. Showing boundaries is useful for displaying objects that do not have borders on the form. You can also specify border style for fields, subforms, content areas, groups, and other objects.

When you move objects, the snapping options enable you to automatically position them in relation to other elements on the page. Objects can snap to the grid, a guideline, another object, or the center of the page.

You can use guidelines as a visual cue or as elements for object snapping. Use the guideline definitions lists to add or remove horizontal or vertical guidelines.

For more information, see *To use the drawing aids*.

Info palette

The Info palette displays the metadata associated with the selected objects. This metadata is stored in the XML source as named children of a `<desc>` element. You can edit some metadata for the form design by using the Form Properties dialog box (Info tab).

Report palette

The Report palette provides information about the form design. The Warnings tab lists errors that are reported as you work in the form design, the Binding tab lists fields based on how you defined their binding data, and the Log tab shows a log of actions reported by Designer.

To clear the Warnings tab, you must address the errors listed. To clear the Log tab, click the palette menu and select Clear Warnings. See *Addressing warning messages in the Report palette*.

How To palette

The How To palette contains a list of help topics about common procedures in Designer. You can scroll the list to locate a topic of interest and click More Info to see the steps.

RELATED LINKS:

Data View palette menu

Tab Order palette menu

Object Library palette menu

Style Catalog palette menu

Layout palette menu

Border palette menu
Object palette menu
Accessibility palette menu
Font palette menu
Paragraph palette menu
Drawing Aids palette menu
Info palette menu
Info tab (Form Properties dialog box)
Report palette menu
To view a list of bound fields

Customizing the workspace

To customize the Layout Editor

You can customize the Layout Editor to show or hide tabs, resize pages, and zoom in and out.

To show or hide tabs

- To show or hide the Design View tab, select View > Design View.
- To show or hide the Master Pages tab, select View > Master Pages.
- To show or hide the XML Source tab, select View > XML Source.
- To show or hide the Preview PDF tab, select View > Preview PDF.

To zoom in and out

- To size the page to 100%, select View > Actual Size.
- To size the page to fit in the Layout Editor, select View > Fit Page.
- To size the page to fit the width of the Layout Editor, select View > Fit Width.
- To zoom in or out of the Layout Editor, select View > Zoom, enter a value from 25% to 500% in the Zoom To box, and then click OK. You can also use the Zoom To box in the toolbar.
- To magnify an area, select View > Zoom Area. The pointer changes to a magnifying glass. Using the magnifying glass, click and drag the area that you want to magnify.
- To display a specific page, zoom out so that you can see the entire page and scroll down to the page that you want to display.

To customize the Script Editor

You can show or hide the Script Editor and change the view from single-line to multiline.

- To show or hide the Script Editor, select Window > Script Editor.

- To change from single-line to multiline view, drag the Expand button on the Script Editor palette border until the palette is the required size.

For more information, see *Scripting Using Designer*.

To customize the palettes

You can customize the palettes to show or hide palettes, dock or move palettes, or size or reset palettes.

To show or hide palettes

- To show a palette, select Window > [name of palette].
- To hide a palette, select palette menu > Hide Palette.
- To show or hide all palettes on one side of the Layout Editor, click Expand on the palette border. Click it again to collapse the palettes.
- To show or hide all the palettes that are in the selected location, select Window > Workspace, and then select one of the commands.

To dock or move palettes

- To dock a palette on the side of the window, drag the palette bar to the side of the Designer window.
- To dock palettes together, drag the palette bar to the bottom of another palette. This procedure applies only to palettes that are docked to the Designer window; it does not apply to floating palettes.
- To move and dock a palette, drag the palette bar.
- To move a palette without docking it, Ctrl+drag the palette bar.
- To move a palette into another palette window, drag the palette tab to the target palette. You cannot move a sub-tab.

To size or reset palettes

- To change the size of a palette, drag any side of the palette.
- To reset palette locations, select Window > Reset Palette Locations.

To use the drawing aids

You can use the Drawing Aids palette to show or hide rulers and object boundaries, set the grid, and set other preferences.

To show or hide the drawing aids

- To quickly show the Drawing Aids palette, double-click the rulers.
- To show or hide the object boundaries, select Show Object Boundaries.
- To show or hide guidelines, or the grid, select Show Guidelines and Show Grid.
- To show or hide rulers, select Horizontal Ruler and Vertical Ruler.
- To show or hide cross hairs, select Long Crosshairs.

To use the grid

- To enable snap to grid, select Snap To Grid.
- To set the grid's origin, type new coordinates for the origin in the X and Y Origin boxes.
- To set the grid's interval, type new values in the X and Y Interval boxes. For example, if the grid is in 1-inch units and you specify an interval of 10, 10 grid points are displayed per inch.

To set snapping options

- To automatically place objects at locations on the grid, select Snap To Grid.
- To automatically place objects on guidelines you define, select Snap To Guideline.
- To automatically position the edge of an object using another object, select Snap To Object.
- To automatically place the center of an object at the vertical or horizontal center of the page, select Snap To Guideline.

To use guidelines

- To automatically place objects on guidelines, select Snap To Guideline.
- To add a guideline, starting from the upper-left corner of the Layout Editor, drag the pointer onto the page. To remove a guideline, select its triangle in the ruler and drag it off the page.
- Alternatively, under Guideline Definitions, to add a horizontal or vertical guideline, click and enter a numeric location. To remove a guideline, select it and click .
- Use the Drawing Aids palette menu to select preset guidelines (for example, to add guidelines for a standard US Letter or Legal-sized page).

To set other drawing aids

- To set the rulers' unit of measurement, select a unit from the Units list.
- To set the color of object boundary borders, click Styles, click the Color box, select More Colors, and either choose a predefined color from a palette or create a custom color by clicking Define Custom Color.
- To set the object boundary border style, click Styles and select the border style for the required object.
- To set the color of grids or guidelines, click the Color box, select More Colors, and either choose a predefined color from a palette or create a custom color by clicking Define Custom Color.

To customize the toolbars

You can customize the toolbars to show, hide, dock, or move toolbars, create new toolbars, remove toolbars that are no longer needed, add frequently-used commands or remove unused commands from a toolbar, or reset to the default toolbars.

To show, hide, dock, or move a toolbar

- To show or hide a toolbar, right-click a toolbar and select or deselect the toolbar that you want to show or hide.

- To dock a toolbar, drag the toolbar to the top of the Designer window.
- To move a toolbar, drag the toolbar to the new location.

To create a toolbar

1) Select Tools > Customize.
2) In the Toolbars tab, click New.
3) Type a name for the toolbar and click OK.
4) Add commands to the toolbar as required.

To delete a toolbar

1) Select Tools > Customize.
2) In the Toolbars tab, select the toolbar that you want to delete and click Delete.

NOTE: You can only delete user-created toolbars.

To rename a toolbar

1) Select Tools > Customize.
2) In the Toolbars tab, select the toolbar that you want to rename and click Rename.
3) Rename the toolbar and click OK.

NOTE: You can only rename user-created toolbars.

To show or hide tool tips

1) Select Tools > Customize.
2) Show or hide toolbar tool tips:
   – To show tool tips , select the Show Tool Tips check box.
   – To hide tool tips, clear the Show Tool Tips checkbox.

To add or remove a command on a toolbar

1) Select Tools > Customize.
2) In the Commands tab, select the toolbar that you want to edit in the Edit Toolbar list.
   Add or remove a button:
   – To add a command to a toolbar, select the command that you want to add and then click Add Tool to Toolbar. Use Move Tool Up and Move Tool Down to position the command in the toolbar.
   – To add a separator after a command in the toolbar, select the command in the toolbar, and then click Add Separator to Toolbar. Use Move Tool Up and Move Tool Down to position the separator in the toolbar.
   – To remove a command or separator from a toolbar, select the command or separator and then click Remove Tool from Toolbar.
3) Click Close.

To reset toolbar commands to the default

1) Select Tools > Customize.
2) In the Commands tab, select the toolbar that you want to reset in the Edit Toolbar list.
3) Click Reset Toolbar to Default.

NOTE: You cannot reset user-created toolbars.

To customize the tool tips for objects

You can display tool tips that show the position, the size, and the name of the object as you work on a form design.

- To show or hide the position of tool tips, select Tools > Options > Workspace, select or deselect Display Position And Size Tool Tips While Editing, and then click OK.
- To show or hide the object tool tips, select Tools > Options > Workspace, select or deselect Display Object Name Tool Tips While Pointing, and then click OK.

RELATED LINKS:

Layout Editor

Drawing Aids palette

Drawing Aids palette menu

Aligning and sizing objects

Managing library palettes

Object Library palette

You can add categories to this palette, rename existing categories, move objects between categories, and remove categories from this palette. You can also create and add your own objects to the categories. Each category corresponds to a folder in the file system.

To add a category to the Object Library palette

1) In the Object Library palette menu, select Add Group.
2) Type a name for the category and click OK. Adding a category to the Object Library palette creates a folder in the file system.

   TIP: To view the location of the new folder, click the category in the Object Library palette and, in the Object Library palette menu, select Group Properties. Notice the folder location in the Location box.

To delete a category from the Object Library palette

1) In the category menu, select Remove Group.
2) Do one of the following tasks:

- To delete a category, select Remove Group and All Its Objects.
- To delete a category and move the objects to another category, select Remove Group and Move Objects to *[category name]*.

3) Click OK.

To move an object to a different category in the Object Library palette

1) In the Object Library palette, right-click the object that you want to move and select Move Object To > *[category name]*.

   Only categories that allow objects to be added to it or removed from it are listed.

To customize the view in the Object Library palette

1) Do one or both of these tasks:
   - To change the size of the icons, in the category menu, select View > *[icon size]*.
   - To sort the icons, in the category menu, select View > Sorted List.

To set permissions for an Object Library palette category

1) In the category menu, select Group Properties.
2) Do one of the following tasks:
   - To allow objects to be added to a category, ensure that Allow Objects To Be Added is selected.
   - To allow objects to be removed from a category, ensure that Allow Objects To Be Removed is selected.
   - To allow objects to be modified in a category, ensure that Allow Objects To Be Modified is selected.

To add an object to the Object Library palette

1) In the Object Library palette, select the category where you want to store the object.
2) Drag the object into the Object Library palette.
3) In the Add Library Object dialog box, type the name of the object.
4) (Optional) Type a description for the object.
5) Click OK.

   If another object of the same name exists in that category, Designer prompts you to confirm whether you want to replace the existing object.

   From the Group Properties dialog box, some categories may have been set up to prevent objects from being added, removed, or modified. If a category prevents new objects from being added, the objects are added to the first category that accepts them.

To delete an object from the Object Library palette

1) In the Object Library palette, select the category where the object is stored.
2) Right-click the object and select Remove Object From Library.

Using the Group Properties dialog box, some categories may have been set up to prevent objects from being added, removed, or modified. You cannot delete an object if the category prevents it.

3) When you are prompted to delete the object, click Yes. The object is removed from the filesystem.

To rename an object in the Object Library palette

1) In the Object Library palette, select the category where the object is stored.
2) Right-click the object and select Object Info.

   Using the Group Properties dialog box, some categories may have been set up to prevent objects from being added, removed, or modified. You cannot delete an object if the category prevents it.
3) In the Name box, type a new name.
4) (Optional) In the Description box, type a description.
5) Click OK.

If another object of the same name exists in that category, Designer prompts you to confirm whether you want to replace the existing object.

To restore default objects in the Object Library palette

You can restore the default objects in the My Favorites, Standard, Barcodes, and Custom categories in Designer.

IMPORTANT: When you restore objects, all the objects provided with Designer are restored. Designer replaces any missing objects and overwrites all objects that have the same names as the originals.

1) Do one or both of these tasks:
   - To restore default objects in all categories at once, in the Object Library palette menu, select Restore Default Objects For All Groups.
   - To restore default objects in a single category, select one of the categories: My Favorites, Standard, Custom. or Barcodes. Then in the category menu, select Restore Default [*category name*] Objects.

To share an Object Library palette category with other form authors or form developers

You can place the Object Library palette objects in a shared folder or web folder where other form authors or form developers can access them.

1) Create a shared folder or web folder that other form authors or form developers can access.
2) Create subfolders for each of the categories in the Object Library palette you want to share.
3) Copy the objects from their existing location to the new location.
4) In Designer, select the category that you want to share in the Object Library palette.
5) In the Object Library palette menu, select Group Properties.
6) In the Location box, type or browse to the location of the new category subfolder that you created in step 2.
7) Click OK.
8) Repeat steps 4 to 7 for each of the categories.

To share an Object Library with other form authors or form developers

You can place your Object Library of categories and objects into a shared folder or web folder where other form authors or developers can access them.

To create a shared library by using an existing library

1) Add and remove categories and objects from your existing library until you create the structure you want to achieve for the shared library.
2) For each category, in the Object Library palette menu, select Group Properties and set the appropriate user permissions.
3) Locate the Objects folder in the directory where Designer is installed, and copy the Objects folder and all of its subfolders to a shared folder or web folder that form authors or developers can access.
4) Rename the LocalLibrary.xml file located in the Object folder to avoid confusion. For example, rename the file as SharedLibrary.xml.
5) Add the shared library to Designer by following the procedure *To add a shared library*.

To create a shared library without using an existing library

1) Create a shared folder or web folder that form authors or developers can access.
2) Create subfolders within the shared folder for each of the categories in the Library that you want to share.
3) Copy the Library objects from their existing location to the new shared location.

   TIP: To view the current location of objects or categories in the Library, click the category in the Object Library palette and select Group Properties from the c*ategory menu. Notice the folder location in the Location box.*
4) Using the form, create a new object library file, where *Category_name* is the name to appear as a category title in the Object Library palette, and *folder* is the location or relative path to the folder that contains the objects for the category.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectLibraryTabSet>
<tab name="Category_name" directory="folder" permission="adm"/>
<tab name="Category_name" directory="folder" permission="adm"/>
.
.
.
</objectLibraryTabSet>
```

5) Save the new object library file to the root of your shared folder with a .xml file name extension.
6) Add the shared library to Designer by following the procedure *To add a shared library*.

To add a shared library

1) In the Object Library palette menu, select Shared Library Location.
2) In the Location box, type the name of or browse to the shared folder that contains the object library file for the shared library you want to add.
3) Click OK.

Fragment Library palette

You can view the fragments in a fragment library as a list of fragment names or as a detailed view that includes the file names. You can also view a preview of the selected fragment. A preview of the selected fragment appears at the bottom of the Fragment Library palette.

Fragment libraries are folders in your file system. To create or delete a fragment library, you simply create or delete a folder. After you create a fragment library, you can open it in Designer.

To open a fragment library

1) In the Fragment Library palette menu, select Open Fragment Library.
2) Select the folder that corresponds to the fragment library.
3) Click OK.

To set the view in the Fragment Library palette

1) Set the view in the Fragment Library palette menu:
   – To view the fragments as a list, select View > List.
   – To view the fragments as a detailed list, select View > Details.

To preview the selected fragment

1) In the Fragment Library palette menu, select Show Preview Pane.

Fragment Library palette

You can view the fragments in a fragment library as a list of fragment names or as a detailed view that includes the file names. You can also view a preview of the selected fragment. A preview of the selected fragment appears at the bottom of the Fragment Library palette.

Fragment libraries are folders in your file system. To create or delete a fragment library, you simply create or delete a folder. After you create a fragment library, you can open it in Designer.

To open a fragment library

1) In the Fragment Library palette menu, select Open Fragment Library.
2) Select the folder that corresponds to the fragment library.
3) Click OK.

To set the view in the Fragment Library palette

1) Set the view in the Fragment Library palette menu:
   – To view the fragments as a list, select View > List.
   – To view the fragments as a detailed list, select View > Details.

To preview the selected fragment

1) In the Fragment Library palette menu, select Show Preview Pane.

2.7. Steps to creating a form design

Basic steps to creating a form design

Here are the basic steps to creating a form design using Designer:

- Plan the form design
- Create the form design
- Save the form design
- Preview and test the form design
- Publish the form design
- Distribute the form

Plan the form design

Keep these considerations in mind when planning a form design:

Data capture requirements

Consider the data you hope to gather or distribute and how you will capture and process it. Will users manually enter all data, or can any data be prepopulated in the form?

Calculations or scripts to associate with the data and where they will be run (client, server, or client and server)

Script will only run on a server if you are deploying forms through Forms and/or Output.

Security requirements.

Will users require a password for such things as opening, printing, copying text or applying signatures? See *Setting Security*, *Using password fields*, and *Using signature fields*.

Determining user needs

Your primary design objective should be to satisfy the needs of the people who use your forms. The clearer your goals, the better the form design.

Create a library of fragments

Consider creating fragments for logos, headers, address blocks, and footers that can be shared among form designs.

- Decide whether you will use the FormCalc or JavaScript language.
- Examine which security options you will apply to the form.
- Set accessibility standards, if needed.

Considering form usability

When planning a form, remember to build usability into the design. Here are a few points to consider:

- A form should be easy to use. A simple layout with clear, meaningful captions will make the form easier to understand. Forms can validate user input and supply prompts to assist users who enter data. If Forms is available, some of the information can be filled in ahead of time through merged data. If you are adept at scripting, you can also develop scripts to process or route data and forms automatically from the click of a button.
- A form should be accessible to users with disabilities.

  See About accessible forms and Tabbing order.

Create the form design

There are several ways you can create a form design. You can start with a blank form, template, or PDF document. If you are creating your first form design using Designer, try one of the tutorials or look at one of the sample forms installed with Designer.

Start with a sample form

If you prefer not to try one of the tutorials, you can start by looking at one of the sample forms installed with Designer. The sample forms illustrate form design techniques, from simple to complex. Designer includes a selection of complete samples, each one accompanied by a form design, sample data and/or schema, as well as the final version of the form. If one of the sample forms suits your requirements, use it as a starting point. For more information about sample forms, see “Sample Forms” in *Sample Form Snippets and Forms*.

You can start a form design based on a template and modify the design to suit your requirements. The Template Manager contains a variety of templates from which to choose.

Considerations when creating form designs

You will need to consider form-specific details when creating form designs. The following table lists items that you may want to keep a record of as you work on a form design.

Item	Specifications
Page size and orientation (for master page)	Page size (for example, Letter) Orientation (for example, Portrait)
Logos and graphics	Required file format (for example, TIF)

Item	Specifications
Form properties	Store title of form with form properties (Yes or No) Locale setting (for example, Viewer's system locale) Default scripting language (for example, JavaScript)
Accessibility	Tool tips required for objects (specify the objects requiring tool tips) Screen reader precedence (for example, Tool Tip)
Tabbing order	Setting (for example, Western)
Common font and drawing properties	Line thickness (for example, same thickness for all lines) All captions (same typeface and size) All data values (same typeface and size)
Object properties	Field borders and background color (Yes or No) Background color of fillable areas (for example, an RGB value) Field names (for example, see supplied "Field Name" list)
Form intelligence	Fields that must be filled (identify the fields) Prompts to specify for required fields (specify the required prompts)

RELATED LINKS:

Creating forms from Workbench

Creating forms based on atemplate, sample, or existing form

Form design layouts

Save the form design

Save your form design as you work.

If you are using Designer with Workbench, the form designs are saved in a AEM forms application to ensure that they are available to others who are logged in to the same server. If you are using the stand-alone version of Designer, form designs are saved using the current name and location.

RELATED LINKS:

Saving forms for Acrobat and Adobe Reader

Working with Data Sources

Preview and test the form design

Previewing and testing your form designs ensures that they look and behave the way you intended.

Preview your form design as you work by using the Preview PDF tab. When you preview a form design, Designer renders your form as a PDF file. See *To preview and test forms in the Preview PDF tab*.

You can test a form using sample data. Testing the form with sample data ensures that data and fields map and that repeating subforms appear as expected. You can automatically generate sample data to preview and test your form instead of creating a sample data file. If your form contains repeating

subforms or subform sets, you can specify the number of times the data will be repeated when you preview the form. See *To preview a form using sample data*.

You can also preview how your form will print by generating sample data to print with the form. If the form contains repeating subforms or subform sets, you can specify the number of times the data will be repeated when you print the form. See *To print a form with sample data*.

RELATED LINKS:

To set preview options for an interactive form

To set preview options for a non-interactive form

Publish the form

This feature is available only in the stand-alone version of Designer.

Publishing a form to a shared folder or web folder allows users or applications access to it. If the form design contains links to fragments, images, or other resources, the links are modified to reflect the new location of the file. Keep in mind that all the files in the form will be copied into a single folder.

If you are working outside of Workbench and want to publish to the repository, you can set up a web folder on your file system and map it to the location in the AEM forms repository.

RELATED LINKS:

Publishing to the AEM forms repository

To publish a form to a repository

Distribute the form

This feature is available only in the stand-alone version of Designer.

If you have Acrobat 8 or later, you can send PDF forms to multiple recipients by using the Acrobat Distribute wizard. The wizard is available from the File menu in Designer.

RELATED LINKS:

Distributing forms

3. Create Form Designs

3.1. Create, open, and save

Creating forms

You create a form design by dragging objects from the Object Library onto the Layout Editor. Alternatively, you can drag objects defined in an XML schema onto the Layout Editor.

The process for creating forms in Designer is different in the stand-alone and Workbench versions of Designer.

Creating forms with the stand-alone version of Designer

When you start Designer, you are prompted to create a new form, create a new form from a template, or open an existing form. You can also explore the sample forms. When you select the New Form or New From Template option, the New Form Assistant appears.

The New Form Assistant guides you through a series of steps where you choose the type of form design or template to create, how people will fill it, and how you will get the information back.

If you prefer, you have the option of making the New Form Assistant unavailable.

If you have Acrobat 8 or later, you can send PDF forms to multiple recipients by using the Acrobat Distribute wizard.The wizard is available from the File menu in Designer. Sending the form by using the wizard certifies the form originator's identity to form recipients and encrypts the data that the recipients submit when they return a filled form. It also adds usage rights to the form so that form fillers can save the form in Adobe Reader.

To create a form design using the New Form Assistant

1) Select File > New. The New Form Assistant appears.
2) In the Getting Started panel, select a method for creating the new form, and then click Next.
3) Follow the on-screen instructions to create a form.
4) Name and save the form.

To create a form design without the New Form Assistant

1) Select File > New. The New Form Assistant appears.
2) Click the Do Not Use Assistant link. The Assistant Options dialog box appears.
3) Set the options for using the New Form Assistant and click OK. The New dialog box appears.

4) In the Standard tab, click an icon to select a page size, and then click OK. An untitled form design is created.

   TIP: *To view the whole page, the actual size of the page, or an area equivalent to the width of the page, click Actual Size ,Fit Page , or Fit Width on the toolbar. The current zoom setting is also displayed in the toolbar.*

To start with a blank form

1) Click the New button on the toolbar.

Creating forms from Workbench

In Workbench, use the Form Design perspective to create and manage XDP and PDF forms for a AEM forms application. When you create a form, it opens in Designer, where you can lay out the form design and add objects. Workbench displays a tab with an image of the first page of the form design, but all form editing is done in Designer.

When you create a form design, you can specify a data model, a data submission method, the Acrobat or Adobe Reader version, and the form type. You can also select Adobe Reader features for the form, such as commenting, digital signatures, and encryption.

1) Create a form design from either Designer or Workbench:
   - In Designer, click File > New.
   - In Workbench, select File > New > Form.
2) Follow the onscreen instructions and then click Finish.

Creating forms based on a template, sample, or existing form

You can create forms based on a template, sample, or existing form in both the stand-alone version of Designer and Designer with Workbench.

1) Open the template, sample, or existing form on which you want to base the form design.
2) Select File > Save As.
3) Browse to the location in which to save the form design.
4) In the File Name box, type a file name for the form design.

   NOTE: *If you create a form using a template that was created in a previous version of Designer, you can use the Compatibility tab (Form Properties dialog box) to update it to version 7.*

Using forms as a model for Guides

NOTE: *Guides is deprecated.*You can use Designer PDF or XDP form designs as the model for creating Guides in Workbench. If you have an existing form guide from an earlier release of AEM forms, you can open it in Guide Builder and your existing Guide definition is imported into Guide Builder. For more information, see Guides based on XDP or PDF forms section in Workbench Help.

NOTE: You cannot create a Guide based on a form design created by importing a PDF file as artwork.

Creating forms by importing a PDF

You can import a PDF document to use as the basis for creating a new form design. For more information, see *Importing PDF files*.

You can create forms by importing a PDF document in both the stand-alone version of Designer and Designer with Workbench.

If you have PDF forms that were created in Acrobat or another application, you can import them as background artwork. After you import the PDF forms, you can add objects to make them interactive.

If you are creating a PDF form design, you can select the Acrobat or Adobe Reader target version. With the target version chosen, you can see when an object is not appropriate for that version.

RELATED LINKS:

Creating and managing templates

Import documents

About forms

Guidelines for forms

Considerations for creating forms for use with Acrobat and Adobe Reader

Keep these considerations in mind when creating forms to be opened and filled in Acrobat or Adobe Reader:

- Consider which version of Acrobat and Adobe Reader people are using when filling the form. Setting the target version of Adobe Reader and Acrobat in Designer can help you create a form design that is compatible. See *Selecting the Acrobat and Adobe Reader target version*.
- Acrobat supports full interactive functionality, including form filling, local saving, digital signatures, review and markup, database and web service calls, and printing.
- Adobe Reader supports a smaller set of functionalities and provides users with the ability to fill and print forms, and in some situations, submit them. To extend the functionality of Adobe Reader to match that of Acrobat, you must set usage rights for the form using extensions. For more information, see *Using PDF security options*.
-
- Acrobat and Adobe Reader support client-side data exchange processing for interactive PDF forms.
-
- Server-side data processing, such as dynamic rendering and redisplay of forms at run time, is available through Forms. For more information see *Designing forms for Forms*.

RELATED LINKS:

Guidelines for forms

Distributing forms

Adding details about the form

You can add descriptive information about a form design or template (for example, file description and author's name) in the form design or template. If you want to save the information as metadata with the PDF form, select Enable Plaintext Metadata in Form Properties > PDF Security tab. When the metadata is saved, it is available to other Adobe applications, such as Acrobat and search utilities. In Acrobat, the metadata appears on the Description tab of the Document Properties dialog box.

You can also add custom properties, such as copyright URLs, to a form.

In Acrobat, the metadata appears in the Custom tab of the Document Properties dialog box.

Keep in mind that certain case-sensitive keywords are reserved for property names:

- Title
- CreationDate
- Creator
- Producer
- Subject
- Keywords
- ModDate
- Trapped
- Author

To add information about a form

1) Select File > Form Properties.
2) Enter the information on the Info tab. For a complete list and description of the items on this tab, see *Info (Form Properties dialog box)*.
3) Embed the information for other applications to use:
   – Click the PDF Security tab.
   – In the Permissions area, select Use A Password To Restrict Printing And Editing Of The Document And Its Security Settings, and then select the required security settings.

To add custom properties

1) Select File > Form Properties.
2) In the Custom Property box on the Info tab, click the Add button.
3) Type the name of the property and press Enter.
4) Type the value of the property and press Enter.
5) Repeat steps 4 and 5 for each property you want to add.
6) For a complete list and description of the items in this tab, see *Info (Form Properties dialog box)*.
7) Embed the information for other applications to use:
   – Click the PDF Security tab.

– In the Permissions area, select Use A Password To Restrict Printing And Editing Of The Document And Its Security Settings, and then select the required security settings.

To remove custom properties

1) Select File > Form Properties.
2) In the Custom Property box on the Info tab, select the property, and click the Remove button.

To add document title to the PDF title bar

By default, the PDF output of a document displays the file name in the Adobe Reader or Adobe Acrobat title window. You can add the document title of static XFA forms to the PDF title bar. If a user opens the PDF, the document title will display on the title bar of the window.

1) To open the Xml source for a document, click View -> XML Source.

   Verify if the following tag is available in the Xml:

   ```
   <?originalXFAVersion http://www.xfa.org/schema/xfa-template/3.3/ ?>
   ```

2) If the tag exists, append `DisplayDocTitle:1?` to the tag. The resultant tag should be:

   ```
   <?originalXFAVersion http://www.xfa.org/schema/xfa-template/3.3/
   DisplayDocTitle:1?>
   ```

   If the tag does not exist, add the tag with the `DisplayDocTitle:1?` append.

   ```
   <?originalXFAVersion http://www.xfa.org/schema/xfa-template/3.3/
   DisplayDocTitle:1?>
   ```

   NOTE: You need to add this tag along with the other processing instructions. in the following parent node:

   ```
   <template xmlns="http://www.xfa.org/schema/xfa-template/3.6/">
   ```

3) Save the document.

To set user permissions on a PDF form

You can set user permissions for accessing, printing, and editing PDF forms output through Designer.

1) Select File > Form Properties.
2) In the PDF Security tab, select Use A Password To Restrict Printing And Editing Of The Document And Its Security Settings, and then select the required security settings.

RELATED LINKS:

Using PDF security options

Saving forms

When you save a form, the file format must be compatible with the way the form will be used in the user community. You can save a form as a PDF file or an XDP file. Each file type has different uses:

Adobe Static PDF Form (*.pdf)

Saves forms as static PDF, based on the Acrobat and Adobe Reader target version specified. Static PDF forms render once and are displayed on the client in the Acrobat or Adobe Reader target version. They are not rerendered in response to user interaction. The PDF form may have been designed with a flowable layout; however, when the static PDF form is created, its layout is fixed and the resulting PDF form will not rerender on the client. Static PDF forms can be interactive or non-interactive.

Adobe Dynamic XML Form (*.pdf)

Saves forms as dynamic PDF, based on the Acrobat and Adobe Reader target version. The form design can contain dynamic elements. Dynamic PDF forms render on the client in Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of an object are possible in Adobe Reader because Adobe Reader has enough information to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If the end user clicks a button that adds a new row to a table, the form is rerendered in Adobe Reader.

Adobe XML Form File (*.xdp)

Sets the default file type for new forms to the native XML-based file format created by Designer. Use this option if you will be using Forms.

Designer Template (*.tds)

Saves the basic structure for a form as a template. It can contain components and settings, such as fonts, page layout, formatting, and scripts. Use it as a starting point for a new form.

When Designer is integrated with Workbench, files are saved in the Workbench folder on your local system:

- If you are using Windows® XP, the Workbench folder is located in \Documents and Settings\<user name>.
- If you are using Windows Vista®, the Workbench folder is located in \Desktop\<user name>.

After you save a form, check in the form and any referenced files in Workbench.

To save a form design

You can save form designs by using their current name and location, or save a copy by using a different name or location.

When Designer is integrated with Workbench, form designs are saved in an application in Workbench. Saving form designs in an application ensures that they are available to others who are logged on to the same server. When you save a form design in Designer the corresponding image of the form in Workbench is updated. After you save a form, check in the form and any referenced files, such as fragments or images, in Workbench.

You can save a form design in several formats. PDF forms saved in Designer should only be edited in Designer.

1) To save a form design, do one of the following actions:
   – Select File > Save.
   – In the toolbar, click Save .

   NOTE: *If you are using Designer with Workbench, check in the form and any referenced files in Workbench.*

To check in a form in Workbench

1) In Designer, in the toolbar, click the Switch To Workbench button.
2) In Workbench, open the Applications view.
3) Navigate to the form you saved, right-click the form and select Check-in.

To save a copy of a form design

1) Select File > Form Properties.
2) Click the Save Options tab and set the save options:
   – To embed accessibility information within a PDF form file, select Generate Accessibility Information (Tags) For Acrobat.
   – To create a log file of the saved PDF form file, select Generate Log File When Saving. The log file is placed in the same directory as the saved form.
   – To embed the form fonts in the saved PDF form file, select Embed Fonts.
3) Click OK.
4) Select File > Save As.
5) To make a copy of the form design under a different name, browse to the location in which to save the form design. In the File Name box, type a different filename.
6) To change the file type, from the Save As Type list, select one of the options.

   NOTE: *Notice the message above the Save Options area that indicates the Acrobat and AdobeReader target version for the form. You can change the target version by selecting File > Form Properties > Defaults > Target Version > Choose Version To Run Form In.*
7) Click Save.

   NOTE: *Sample form designs and templates are provided. To use them, see the associated readme files. The sample form designs are installed in the EN\Samples folder of the Designer installation folder. The sample templates are installed in the Templates folder of the Designer installation folder. Templates can be viewed and managed through the Template Manager.*

RELATED LINKS:

Import documents

Adding details about the form

Selecting the Acrobat and Adobe Reader target version

Opening forms

In addition to opening form designs (XDP), templates (TDS), or PDF forms created and modified using only Designer, you can open other files of different formats. These formats are described in detail in *Import documents*.

NOTE: PDF forms created in a program other than Designer must be imported.

When you open a file, either a form design, template, or a file of another format, Designer attempts to match fonts in the file with fonts that are available on your computer. If the file contains an unavailable font, Designer displays a dialog box showing the missing font and a suggested replacement. You can accept the replacement font or change it. Later, when you save the file, you can indicate whether you want the replacement fonts saved with the file.

If you open a form that was created in a previous version of Designer, you can use the Compatibility tab (Form Properties dialog box) to update it to the current version.

You can open forms in Designer or in Workbench. To open a form in Workbench, in the Applications view, double-click the form. When you open a form from Workbench, the form opens in Designer. In Workbench, a corresponding tab is displayed with an image of the first page of the form.

If you are using Workbench, you might need to synchronize files, such as images or fragments, from Workbench to your local system. In Workbench, forms and fragments are managed in the Applications view, and images are managed in the Resources view. To synchronize a file, right-click the file, and select Synchronize.

1) Select File > Open.
2) Browse to the location or Workbench application where the form design is stored, select the file, and click Open. If the file contains fonts that are not available on your computer, the Missing Fonts dialog box appears.
3) If necessary, select a font substitution method:
   – To accept the suggested replacement fonts, click OK.
   – To change the replacement font, click Change Substitute Font. Select a font and font style and then click OK. In the Missing Fonts dialog box, click OK.

   NOTE: *This font substitution is not permanent. Designer must map the unavailable fonts every time the form opens. To make the font substitution permanent, before saving the file, select File > Form Properties > Save Options and then selecting Embed Fonts.*

Selecting the Acrobat and Adobe Reader target version

You can design a form to run on a specific target version of Acrobat and Adobe Reader. Designer displays warning messages when you try to use a feature that the selected target version does not support.

When you specify a target version of Acrobat and Adobe Reader, you must ensure that you have the correct version of Acrobat installed that matches the target version saved; otherwise, you may get an error or warning message when you click the Preview PDF tab. The target version you select is also displayed at the bottom of the Save As dialog box when you save a form.

The preview will use the same format as the saved form. You can right-click the Preview PDF tab to see which format is currently selected. To indicate the format of an unsaved form, you must change the Tools > Options > Document Handling > Default File Type For New setting.

1) Select File > Form Properties.
2) Click the Defaults tab and, in the Choose Version To Run Form In list, select the target version of Acrobat and Adobe Reader.
3) Click OK.

RELATED LINKS:

Defaults (Form Properties dialog box)

Saving forms for Acrobat and Adobe Reader

When you create forms for Acrobat and Adobe Reader, you must save them as PDF files. PDF files saved from Designer should only be edited using Designer.

You can select the target version of Adobe Reader or Acrobat that you plan to run the form on. Selecting the target version allows you to see features that are not supported. See *Selecting theAcrobatandAdobeReadertarget version*.

NOTE: Ensu re that you have the version of Acrobat installed that matches the target version saved; otherwise, you may get an error or warning message when you click the Preview PDF tab.

1) Select File > Save As.
2) In the File Name box, type the name for the file.
3) In the Save As Type, select the required type.
4) Click OK.

RELATED LINKS:

Saving forms

Addressing warning messages in the Report palette

Target version warning messages

Distributing forms

This feature is available only in the stand-alone version of Designer.

If you have Acrobat 8 or later, you can send PDF forms to multiple recipients by using the Acrobat Distribute wizard. The wizard is available from the File menu in Designer.

The Acrobat Distribute wizard is designed for form authors who want to distribute forms and collect the form data locally on their computers. When you select File > Distribute Form, the Distribute Forms wizard opens and walks you trough a few step-by-step panes to prepare a form for distribution. Sending the form by using the wizard certifies the form originator's identity to form recipients and encrypts the data that the recipients submit when they return a filled form. It also adds usage rights to the form so that form fillers can save the form in Adobe Reader. When you close a form design in Designer, you can choose to distribute the form (if you started the form design process in Acrobat).

To distribute a form using Designer

1) Save the form as a PDF file.
2) To distribute the form, select File> Distribute Form.

Publishing forms

This feature is available only in the stand-alone version of Designer.

You can publish your form designs to a shared or web folder where many users or other applications can access them. If a form design contains links to external files, the links are modified to reflect the new location of the file. You need write access to the folders to which you publish. Your network administrator can set up the permissions you require.

RELATED LINKS:

Saving forms

Publishing to the AEM forms repository

If you are using the stand-alone version of Designer and want to use resources, such as images and fragments, that reside in the AEM forms repository, you can set up a web folder on your file system and map it to the repository. When you publish the form, links to the resources are maintained. You use the Publish to Repository command to copy the files into the AEM forms repository.

Alternatively, you can create the form design on your file system. When you complete the form design, you can drag and drop the folder and any sub-folders into the appropriate folder in the Workbench Resources view.

You need access to AEM forms to read and write files in the AEM forms repository.Your network administrator can set up the permissions your require.

To publish a form to a repository

You can put a copy of a form and its linked files in a shared or web folder where it can be accessed by many users or other applications. When you publish a form to a repository, changes to file names, file types, and references to external files (images) occur only in the published copy. If the form contains a link to an external file, when the form and files are published, the list in the form is modified to reflect the new location of the file.

You can publish the form in PDF or in XDP format.

- When you publish the form in PDF, information such as images and schemas is embedded in the form, and no external files are copied to the publish folder.
- When you publish the form in XDP format, the links to externally referenced files are modified to be a relative path. The relative path lets you have one location for images that multiple forms can reference without having to have multiple copies of the images.

If the externally linked files are not contained in a folder below the shared or web folder, the files are copied to the publish folder and the links in the XDP file are adjusted to reflect the new absolute path location of the files.

When publishing a form to the AEM forms repository, all external files are copied to the web folder.

To publish a form to the repository

1) Select File > Publish to Repository.
2) (Optional) To change the file type, select a new file type in the Save As Type list.
3) Click Save.

Before you publish a form to a repository, you can optionally specify additional save options.

To set save options prior to publishing a form:

1) Select File > Form Properties > Save Options and do one or more of the following:
   – To embed accessibility information in a PDF form file, select Generate Accessibility Information (Tags) For Acrobat.
   – To create a log file of the saved PDF form file, select Generate Log File When Saving. The log file is placed in the same directory as the saved form.
   – To embed the form fonts in the saved PDF form file, select Embed Fonts.
2) Click OK.

Publishing a form to a repository

You can put a copy of a form and its linked files in a shared or web folder where it can be accessed by many users or other applications. When you publish a form to a repository, changes to file names, file types, and references to external files (images) occur only in the published copy. If the form contains a link to an external file, when the form and files are published, the list in the form is modified to reflect the new location of the file.

You can publish the form in PDF or in XDP format.

- When you publish the form in PDF, information such as images and schemas is embedded in the form, and no external files are copied to the publish folder.
- When you publish the form in XDP format, the links to externally referenced files are modified to be a relative path. The relative path lets you have one location for images that multiple forms can reference without having to have multiple copies of the images.

If the externally linked files are not contained in a folder below the shared or web folder, the files are copied to the publish folder and the links in the XDP file are adjusted to reflect the new absolute path location of the files.

Creating and managing templates

A template provides the basic structure for the form design and contains predefined file-creation information that Designer applies to the form design. Templates contain definitions of objects used in a form design as well as the form design's page layout. A template can store any properties that you would use in more than one form design, including these:

- Master page layouts, which influence the format of pages

- Boilerplate objects, including formatted headers and footers
- Embedded images or objects that link to frequently used graphics, such as logos or watermarks
- Any text variables or floating fields that you use as placeholders for data
- Predefined formula equations and script fragments
- File properties and file creation information
- Other properties that facilitate or automate your work

RELATED LINKS:

About forms

Creating forms based on a template, sample, or existing form

Creating a template

Creating a template is very much like creating a form design. However, in a template, you design the layout and select the file settings and preferences that you would typically need in more than one form design. You can optionally include and define any objects that could be applied to more than one form design. Including layouts and properties that are common to many form designs saves you time because less work is needed to prepare form designs that are based on the template.

One of the easiest way to create a template (TDS) file is by using the New Form Assistant. The New Form Assistant guides you through the steps involved in creating a template, such as previewing and selecting the template you want to use, entering business contact information, and adding email submit and print buttons. However, if you prefer not to use the New Form Assistant, you can select Tools > Template Manager to open the Template Manager dialog box where you can directly select the template you want.

To create a template in the stand-alone version of Designer

1) Open the New Form Assistant by using one of these methods:
   - Select File > New.
   - In the toolbar, click the Down Arrow next to the New button and select New.
2) In the New Form Assistant, select Based on a Template and click Next.
3) Follow the onscreen instructions to customize the template to suit your needs.
4) Select File > Save As.
5) In the Save As Type list, select Adobe Designer Template (*.tds).
6) Click Save.

To create a template with Designer and Workbench

1) In Designer, select File > New.
2) Follow the onscreen instructions until the New Form Assistant opens in Designer, and then select Based on a Template.
3) Follow the onscreen instructions to customize the template to suit your needs.
4) Select File > Save As.
5) In the Save As Type list, select Adobe Designer Template (*.tds).

6) Click Save.

Managing templates

The Template Manager contains a variety of sample templates that are provided with Designer. It can also contain any templates you create and use with Designer.

Use the Template Manager to do these tasks:

- Preview templates
- Select a default template for creating form designs
- Add, delete, and rename templates, as well as add and remove groups

In the Template Manager, you can also add or delete tabs, move templates between tabs, and determine where in the file system Designer stores templates. Each tab corresponds to a folder in the file system. By default, template files are stored in the Designer Templates folder.

By default, templates are stored in the Designer Templates folder (\EN\Templates).

Each time you open Designer, the application copies the template files in the Templates folder to the user profile for each language. This could be an issue in certain multiuser environments where disk quotas are restricted. To save disk space, set up a common directory for templates. For further information about setting up a common directory for templates, see the technical note at http://kb2.adobe.com/cps/524/cpsid_52464.html.

To add a template to the Template Manager

1) Select Tools > Template Manager.
2) Click the tab in which to add the template.
3) In the palette menu, select Add Template.
4) Browse to the appropriate folder, select the template file (TDS) you want to add, and click Open. A copy of the template file is added to the Template Manager.

To delete a template from the Template Manager

1) Select Tools > Template Manager.
2) Click the tab where the template is stored.
3) Right-click the template and select Delete Template.
4) When you are prompted to delete the template, click Yes. The file is removed from your computer's file system.

To rename a template in the Template Manager

1) Select Tools > Template Manager.
2) Click the tab where the template is stored.
3) Right-click the template and select Rename Template.
4) Type a different name for the template, and press Enter.

To specify a default template in the Template Manager

1) Select Tools > Template Manager.
2) Click the tab where the template is stored.
3) Select the template.
4) Click Set Selected as Default.

To restore the default templates in the Template Manager

You can restore the default templates that are available in the Blank tab and the Common Forms tab in the Template Manager dialog box.

IMPORTANT: *When you restore templates, all of the templates that were provided with Designer in the Blank tab and the Forms tab are restored. Designer replaces any missing templates and overwrites all templates with the same names as the originals.*

1) Select Tools > Template Manager.
2) Click either the Blank tab or the Forms tab.
3) In the palette menu, select Restore Default Templates.

To add a tab to the Template Manager

When you add a tab to the Template Manager, a subfolder with the same name is created in the Designer Templates folder.

1) Select Tools > Template Manager.
2) In the palette menu, select Add Category.
3) Type a name for the tab and click OK.

To delete a tab from the Template Manager

When you delete a tab from the Template Manager the corresponding folder is removed from the Designer Templates folder.

1) Select Tools > Template Manager.
2) Click the tab you want to delete.
3) In the palette menu, select Remove Category.
4) When you are prompted to remove the tab, do one of these actions:
   - To delete the selected tab and its associated category folder and templates, select Remove The Category And Delete All Contained Templates.
   - To move the templates to a different tab before deleting the selected tab and its associated category folder, select Remove The Category And Move The Contained Templates To The Following Category: *[tab name]*, and then click the tab where you want to move the templates.
5) Click OK.

To move a template to a different tab in the Template Manager

1) Select Tools > Template Manager.
2) Click the tab you want, right-click the template that you want to move, and select Move Template To > *[tab name]*.

To determine where the template files are stored

1) Select Tools > Template Manager.
2) Click the tab you want.
3) In the palette menu, select Category Properties. The Template Category Properties box displays the location of the templates that are currently listed in the selected tab.

To customize the view in the Template Manager

1) Select Tools > Template Manager.
2) Click the tab you want.
3) In the palette menu, select View and perform one of these actions:
   – To display the template names as icons, select Large Icons.
   – To display the template names in a list, select List.

Importing a template

You can import any template file (TDS) into Designer and copy the template into the Template Manager, create a new form based on the template, or edit the template.

When you import a template, a copy is automatically added either to the templates folder that is indicated in the form's XML source code or to the folder named Other, if no location is specified. The imported template appears in the Template Manager, listed in the corresponding tab. If a template with the same name already exists in the tab, Designer displays a message prompting you to replace the existing file.

1) Open the template file you want to import by using one of these methods:
   – Select File > Open, browse to the folder you want, select the template file (TDS), and click Open.
   – In Windows Explorer, double-click the template file (TDS).
   – Drag the template file (TDS) from Windows Explorer into the Designer workspace.
2) In the Template Options dialog box, select one of these options:
   – To add the template to the Template Manager, select Copy This Template Into The Template Manager.
   – To create a PDF form based on the template, select Create A New Form Based On This Template.
   – To open the template in Designer for editing, select Edit This Template.
3) Click OK.

Creating and customizing a form based on a template

Template files (TDS) sometimes contain customizable text or images within field captions, text objects, and static images. When you create a form based on a template that contains fields with customizable text or images, the New Form Assistant opens enabling you to change the information displayed in each field to suit your own needs. For example, the sample templates available in the Template Manager contain customizable text and images, which you can replace with your company's name, address, telephone number, and logo.

The information that you enter for each field is saved and reused to repopulate the same fields in the New Form Assistant the next time you open the template.

To create a form based on a template in the stand-alone version of Designer

1) Open the New Form Assistant by using one of these methods:
   – Select File > New.
   – In the toolbar, click the down arrow next to the New button and select New.
2) In the New Form Assistant, select Based on a Template and click Next.
3) Follow the onscreen instructions to customize the new form to suit your needs.
4) Name and save the file.

To create a form based on a template with Designer and Workbench

1) In Designer, select File > New.
2) Follow the onscreen instructions until the New Form Assistant opens in Designer, and then select Based on a Template and click Next.
3) Follow the onscreen instructions to customize the new form to suit your needs.
4) Name and save the file.

3.2. Page layout

Setting up pages

Pages represent the canvas on which you build your form design. To lay out a form design, you drag objects, such as Text Fields, from the Object Library palette onto the page. Use the Design View tab to view, add, delete, and edit pages.

To show or hide a page

You can show or hide pages in a form design by using different methods:

- To show the Design View tab if another tab is active, in the Layout Editor, click the Design View tab.
- To show or hide the Design View tab, select View > Design View.

To add a page

You can add a page by using the Insert menu or Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- To add a page, click the Design View tab and select Insert > New Page.
- To add a page in the Hierarchy palette, right-click the required page-level subform and select New Page.

To delete a page

You can delete a page by using the Insert menu or Hierarchy palette.

TIP: *if the Hierarchy palette is not visible, select Window > Hierarchy.*

- To delete a page, click anywhere on the page and select Edit > Delete Page.
- To delete a page in the Hierarchy palette, right-click the page-level subform that corresponds to the page and select Delete.

To move between multiple pages

If your form design has multiple pages, you can easily move from one to another by using the Layout Editor or Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- In the Design View tab, scroll up or down to the required page.
- In the Hierarchy palette, click the required page.

To reorder pages

If you need to change the order of the pages, you can use the Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

1) In the Hierarchy palette, select the subform that corresponds to the page you want to move.
2) Drag the subform to a new position below the “form1” node.
3) Repeat steps 1 and 2 until all the pages are in the order you prefer.

To rename a page

You can change the name of a page by using the Hierarchy palette or the Binding tab of the Object palette. The default name of a page node in the Hierarchy palette is “(untitled Subform) (page 1)”, “(untitled Subform) (page 2)”, and so on.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- In the Hierarchy palette, right-click the page node and select Rename Object. Type a new name for the page.
- With the Design View tab selected, click anywhere on the page, click the Object palette and, in the Binding tab, type a new name for the page.

Using master pages

Master pages allow you add objects that will appear in the same position throughout the form design. They are useful when you want to adjust the size and position of content areas, add page numbering, and create single- or double-sided features (such as headers and footers).

A typical form design that has two pages with different dimensions contains at least two master pages, one for each page. If a form design has multiple pages that have the same dimensions and orientation, the form design needs only a single master page.

You can start designing pages immediately with the default master page. If your design is more complex (for example, you need to create design components such as page headers and footers), it is best to work with those components directly on the master page.

You can place objects anywhere on a master page. Each master page can have a unique design, which you can apply to one or more pages. Master pages save you the effort of creating the same layout directly on individual pages, one page at a time. If several pages that have the same layout also need the same set of objects (for example, a standard footer), placing and maintaining one set of those objects on a master page is more efficient than duplicating the objects on every page. When the objects are placed on a master page, all associated pages display those objects in the background. Changing the master page after automatically updates all associated pages.

Use the Pagination tab to indicate whether the master page is used in odd (front), even (back), or blank pages and its placement in the page set. You can also choose whether pages that are rendered according to the selected master page should contribute to the total page count.

RELATED LINKS:

Headers, footers, and other background items

Applying master pages to forms whose number of pages vary

Master page properties in the Master Page tab

Master page properties in the Pagination tab

To use master pages with repeating subforms

You can assign a different master page for any pages that occur after the first page in a form if you have a repeating subform that does not fit on one page but requires subsequent pages. For example, the subform may begin in the middle of the first page in the form. To have the subform begin at the top of the pages that occur after the first page, you create a different master page.

When to add a master page

Normally, you add additional master pages when your require two or more different page sizes, a different page orientation, or different print mode settings. Form filling is not supported on Master page fields. It is recommended to use master page to show only static and read-only content. Use normal pages for data capture fields.

If a form has a fixed layout, the number of pages in the form is also fixed. If the form contains subforms that adjust to accommodate the amount of data being merged, Forms adds additional pages automatically if the amount of data to be merged cannot completely fit on a single page.

NOTE: When you create a form design for a form that has a fixed layout that accepts merged data, you must ensure that filled objects do not expand to the point that they overrun the content area. Otherwise, unwanted pages may get added to the form automatically when the form is rendered.

A form design may contain more than one content area. For example, a master page may have two content areas to provide a two-column layout, and two master pages of this type (one odd and one even) may be needed to support double-sided pages.

To show or hide a master page

You can show or hide a master page by using different methods.

- To show the Master Pages tab if another tab is active, in the Layout Editor, click the Master Pages tab.
- To show or hide the Master Pages tab, select View > Master Pages.

To add a master page

You can add a master page by using the Insert menu or Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- To add a master page, click the Master Pages tab and select Insert > New Master Page.
- To add a master page in the Hierarchy palette, right-click the Master Pages node and select New Master Page.

To delete a master page

You can delete a master page by using the Edit menu or Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- To delete a master page, click the Master Pages tab, select the content area that corresponds to that master page and select Edit > Delete Master Page.
- To delete a master page in the Hierarchy palette, right-click the master page and select Delete.

To move between multiple master pages

If your form design has multiple master pages, you can easily move from one to another by using the Layout Editor or Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- In the Master Pages tab, scroll up or down to the required page.
- In the Hierarchy palette, click the required master page.

To reorder master pages

If you need to change the order of the master pages in a form design, you can use the Hierarchy palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

1) In the Hierarchy palette, drag the master page to a new position below the Master Pages node.
2) Repeat step 1 until all of the master pages are in the order you prefer.

To rename a master page

You can change the name of a master page by using the Hierarchy palette or the Master Page tab of the Object palette.

TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

- In the Hierarchy palette, right-click the master page, select Rename Object, and type a new name for the master page.
- With the Master Pages tab selected, click anywhere on the master page, select the Object palette and, in the Master Page tab, type a new name for the master page.

To add page numbering

In the Master Pages tab, you can add an object that displays the current page and total page count of the form, for example, Page 1 of 30.

1) In the Object Library palette, click the Custom tab category.
2) Click the Page n of m object, and then drag it onto the form design.

To specify the size of a master page

In the Master Pages tab, you can view changes to a form's basic page layout, including the page size. When you select a master page in the Hierarchy palette, you can change its basic page layout properties in the Master Page tab of the Object palette.

TIP: *Content areas define the region in which objects can be placed on the pages of a form. If you are creating a form that contains subforms set to flow content, it is best to define the size and position of all content areas before you add other objects to the master page.*

1) In the Hierarchy palette, under the Master Pages node, select the master page you want to resize.
2) In the Object palette, select a page size from the Paper Type list.
3) If you selected the Custom type, type the page dimensions in the Height and Width boxes.

To specify the orientation of a master page

In the Master Pages tab, you can view changes to a form's basic page layout, including the orientation. When you select a master page in the Hierarchy palette, you can change its basic page layout properties in the Master Page tab of the Object palette.

1) In the Hierarchy palette, under the Master Pages node, select the master page you want to orient.
2) In the Object palette, select Portrait or Landscape.

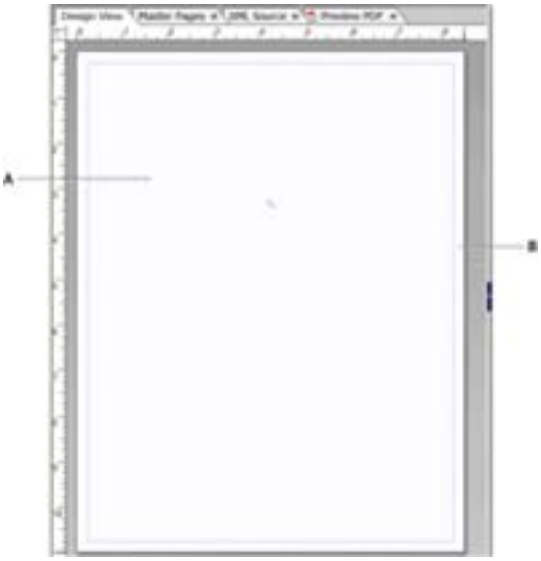
   NOTE: Set the page orientation explicitly as portrait or landscape for the Custom Paper Type. If the form width is greater than the form height, the orientation does not change to landscape automatically in when Custom Paper Type is selected.

Specifying white space around the edges of a form

In a traditional printed document, *margin* refers to an area around the outside edges of the page where no printing occurs. In interactive forms, the term *margin* refers only to the buffer of white space around the inside edges of an object's border.

To increase the amount of white space around the outside edges of a form, you must reduce the size of the content area on the master page. By default, the border of the content area is 0.25 inches from the edges of the form. To create extra white space in which to place a header or footer or to make adjustments to support printer limitations, you must make the default content area smaller.

The area bounded by the content area determines where objects can be placed on the pages of a form. The default subform on a page is the same size as the default content area on a master page, and the margin settings of the default subform are set to 0 inches. Changing the margin settings of the default subform on a page also influences where objects are positioned. Increasing the subform's margin settings decreases the area where the subform's objects may be rendered.



A.

Objects can be placed on pages inside the area bounded by a content area. The border represents the edge of the content area.

B.

This area external to the content area represents white space on the page.

All of the objects that you place on a master page are displayed on each associated page, regardless of whether you place those objects inside or outside the content area. If you are setting up a watermark, you place the objects that make up the watermark inside the content area.

If you are designing a form that contains subforms that flow content, and you do not want the objects on the master page to interfere with objects that are placed on pages, you must position the master page objects outside the content area; that is, somewhere on the white space that you create on the master page. If you place objects inside the content area, other objects may be placed on top of the master page objects when the form is rendered.

RELATED LINKS:

Using content areas

About subforms

Copying, moving, and resizing content areas

Headers, footers, and other background items

If most of the pages in your form design require a certain amount of standardized, fixed content, you can place that content on master pages to provide a consistent background and make editing easier. For example, you can arrange text, images, and geometric shapes on a master page to have them appear in the same location on multiple pages.

If you create headers and footers and arrange other background items on the master page, follow these general guidelines:

- If you are creating a form that contains subforms set to flow content, it is best to define the size and position of all content areas before you add additional objects to the master page.
- Objects can be placed anywhere on a master page, inside or outside the content area. As a general rule, do not place objects inside the content area unless you intend to have other objects placed on top of the master page objects when the form is rendered. If you are setting up a watermark, place it inside the content area.
- You will need text objects to hold read-only text, image or image field objects to position images (such as a logo), and you may consider adding geometric objects such as circles, lines, or rectangles to improve the appearance of the form design.
- All of the objects on a master page are displayed as background objects on pages when the form is rendered.

RELATED LINKS:

Copying, moving, and resizing content areas

Assigning additional master pages to complex form designs

Master pages provide the underlying format and background for pages. All new form designs are created with a default master page, which is applied to the first page. Additional pages are formatted according to the layout of the default master page unless you create a different master page to use.

A form design can contain any number of master pages, each having a unique layout. If some of the pages need different layouts or boilerplate objects, you can create a master page for each unique layout.

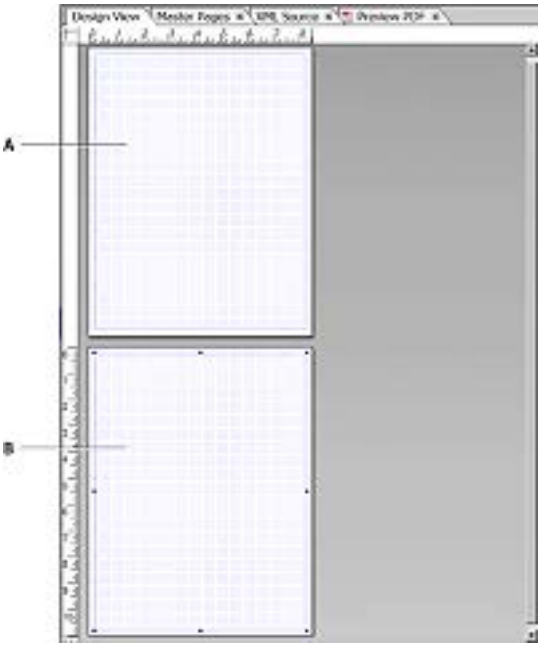
To assign a master page to a specific page

1) Select the subform by using one of these methods:
   - In the Design View tab, select the default subform on the page.
   - In the Hierarchy palette, select the subform that corresponds to the page.
2) In the Object palette, click the Pagination tab and, in the Place list, select On Page > *[page]*, where *[page]* is the name of the master page you want use.

Example of assigning a second master page to a form design

The following example provides steps for assigning a second master page to a page in an interactive form design or a form design that has a fixed layout.

This form design contains two pages and two master pages. The first master page has a portrait orientation and the second master page has a landscape orientation.

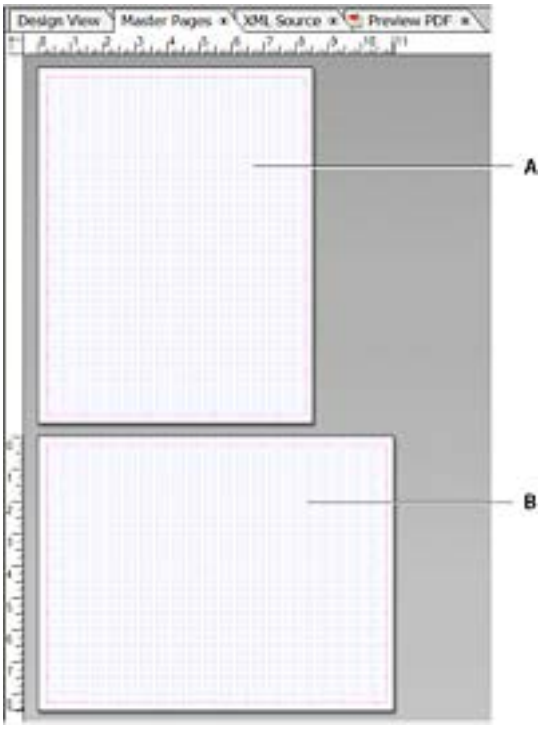


A.

Page 1

B.

Page 2



A.

Master page 1:Portrait

B.

Master page 2:Landscape

1) Click the Design View tab.
2) Select the subform on page 2.
3) In the Object palette, click the Pagination tab and, in the Place list, select On Page > *[page]*, where *[page]* is the name of master page 2.

Page 2 inherits the layout of master page 2 and now has a landscape layout.

Applying master pages to forms whose number of pages vary

Forms containing subforms that are set to flow content have a varying number of pages. When the form is rendered, objects are placed inside content areas and, if the data fills the content area, a page break is automatically inserted. By default, all of the pages inherit the format of the default master page.

When more than one master page is available, each one can influence how pages are formatted. In this case, the layout of a master page is applied when the subform that it is associated with is rendered. That is, if the subform is rendered on page 2 of the form, the format of the assigned master page is applied to page 2.

Two options are available for applying master pages to forms whose number of pages vary. The nature of the type of form should dictate which of these options are used.

First, you can limit the number of times a master page is used in a form by setting minimum and maximum page-occurrence settings. For example, the first master page needs a minimum and maximum page-occurrence setting of one to indicate that it always occurs only once. For master pages, the default minimum count is 0 and the maximum count is -1. For the second master page, the layout can be rendered on a page an infinite number of times.

Alternatively, you can define the placement of the master pages in the page set, such as in the first printed page, the last printed page, or the printed pages that occur in between. For example, a pay statement has company and employee-specific information on the first page. Subsequent pages show very little company or employee information and a significant amount of time card information. You can define two master pages. The first master page will have the company logo and contact information first, outside the content area. A small content area will follow to receive employee-specific information. The second master page will have a larger content area, possibly covering the whole page, to receive time card information.

By default, all form designs are created with the option of applying master pages by setting minimum and maximum occurrence values. Although different, the result of the rendered page using either option is the same. However, if the form is intended for double-sided printing, use the placement options.

Keep in mind that if you define the placement of the master pages in a form design and then you switch to setting page-occurrence values, you may need to make some changes to the form before it will behave

as expected. Also, defining the placement of master pages is recommended only for form designs saved as Acrobat 8 (Static) PDF forms or for form designs intended for printing, including PCL and Postscript.

To use the first master page one time only

You can set minimum and maximum page-occurrence values to use the first master page one time only, or you can specify the master page's placement as the first page in the page set.

By default, all form designs are created with the option of applying master pages by setting minimum and maximum occurrence values. Although different, the result of the rendered page using either option is the same. However, if the form is intended for double-sided printing, use the placement options.

Keep in mind that if you define the placement of the master pages in a form design and then you switch to setting page-occurrence values, you may have to make some changes to the form before it will behave as expected. Also, defining the placement of master pages is recommended only for form designs saved as Acrobat 8 (Static) PDF forms or for form designs intended for printing, including PCL and Postscript.

To use the first master page one time only by specifying minimum and maximum page-occurrence values

1) In the Hierarchy tab, click the first master page, Page1.
2) Click the Object palette and do the following tasks:
   – Select Restrict Page Occurrence.
   – Select Max and then type `1` in the corresponding box.
3) Save the form design.

To use the first master page one time only by specifying the master page's placement in the page set

1) In the Hierarchy tab, click the first master page, Page1.
2) In the Object palette, click the Pagination tab
3) Select First Page (in Page Set) in the Placement list.
4) Save the form design.

Controlling the order and visibility of pages

Page sets allow you to control the order and visibility of pages in the form design. Form designs contain at least one page set.

RELATED LINKS:

Making objects visible, invisible, or hidden

Setting up pages

Page set properties in the Page Set tab

Using page sets to control single-sided and double-sided printing in a form

Specifying page placement and printing options in existing form designs

To specify the master page placement in a page set

To move between multiple page sets

1) In the Hierarchy palette, click the page set.

   TIP: *If the Hierarchy palette is not visible, select Window > Hierarchy.*

To add or delete a page set

- To add a page set, in the Hierarchy palette, right-click the Master Pages node and select Insert Page Set.
- To delete a page set, in the Hierarchy palette, right-click the page set and select Delete.

To reorder page sets

1) In the Hierarchy palette, drag the page set to a new position below the Master Pages node.
2) Repeat step 1 until all of the page sets are in the order you want.

To rename a page set

1) In the Hierarchy palette, right-click the page set and select Rename Object.
2) Type a new name for the page set.

NOTE: You can also rename the selected page set in the Page Set tab of the Object palette.

To manage page sets by controlling the number of occurrences of each page

1) In the Hierarchy palette, select the page set.
2) In the Object palette, select Page Occurrence from the Printing list. Designer notifies you that if you previously selected Print on Front Side Only or Print on Both Sides and are now switching back to the Page Occurrence option, you may need to make changes to the form before it will behave as expected. Click Yes to continue.

Setting page break controls

Designer provides ways to control the page breaks that are inserted when a form that has a flowable layout expands beyond the bottom edge of a rendered page:

Widow and Orphan Control

Lets you prevent a single line of text from being separated from the rest of a paragraph when a page break is introduced. A *widow* is the last line of a paragraph that appears by itself at the top of the next page. An *orphan* is the first line of a paragraph that appears by itself at the bottom of a page.

Keep With Next

Lets you keep a text object or text field object with the next object in the document when a page break is introduced.

Allow Page Breaks Within Content

Allows page breaks within the content of a text object or text field object.

When both the Keep with Next and Allow Page Breaks Within Content options are selected, the Allow Page Breaks Within Content option is considered first followed by the Keep with Next option.

The Widow and Orphan, Keep With Next, and Allow Page Breaks Within Content options are not available for objects on a master page or in artwork.

You can set the default page break options for new forms, page break options for an individual form, and page break options for selected objects by using the various options in the Form Properties dialog box, Options dialog box, and Object palette.

To control widow and orphan lines for new forms

Use the Formatting page in the Options dialog box to control widow and orphan lines for new forms. The widow and orphan functionality can prevent a single line of text in all text and text field objects in the form from being separated from the rest of a paragraph when page breaks are introduced. The widow and orphan control ensures that a minimum of two lines from the paragraph are kept together.

To control widow and orphan lines for new forms:

1) Select Tools > Options.
2) Click Formatting and select Widow and Orphan Control.
3) Click OK.

To control widow and orphan lines for a form

Use the Formatting tab in the Form Properties dialog box to control window and orphan lines in all text and text field objects for individual forms.

The default setting that initially appears in the Form Properties dialog box is inherited from the Options dialog box on the Formatting page. The default setting automatically applies to all new forms. If you change the default setting in the Form Properties dialog box, the new setting overrides the setting in the Options dialog box for the current form. The Widow and Orphan Control option in the Options dialog box does not change.

NOTE: To modify widow and orphan lines for new forms, use the Options dialog box. (See *To control widow and orphan lines for new forms*.)

To control widow and orphan lines for an individual form:

1) Select File > Form Properties.
2) Click the Formatting tab and select Widow and Orphan Control.
3) Click OK.

To remove widow and orphan control from a form

You can remove widow and orphan control from the text and text field objects.

To remove widow and orphan control from a form:

1) Select File > Form Properties.
2) Click the Formatting tab and deselect Widow and Orphan Control.
3) Click OK.

RELATED LINKS:

To allow page breaks within a text object

To keep a text object with the next object in the form

Allowing page breaks within a text field

To keep a text field with the next object in the form

Formatting (Options dialog box)

Formatting tab (Form Properties dialog box)

3.3. Styles

You can create and manage style sheets and styles to provide consistent formatting within a single form or across multiple forms. For example, you can control the look of caption and field value text, the appearance of object borders and background colors, as well as the size and style of radio buttons and check box objects.

You use the Style Catalog to manage styles sheets, and to edit and apply styles to objects in a form design. The Style Catalog lists the various style sheets available with a form and the styles included with each style sheet. The Style Catalog organizes the style sheets into different panels, one for each style sheet. The first panel is for the internal style sheet, which is embedded within the form. Below the internal style sheet panel are panels for each external style sheet (Designer Style Sheet.xfs) that you add to the Style Catalog. Each panel bar shows the name of the style sheet. If you hover the mouse over the panel bar, the location of the style sheet is displayed in a tooltip. Menus are also available on each panel bar and a context menu is available with each listed style. Additional commands are available on the Style Catalog menu.

An internal style sheet is automatically associated with each new form you create. The internal style sheet is useful when you are creating a single form and you want the style sheet and styles embedded in the form for easy editing. However, if you are creating more than one form design, you can create multiple external style sheets in Designer, and add them to any number of form designs.

You can extract the styles from the internal style sheet to a new external style sheet for use in other forms. Alternatively, you can add the styles in various external style sheets to the internal style sheet by embedding the external style sheets within the form design.

Also, using options in the Form Properties and Options dialog boxes, you can specify default fonts for captions and values in new or existing forms to quickly change the fonts for all form objects.

To sort the style sheets in the Style Catalog

The Style Catalog organizes style sheets into different panels, one for each style sheet. The first panel is for the internal style sheet. Below the internal style sheet panel are panels for each Designer Style Sheet file (XFS) file that you add to the Style Catalog.

By default, Designer arranges the style sheets in the order they are added to the Style Catalog. You can resort the external style sheet files list in the Style Catalog by name or by type of style sheet.

1) On the Style Catalog palette menu, point to Sort, and then perform one of the following actions:
   - To sort the list of style sheets by name, click Name.
   - To sort the list of style sheets by type, click Type.
   - To unsort the list of style sheets, click Unsort.

To select a default style sheet for new forms

You can select an Designer Style Sheet (XFS) file for new forms. When you create a new form, the selected XFS file appears in the Style Catalog ready for use.

1) Click Tools > Options.
2) Select the Formatting panel.
3) Select Use Style Sheet.
4) Click the browse button, and select the Designer Style Sheet (XFS) file to use.

To add a style sheet to a form

You can add one or more Designer Style Sheet (XFS) files to a form. After you add a new style sheet, the Style Catalog displays a new panel listing the styles that are available for use in the form.

You can add the styles in the XFS files listed in the Styles catalog to the internal style sheet, by embedding the styles in the form. See *To embed a style sheet*.

You can hover the mouse over the panel toggle bar, to display the location of the style sheet.

1) Select Windows > Style Catalog.
2) On the Style Catalog palette menu, click Add style sheet.
3) In the Open dialog box, navigate to and select the Designer Style Sheet (XFS) file to use.
4) Click Open.

To create a new style sheet

When you create a new style sheet, Designer opens a blank Designer Style Sheet (XFS) file in which to create new styles. Designer displays style sheets using a different background color than forms to distinguish them. In the Hierarchy palette, the top-level subform is named Style sheet.

1) Click the Style Catalog menu, and select New Style Sheet.
2) Name and save the style sheet file.

To embed a style sheet

You can embed the styles from any number of Designer Style Sheet file (XFS) files within a form design, for ease of editing.

When you select the Embed Style Sheets command, Designer adds the styles from all of the external style sheet panels in the Style Catalog to the internal style sheet panel. Any references from form objects or default styles are reset to the new internal styles and the references to the external style sheets are removed from the form.

The Embed Style Sheets command is available when the Style Catalog contains one or more external style sheets.

1) Click the Style Catalog menu.
2) Select Embed Style Sheets.

To extract styles to a new style sheet

You can extract the styles from the internal style sheet to a new Designer Style Sheet file (XFS). You can also replace the internal style sheet with a reference to the new external style sheet, when needed.

1) Click the menu on the Internal style sheet panel.
2) Select Extract Styles To A New Style Sheet.
3) In the File Name box, enter a name for the new style sheet file.
4) (Optional) Select Replace Internal Style Sheet With Reference To The New External Style Sheet.
5) Click Save.

To remove a style sheet

You can remove an external Designer Style Sheet file (XFS) from the Style Catalog at any time.

The Style Catalog organizes the style sheets that are available with a form into different panels. The first panel is for the internal style sheet. You cannot remove the internal style sheet. Below the internal style sheet panel are panels for each Designer Style Sheet file (XFS) referenced by the form.

1) Click the menu on the panel of the external style sheet you want to remove.
2) Click Remove Style Sheet.

To replace a style sheet

You can replace any Designer Style Sheet file (XFS) in the Style Catalog at any time. The Style Catalog organizes the style sheets that are available with a form into different panels. The first panel is for the internal style sheet. You cannot replace the internal style sheet. Below the internal style sheet panel are panels for each Designer Style Sheet file (XFS) referenced by the form.

NOTE: If the style names are the same in two different style sheets, when you replace the one style sheet with the other, the objects in the form are automatically updated with the style properties from the replacement style sheet.

1) Click the menu on the panel of the external style sheet you want to replace.
2) Click Replace Style Sheet.

To edit a style sheet file

You can edit the styles in an external Designer Style Sheet file (XFS). You do not need to add a style sheet file to the Style Catalog to edit it.

1) To open a style sheet for editing, do one of the following actions:
   – If the style sheet file is in the Style Catalog, click the menu on the style sheet panel you want to edit, and then click Edit Style Sheet. Designer opens the style sheet file.
   – If the style sheet is not in the Style Catalog, click File > Open to select and open the file in Designer. Designer adds the styles to the Internal Style Sheet panel.
2) In the Style Catalog, right-click the style to edit, and select Edit Style.
3) In the Style Editor, change formatting options as needed.
4) Save a close the style sheet file.

Create a new style from an object

You can create a new style from an object in an external Designer Style Sheet file (XFS).

1) Click the menu on the external style sheet panel you want to use to create the new style.
2) Select Edit Style Sheet.
3) On the Design View tab, right-click the object to use to create the new style, point to Styles, and then click Create New Style From Object.
4) In the Style Editor, name the style and select formatting options as needed.
5) Click OK.
6) Save changes and close style sheet file.

Creating a new style for an object type

You can create new styles in the internal style sheet of a form design, or in an external Designer Style Sheet file (XFS) file.

When creating a style for an object with caption, value, or border formatting, you can select an existing style of the same object type or of the Common Style type from which the new style can inherit the properties specified in the style sheet.

After you create the new style, it appears in the Style Catalog.

To create a new style in the internal style sheet

You create a new style for the internal style sheet within the current form design.

1) Click the menu on the internal style sheet panel.
2) Select Create New Style. The Style Editor dialog box is displayed.
3) Select the General panel.
4) In the Style Name box, type a name for the style.
5) In the Type list, select the object type you want to create a style for.
6) Select formatting options as needed, and then click OK.
7) Save changes and close the style sheet file.

To create a new style in an external style sheet file

To create a new style in an external Designer Style Sheet file (XFS), open the style sheet file in Designer and create styles in that file.

1) Click the menu on the external style sheet panel to which you want to add a style.
2) Select Edit Style Sheet.
3) In the Style Catalog, right-click the style to edit, and select Create New Style. The Style Editor dialog box is displayed.
4) Select the General panel.
5) In the Style Name box, type a name for the style.
6) In the Type list, select the object type you want to create a style for.
7) Select formatting options as needed, and then click OK.
8) Save changes and close the style sheet file.

Creating a common style

A common style is a generic style type, which can be referenced by any object type. As a result, you can define a single common style with properties, which can be inherited by all other object types. By default, a common style defines caption text formatting, value text formatting, and border properties. You can choose to define only text formatting properties or only border properties.

To create a common style in the internal style sheet

You create a common style for the internal style sheet within the current form design.

1) Click the menu on the internal style sheet panel.
2) Select Create New Style. The Style Editor dialog box is displayed.
3) Select the General panel.

4) In the Style Name box, type a name for the style.
5) In the Type list, select Common Style.
6) Select formatting options as needed, and then click OK.
7) Save changes and close style sheet file.

To create a common style in an external style sheet file

To create a common style in an external Designer Style Sheet file (XFS), open the style sheet file in Designer and create the common style in that file.

1) Click the menu on the external style sheet panel in which you want to create a common style.
2) Select Edit Style Sheet.
3) Click the menu on the internal style sheet panel.
4) Select Create New Style. The Style Editor dialog box is displayed.
5) Select the General panel.
6) In the Style Name box, type a name for the style.
7) In the Type list, select Common Style.
8) Select formatting options as needed, and then click OK.
9) Save changes and close the style sheet file.

To apply a style to an object

The Style Catalog lists the styles that are available with each style sheet. Each style in a style sheet is associated with a particular object type. You can apply a style to the same object type. For example, you can apply a Text Field style to a Text Field object. If the style you want to apply is not compatible with the selected object, the Apply Style command is unavailable.

You can also create common styles with properties that you can apply to all object types. See *To create a common style*.

1) On the Design View tab, select the object type you want to apply a style to.
2) In the Style Catalog, right click the style you want to apply.
3) Click Apply Style.

To remove a style from an object

You can remove a style directly from an object on the Design View tab.

1) On the Design View tab, select the object you want to remove a style from.
2) Right-click, and select Styles > Remove Style.

Editing a style

You can edit styles in the internal style sheet of a form design, or in an external style sheet that references an Designer Style Sheet file (XFS). You edit the styles in an internal style sheet in the current form design.

You edit the styles in an external style sheet, by opening the XFS style sheet in Designer. When you edit a style sheet file in Designer, the page background (in the Design view) is colored to differentiate it from the white background used for form designs.

With the internal style sheet, changes are immediately applied to all objects that use that style. With external style sheet files, the chances are applies when you save the form.

To edit a style in the internal style sheet

1) In the Style Catalog, expand the internal style sheet panel.
2) Right click the style you want to edit, and select Edit Style.
3) In the Style Editor, select options as needed, and click OK.
4) Saves the changes.

To edit a style in an external style sheet file

1) To open the style sheet file for editing, do one of the following actions:
   – If the style sheet file is in the Style Catalog, click the menu on the external style sheet panel to edit, and then click Edit Style Sheet. Designer opens the style sheet file.
   – If the style sheet is not in the Style Catalog, click File > Open to select and open the style sheet file in Designer.
2) In the Internal Style Sheet panel, right click the style you want to edit, and click Edit Style.
3) In the Style Editor, select formatting options as needed, and click OK.
4) Save the changes.

Deleting a style form a style sheet

You can delete a style from the internal style sheet of a form design, or from an external style sheet that references an Designer Style Sheet file (XFS). You delete styles directly from the internal style sheet of a form design. You delete the styles from an external style sheet, by opening the XFS style sheet in Designer.

To delete a style from the internal style sheet

1) In the Style Catalog, expand the internal style sheet panel.
2) Right click the style you want to delete, and select Delete Style.
3) Click Yes to confirm that you want to delete the style.

To delete a style from an external style sheet

1) In the Style Catalog, click the menu on the external style sheet panel.
2) Select Edit Style Sheet. Designer opens the style sheet file, temporarily removes the associated panel from the Style Catalog, and adds the styles to the internal style sheet panel.
3) In the internal style panel, right click the style you want to delete., and click Delete Style.

4) Save the changes and close the style sheet file. Designer moves the remaining styles from the internal style sheet panel back to the external style sheet panel.

To set a default style for new objects

You can select default styles for new objects from any style sheet in the Style Catalog. You can select a default style for most objects in the Object Library. The default styles you select for an object type are shown in the Style Catalog in bold italic text.

When the Apply Default Styles To New Forms option is selected and you add a new object to a form, Designer automatically applies the default styles for that object type to the new object.

1) In the Style Catalog, right click the object style you want to set as a default.
2) Click Set As Default Style.

To apply default styles to new objects

Before you can apply default styles to new objects, select a default style for each object type you add to your form design.

1) Click the Style Catalog menu.
2) Select Apply Default Styles To New Objects.

To edit default styles for new objects

You can change or clear the default styles for new objects.

1) Click the Style Catalog menu.
2) Select Edit Default Styles Settings.
3) Click the list next to the objects you want and do one of the following actions:
   - To clear the default style, select None.
   - To change the default style, select a substitute style.

RELATED LINKS:

Default Fonts (Form Properties dialog box)

Default Fonts (Options dialog box)

To set a default font for captions in new forms

To set a default font for captions in an existing form

To set a default font for values in new forms

To set a default font for values in an existing form

3.4. Printing forms

Setting up forms for printing

For forms intended for printing, you can select single-sided or double-sided printing for each page set in the form design. For forms intended for double-sided printing, you can specify which side of the paper the master page is used: odd (front), even (back), no odd/even restrictions, or blank.

Keep in mind that specifying single-sided or double-sided printing is recommended only for form designs saved as Acrobat 8 (Static) PDF forms or for form designs intended for printing, including PCL and Post-script.

When setting up forms for printing, it is recommended that you review the input data and consider preparing sample data files or have Designer automatically generate a sample data file to determine whether the layout, formatting, content, and behavior of a form responds as expected when it is printed. For example, when setting up forms for double-sided printing, you need to have sample data files for testing how a rendered form looks when the printed page ends on an odd page or an even page, and when the rendered form fits on one printed page.

You can also specify print settings for PDF forms. For example, for Acrobat 8 (Static) PDF forms and Acrobat 8 (Dynamic) XML forms, you can select the number of copies to print and whether to print single-sided or double-sided. When a user chooses to print the form, these options are automatically applied. For all PDF forms, you can specify that the form be printed immediately when it is opened. In this case, the Print dialog box appears when the user opens the form, and you can choose to print the form on the user's default printer or on a printer that you specify.

RELATED LINKS:

Specifying page placement and printing options in existing form designs

To use master pages in a form intended for double-sided printing

To specify the master page placement in a page set

Inserting a blank page after the first page in a form intended for double-sided printing

Using page sets to control single-sided and double-sided printing in a form

To automatically print a PDF form when it is opened

Specifying page placement and printing options in existing form designs

In form designs created using a version earlier than Designer 8, you control repetitive rendering of pages by specifying the Restrict Page Occurrence, Min, Count, and Max options in the Object palette for the selected page set or master page.

You can update an older form design to use page placement and printing options, which also take occurrence into account. Using the page placement and printing options, you can specify the pages on which a master page is used as well as the settings for single-sided and double-sided printing.

The result of specifying single-sided printing in a form design closely matches the result of setting minimum and maximum page set and master page-occurrence values. Therefore, it is recommended that you begin by updating the form design for single-sided printing. Then, when you are satisfied that the form renders properly, you can update the form design for double-sided printing, if necessary. When you select double-sided printing, you can control whether the master page and other objects, including subforms, subform sets, and tables, occur on the odd (front) or even (back) pages.

Notice that when you select the page placement and printing options in the form design, Designer disables the occurrence settings because management of the page set occurrences is now controlled by the placement and printing options that you select. Therefore, it is recommended that you back up the form design before you begin.

To specify page placement and printing options in existing form designs

1) In the Hierarchy palette, select the page set.
2) In the Object palette, select Print on Front Side Only from the Printing list. Designer notifies you that the Restrict Page Occurrence option will be disabled. Click Yes to continue.
3) Select the first master page in the page set and click the Master Page tab in the Object palette. Notice that the Restrict Page Occurrence option is now disabled. Before this option was disabled, this master page had a minimum and maximum page-occurrence setting of one, indicating that it always occurs only once.
4) Click the Pagination tab and notice that the Odd/Even and Placement lists are now enabled and that there are no placement restrictions for this master page.
5) To specify that the master page occurs once, select First Page (in Page Set) from the Placement list.
6) Preview the form in the Preview PDF tab to confirm that the master page is used once.
7) Select the next master page in the page set.
8) For master pages in the page set where the minimum page-occurrence setting was 0 and the maximum setting was 1, select one of these options from the Placement list in the Pagination tab in the Object palette:
   – To specify that the master page is used for pages between but not including the first and last pages, select Rest of Pages.
   – To specify that the master page is used for all pages but not including the first page, select No Placement Restrictions.
9) For master pages in the page set where the minimum and maximum page-occurrence settings were one, select Last Page (in Page Set) from the Placement list to specify that the master page is used for the last page.
10) Preview the form in the Preview PDF tab.

To specify double-sided printing for an existing form design

1) Follow the previous procedure for specifying page placement and single-sided printing options in existing form designs.
2) Select File > Form Properties.
3) Click the Defaults tab and, in the Choose Version To Run Form In list, select Acrobat and Adobe Reader 9.0 or later.
4) Click the Preview tab and, in the Preview Type list, select Print Form (Two-sided).
5) (Optional) To test the form that you are previewing by using a data source that you created, use the Browse button to navigate to the file. You can also enter the full path to your test data file in the Data File box. You should plan to have several data source files available for testing forms that are intended for double-sided printing. For example, you should have a data source file that fills only

one page, a data source file where the rendered form ends on an even page, and a data source file where the rendered form ends on an odd page.

6) (Optional) To test the form you are previewing by using an automatically generated data source, click Generate Preview Data. If the form contains repeating subforms or subform sets, indicate the number of times each subform or subform set will repeat in the data file. You can use the Browse button to navigate to the location where you want the file saved. Type a name for the file and then click Generate.
7) In the Preview Adobe XML Form As list, select Static PDF Form, and then click OK.
8) In the Hierarchy palette, select the page set.
9) In the Object palette, select Print on Both Sides from the Printing list.
10) Preview the form in the Preview PDF tab. It is recommended that you preview the form as you work. If you encounter unexpected behavior in the previewed form, review the settings for the Place and After option in the Pagination tab for the subforms, subform sets, and tables in the form design. For example, a subform that is set to be placed at the top of the next even page may cause an unexpected empty odd page in the middle of the form.
11) Here is a list of ways that you can set up the form for double-sided printing:
    - *To use master pages in a form intended for double-sided printing*
    - *To specify the master page placement in a page set*
    - *Inserting a blank page after the first page in a form intended for double-sided printing*
    - *Using page sets to control single-sided and double-sided printing in a form*

To prepare a form design to use with preprinted paper

You can create a form design that generates a *preprint form*, which is a form printed on paper that already has lines, graphics, or text printed on it. For example, you can generate a preprint form for printing information on an income tax statement.

Notice that the following description lets you create a form that prints on preprinted paper and also generates a PDF for archiving. However, the resulting PDF will not be accessible to users with vision impairment.

After the form design is ready, a batch process gets the data from the SAP system and prints the information on preprinted paper. The form design can also generate a PDF form that contains both the lines, graphics, or text and the information. This PDF form is useful for previewing or archiving. You must set the appropriate options in your application for the process to work with preprinted paper.

To prepare the preprint form, create two sets of master pages:

Master pages for printing the form

This set of master pages contains the fields that will print on the preprinted page. They do not contain any of the lines, graphics, or text on the preprinted paper. Their purpose is to print data on the preprinted paper.

Master pages for previewing the form

This set of master pages contains the lines, graphics, and text that are on the preprinted paper. The SAP system does not print these master pages. These master pages are used for generating the PDF form for archiving or previewing.

When creating the preprint form design, place the fields that you want printed on the master pages for printing, and place the objects that are on the preprinted paper on the master pages for previewing. After you create the preview and print master pages, add a FormCalc script to generate the preprint form. When the PDF form is rendered, you have a preview of how the form will look when the information prints on the preprinted paper, and you can archive it.

Here is an example of how you can create a preprint form design that produces a PDF form for previewing and archiving.

1) Create the Preview First Page, Preview Next Page, Print First Page, and Print Next Page master pages.
2) Select one of the first master pages and then click the Object palette.
3) To use the selected master page one time only by specifying minimum and maximum page-occurrence values, click the Master Page tab, click Restrict Page Occurrence, and then do the following tasks:
   – Select Min and type `0` in the corresponding box.
   – Select Max and type `1` in the corresponding box.
4) Repeat step 3 for the other first master page.
5) To use the selected master page one time only by specifying the master page's placement in the page set, click the Pagination tab, select First Page (in Page Set) in the Placement drop-down list.
6) Repeat step 5 for the other first master page.
7) Place the objects that you want on the preview and print master pages.
8) Write the following FormCalc script in the `form:ready` event of the topmost subform.

```
if ( $xfa.config.present.destination ne "pdf" ) then
$.break.before = "pageArea"
$.break.beforeTarget = "#PrintFirstPage"
endif
```

9) Modify the `$.break.beforeTarget` value to match the name of the print master page. For example, use `PrintFirstPage`.

When the form renders, the form displays as it will look when output on preprinted paper.

RELATED LINKS:

Specifying page placement and printing options in existing form designs

To use master pages in a form intended for double-sided printing

To specify the master page placement in a page set

Inserting a blank page after the first page in a form intended for double-sided printing

Using page sets to control single-sided and double-sided printing in a form

To use master pages in a form intended for double-sided printing

For forms intended for double-sided printing, you can specify which side of the paper the master page is applied to: odd (front), even (back), no odd/even restrictions, or blank.

You can use master pages for the odd (front) and even (back) sides of a form intended for double-sided printing. For example, you may want the page numbers of a double-sided printed form to appear at the lower-right corner of the odd (front) printed pages and the lower-left corner of an even (back) printed pages.

Select the Blank Pages option when you need a blank page inserted between two odd (front) double-sided printed pages.

NOTE: The options in the Odd/Even list are not relevant for form designs that are based on setting minimum and maximum page-occurrence values.

1) In the Hierarchy palette, under the Master Pages node, select the master page.
2) In the Object palette, click the Pagination tab.
3) In the Odd/Even list, select the printed page where the master page is applied.

RELATED LINKS:

Specifying page placement and printing options in existing form designs

To specify the master page placement in a page set

Inserting a blank page after the first page in a form intended for double-sided printing

Using page sets to control single-sided and double-sided printing in a form

To specify the master page placement in a page set

Form designs contain at least one page set. For forms intended for printing, you need to specify where in the page set the master page is used; for example, in the first printed page, the last printed page, or the printed pages in between.

NOTE: The first page in the root page set cannot be blank; it must contain a content area.

In addition, for form designs that contain more than one page set, you must specify a break in a subform for printing to transition from one page set to the next.

NOTE: Specifying master page placement is not relevant for form designs that are based on setting minimum and maximum page-occurrence values.

The Only Page (in Page Set) option is useful when the rendered form can be printed on one side of a printed page. For example, although the data fits on one printed page in the form below, two pages are printed because there is a master page for the first printed page and a master page for the last printed page in the page set. In the example, the header information appears on the first page master page and the footer information on the last page master page.

To prevent two pages from being printed when everything can fit on one page, create a master page and select the Only Page (in Page Set) option. Whenever the data of the rendered form fits on one side of a printed page, this master page is used.

1) In the Hierarchy palette, under the Master Pages node, select the master page you want to place.
2) In the Object palette, click the Pagination tab.
3) In the Placement list, select the placement of the master page.

RELATED LINKS:

Specifying page placement and printing options in existing form designs

To use master pages in a form intended for double-sided printing

Inserting a blank page after the first page in a form intended for double-sided printing

Using page sets to control single-sided and double-sided printing in a form

Inserting a blank page after the first page in a form intended for double-sided printing

To insert a blank page after the first page in a form that is intended for double-sided printing, the page set must include master pages for the first, odd, and even printed pages. For example, you may want a blank page to print on the back side of the cover letter. By specifying that the area that follows the cover letter subform begin on an odd page, a blank page will be inserted as the first even page when the form is printed.

A.

This page uses the front_page master page

B.

The blank page is inserted as the first even page by using the blank_page master page

C.

This page uses the odd_page master page

D.

This page uses the even_page master page

E.

The area after this subform is set to Go To Next Odd Page

To insert a blank page after the first page in a form intended for double-sided printing

1) Add a new master page to the page set.
2) Click anywhere in the master page.
3) Click the Pagination tab and select Blank Pages from the Odd/Even list.
4) Select the subform or subform set that occurs before the blank page.
5) Click the Pagination tab and select Go To Next Odd Page from the After list.
6) Preview the form in the Preview PDF tab.

RELATED LINKS:

Setting up forms for printing

Specifying page placement and printing options in existing form designs

To use master pages in a form intended for double-sided printing

Using page sets to control single-sided and double-sided printing in a form

For forms intended for printing, you can specify single-sided or double-sided printing for each page set in the form design.

NOTE: Specifying single-sided or double-sided printing is recommended for form designs intended for printing, including PCL and Postscript.

For example, you may want to create a form design to be used to generate a preprint and plain copy form. A *preprint form* is a form that is printed on paper that already has items printed on it. A *plain copy form* is a form that is printed on blank paper.

For this example, you create five master pages that are combined into page sets. You can specify which page sets print single-sided and which ones print double-sided.

MasterPage1

Contains the title page information and is used in the first page in the page set and prints single sided. MasterPage1 includes a page break in a subform that allows printing to transition to the PrePrint page set.

MasterPage2

The paper that the preprint form is printed on contains the company logo and contact information. This master page leaves room for that information outside the content area. A small content area follows next to receive employee-specific information. It occurs once for the preprint form and therefore is assigned as the first page in the second page set. The second page set requires double-sided printing.

MasterPage3

Has a larger content area, possibly covering the whole page, to receive more information. It is used for the second and subsequent pages of the preprint form. It is assigned the Rest of Pages placement option in the second page set. MasterPage3 includes a page break in a subform that allows printing to transition to the PlainCopy page set.

MasterPage4

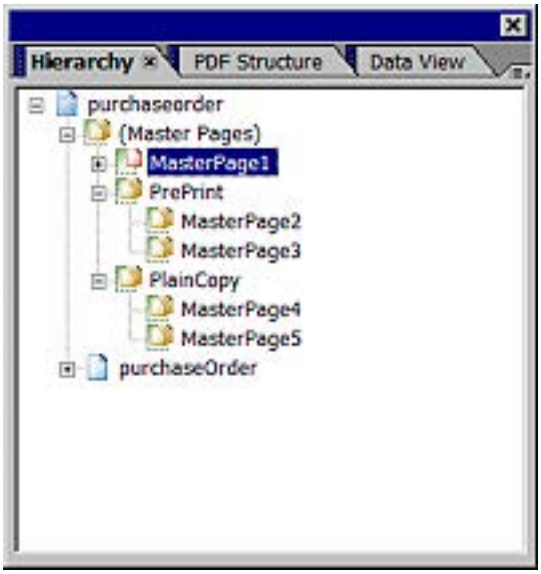
Has the company logo and contact information first, outside the content area. A small content area follows next to receive employee-specific information. It occurs once for the plain copy form and therefore is assigned as the first page in the third page set. The third page set also requires double-sided printing.

MasterPage5

Has a larger content area, possibly covering the whole page, to receive more information. It is used for the second and subsequent pages of the plain copy form. It occurs 0 to infinite times. It is assigned the Rest of Pages placement option in the second page set.

The master pages are grouped into page sets:

- The Main page set is set to print single-sided.
- The PrePrint page set is set to print double-sided.
- The PlainCopy page set is set to print double sided.



The form is rendered according to the order of the page set in the hierarchy:

- The first subform is always placed on MasterPage1.
- The next subform is placed on MasterPage2 and then MasterPage3 if the form is a preprint form, or on MasterPage4 and then MasterPage5 if the form is a plain copy form.

RELATED LINKS:

Controlling the order and visibility of pages

Specifying page placement and printing options in existing form designs

To use master pages in a form intended for double-sided printing

To specify the master page placement in a page set

Inserting a blank page after the first page in a form intended for double-sided printing

To print the contents of the Design View or Master Pages tab

1) In the Layout Editor, click the Design View tab or Master Pages tab.
2) Select File > Print.
3) Set the required options and then click OK.

NOTE: *The options that display in the Printer and Print range sections of the dialog box depend on the printer that your computer is connected to. For more information about your printer's options, see the printer documentation.*

RELATED LINKS:

To print a form with sample data

To print a form with sample data

Designer lets you print a final version of the current form design with sample data. When you print the form, the data values from the sample XML file will appear in the respective objects.

By testing your form with sample data, you can see the final product and verify your design.

1) Select File > Print.
2) Select Print Form With Data.
3) (Optional) To print the form with a data source, use the Browse button to navigate to the file. You can also enter the full path to your test data file in the Data File box.
4) (Optional) To print the form with an automatically generated data source, click Generate Data File. If the form contains repeating subforms or subform sets, indicate the number of times each subform or subform set will repeat in the data file. You can use the Browse button to navigate to the location where you want the file saved. Type a name for the file and then click Generate.
5) Click OK.

RELATED LINKS:

To create a data connection to an XML schema

To create sample data to print with your form

You can print a final version of the current form design using sample data to ensure that the form prints correctly. If you do not have sample data, you can create it using Acrobat.

After you create the sample data file, you can specify which data file to use when printing the form. The next time you print the form in Designer, the sample data will appear in the respective objects.

1) In Designer, save the form design as a PDF file.
2) Open the PDF file in Acrobat and enter values in the fields you want to test.
3) In Acrobat, do these tasks:
   – Select Advanced > Forms > Export Form Data.
   – In the Export Form Data As dialog box, name the file and save it as XML Data Package (*.xdp).
4) In Designer, do these tasks:
   – Select File > Print.
   – Select Print Form With Data.
   – To print the form with a data source, use the Browse button to navigate to the test data XML file. You can also enter the full path to your test data XML file in the Data File box.
   – Click OK.

RELATED LINKS:

To print a form with sample data

To automatically generate sample data to print with your form

You can generate sample data to print with your form instead of creating a sample data file using Acrobat. Also, if your form contains repeating subforms or subform sets, you can specify the number of times the data will be repeated when you print the form.

Designer generates sample data that is valid for the corresponding objects in the form, with a few exceptions:

- Sample data is not generated according to any validation scripts that may be specified for an object.
- The minimum and maximum count for a subform will restrict the number of repeating subforms that you specify for the generated sample data file.
- The default value you select for a 2D barcode is retained in the generated sample data file.

After you generate the sample data file, you can edit the file, if required.

Specify which data file to use when printing the form. The next time you print the form in Designer, the sample data will appear in the respective objects.

1) Select File > Print.
2) Select Print Form With Data.
3) Click Generate Data File.
4) In the Data File box, use the Browse button to navigate to the location of the test data file. You can also type the full path, including a file name, for the test data file.

5) In the Repeating Elements list, select the number to the right of a subform and type the number of times it will repeat in the data file.
6) Click Generate.

RELATED LINKS:

To print a form with sample data

To specify settings for printing a PDF form

To print a PDF form, you can specify the settings in the form design, instead of placing the effort on the user to properly configure the print job. For example, you can select the number of copies to print and single-sided or double-sided printing or select page scaling options. These settings are automatically applied when the user prints the form.

NOTE: Specifying settings for printing is recommended only for form designs saved as Acrobat 8 (Static) PDF forms or Acrobat 8 (Dynamic) XML forms.

1) Select File > Form Properties.
2) Click the PDF Print Options tab and select Use These Print Settings for Printing This PDF Form.
3) Select the number of copies to print.
4) Select one of the following options for duplex printing. To print double-sided, the selected printer must support duplex printing.
   - To print on one side of the paper, select Simplex.
   - To print on both sides of the paper where the paper flips along the long edge, select Duplex Flip Long Edge.
   - To print on both sides of the paper where the paper flips along the short edge, select Duplex Flip Short Edge.
5) Select one of the following options for page scaling:
   - To use the page scaling options selected in the Adobe Acrobat/Reader Print dialog box, select Use Adobe Acrobat/Reader Setting.
   - To set Page Scaling to None in the Adobe Acrobat/Reader Print dialog box, select No Page Scaling. The Adobe Acrobat/Reader user can change the Page Scaling setting. Select Prevent User From Changing to prevent Adobe Acrobat/Reader users from disabling page scaling.
   - To automatically select the paper source based on the page size, select Paper Source By Page Size.
6) Click OK.

RELATED LINKS:

To automatically print a PDF form when it is opened

To automatically print a PDF form when it is opened

You can specify that the Print dialog box appears when the PDF form opens, allowing the user to print the form immediately.

1) Select File > Form Properties.
2) Click the PDF Print Options tab and select Automatically Print the Form When it is Opened.
3) (Optional) To print the PDF form by using the default printer on the user's computer, select Print to the User's Default Printer.
4) (Optional) To print the PDF form using a specified printer, select Print To and select the printer from the list. You can also type the printer name. This is useful when you want to specify a printer that is not available to you. The printer name must match exactly the name of the printer as installed on the user's computer.
5) Click OK.

3.5. Tabbing order

Many users use the Tab key to move between fields and buttons in a form instead of using the mouse. Designer lets you set the tabbing order between objects in a form.

Tabbing order is important for interactive forms and forms that have a fixed layout. For interactive forms, the tabbing order affects the end user's experience when filling the form.

For both interactive and non-interactive forms, tabbing order is critical if your forms need to be accessible to users with vision or mobility impairments. These users typically do not use a mouse to navigate through the form, so they depend on the keyboard keys and a good tabbing order sequence to ensure that they have full access to all the fields on the form.

Accessible forms require a tabbing order, whether the form is interactive or designed for print. Additionally, a screen reader will read the form in geographic order, which in Designer is set when you use the default tabbing order.

Designer automatically sets a default tabbing order for each form. This tabbing order can be easily changed to better reflect the logical flow of the form and to accommodate particular user requirements.

How the default tabbing order works

The default tabbing order for objects in a form is from left to right, top to bottom, starting from the upper-left corner. Tabbing order respects the existence of subforms, radio buttons, and content areas. For example, if two subforms exist side-by-side, and each subform contains a number of field objects, the tabbing sequence will go through the fields in the first subform before moving on to the next.

NOTE: Designer does not include circle, line, or rectangle objects in the tabbing order.

Tabbing order is also determined by the vertical position of objects on a page and its master page.

The tabbing order starts from the object with the smallest vertical coordinate and ends with the object with the largest vertical coordinate, regardless of whether the object is on the body or master page.

For objects that contain objects, such as content areas, all child objects are tabbed through before tabbing to the next higher-level object.

The following list provides an example of the tabbing order for objects on body and master pages:

- Image object on the master page with a vertical coordinate of 1.
- Content area object on the page with a vertical coordinate of 4. All objects in the content area are tabbed through before tabbing to the subform object on the master page.
- Subform object on the master page with a vertical coordinate of 10. All objects in the content area are tabbed through before tabbing to the text object on the master page.
- Text object on the master page with a vertical coordinate of 12.

Because the tabbing order is important, ensure that you position objects precisely on the form, relative to each other. For example, you can position and size an object using its coordinates and you can snap objects to points on a grid.

You can change the default tabbing order if you require a different sequence in your form. For example, you may want to change the tabbing order to move through objects in a column, from top to bottom, and then left to right.

NOTE: In Acrobat 6.0.2, tabbing to a group of radio buttons makes the upper-left radio button active. Use the Tab key to move through the radio buttons. In Acrobat 7.0.5 and later, the selected radio button becomes active. Use the arrow keys to move between the radio buttons in a group and the Tab key to move out of the group.

RELATED LINKS:

To position objects

Viewing the default tabbing order

Changing the tabbing order

Viewing the default tabbing order

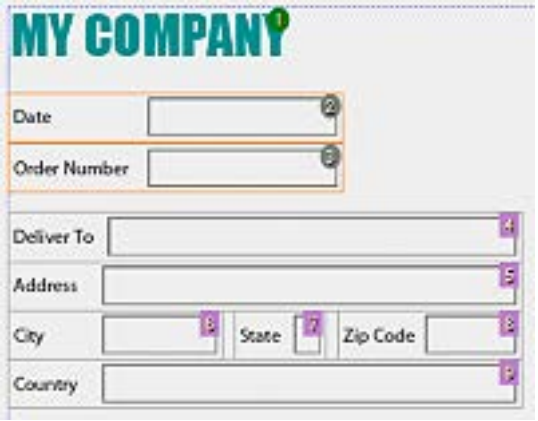
One of the final tasks to do when creating a form is to set the tabbing order. Before you do this task, make sure you have finished laying out the form design so that all the objects are in their correct position on the form. First, you should examine the default tabbing order, which is created automatically by Designer.

When you design or open a form in Designer, the tabbing order is not visible. To see the tabbing order for the form, select View > Show Tab Order or click Show Order on the Tab Order palette.

NOTE: If the Tab Order palette is not visible, select Window > Tab Order.

The tabbing order is displayed on the form as a series of consecutive numbers inside colored shapes:

- Numbers inside a gray circle indicates the default tabbing order for the objects in the content area.
- Numbers inside a green circle indicate the tabbing order for master page objects.
- Numbers inside a lavender square indicate the tabbing order for the objects inside a fragment.



Examine the default tabbing order carefully to determine whether it suits your particular requirements or whether you need to change it. Even if you modify the tabbing order, you can quickly return to the default tabbing order by selecting the Automatic option in the Tab Order palette.

NOTE: While viewing the tabbing order, you cannot edit any parts of the form.

To hide the tabbing order and return to normal form-editing mode

Perform one of the following actions:

- Select View > Hide Tab Order
- Click Hide Order in the Tab Order palette
- Open a different palette, such as the Hierarchy palette.

RELATED LINKS:

How the default tabbing order works

Changing the tabbing order

Using the Tab Order palette

The Tab Order palette provides an alternative view of the tabbing order on the form. It shows all the objects on the form as a numbered list, where each number represents the position of the object in the tabbing flow. The palette is also where you modify the tab order, if required.

To open the Tab Order palette, select Window > Tab Order.

The Tab Order palette may show the following visual markers in the list:

- A gray bar marks each page of the form. The tabbing order on each page starts with the number 1.
- The letter *M* inside a green circle indicates master page objects (visible only when viewing the form on the Design View tab).
- A range of numbers indicates objects within a fragment.
- A yellow background indicates the currently selected item.
- A lock icon beside the first object on the page indicates that the object cannot be moved within the tabbing order (visible only when viewing the form on the Master Pages tab).

The list shows exactly the same tabbing order numbers as the numbers displayed on the form itself.

The difference is that the numbers displayed on the form are for information purpose only, whereas the numbers on the list can be changed to modify the tabbing order.

In the Tab Order palette, you can also show or hide the tabbing order on the form and switch between the default and custom tabbing order.

RELATED LINKS:

Changing the tabbing order

Viewing the default tabbing order

Changing the tabbing order

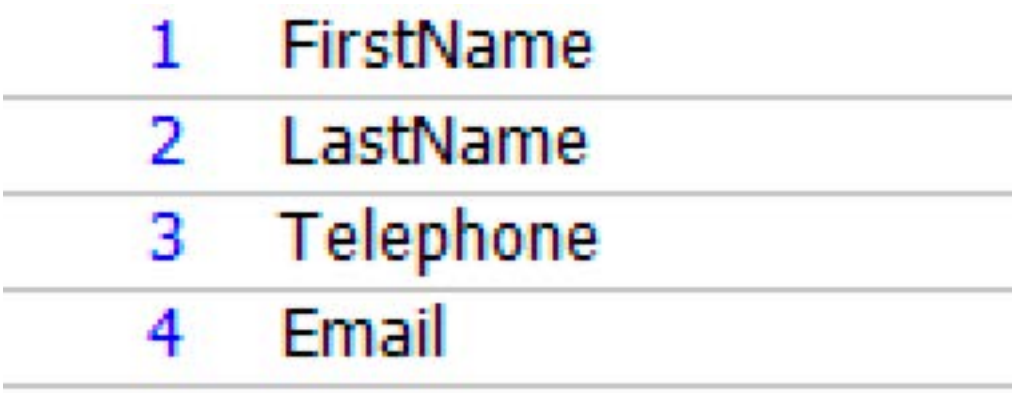
After examining the default tabbing order, you may decide that you need a different sequence for some of the objects on the form. For example, if you have two groups of address fields situated side-by-side, you may want to tab through the fields in the first group before moving to the second group.

You change the position of an object in the tabbing order by moving the object up or down in the Tab Order palette list. You can move a single object or a group of objects. When you move the object to a new place in the order, Designer reassigns the numbers to accommodate the object in its new place.

For example, you have four objects on the page, which are arranged in the following layout.

Therefore, their default tabbing order will be as shown in this illustration.

You may want to change this tabbing order to a more logical one, such as First Name, Last Name, Telephone, Email. All you have to do is move the LastName object one position up in the list. The tabbing order numbers are reassigned to reflect this move.



Although the tabbing order for the objects on the master page and the objects inside a fragment are displayed on the form, you cannot change the order for these objects on the Design View tab. For the master page objects, click the Master Pages tab and customize the tabbing order. (See *Changing the tabbing order on master pages*). For the fragments, open the required fragment and change the order inside the fragment. (See *To change the tabbing order in fragments*).

If your form has more than one page, the tabbing order for each page starts at the number 1, and you can change the order only inside each page.

Before you change the tabbing order, you should prepare your working area in the following way:

1) Select Window > Tab Order. The Tab Order palette appears on the left side of the working area.
2) Click Show Order and then select Custom.

   The tabbing order numbers displayed on the form are now inside blue squares 16. This square visually indicates that you are now in custom tabbing mode and can change the position of one or more objects in the tabbing sequence.

RELATED LINKS:

Using the Tab Order palette

To change the tabbing order for a single object using the mouse

Changing the tabbing order for a group of objects

To view the tabbing order using visual aids

To change the tabbing order for a single object using the mouse

You change the tabbing order by selecting the objects and changing their positions in the Tab Order palette list.

To change the order using drag-and-drop

1) Drag the selected object up or down the list and place it at the required location. A black handle marks your current position within the list before you place the object.

To change the order using the arrow buttons

1) In the Tab Order palette, click the up or down arrow buttons until the selected object is placed in the correct position.

To change the order using the menu

1) In the Tab Order palette menu, select Move First, Move Up, Move Down, or Move Last.

To change the order by editing the number

1) In the Tab Order palette list, click the selected object to make the number listed beside the object name editable. Then, type the new number indicating the new position of the object in the tabbing order and press Enter.

To change the order using copy and paste

1) Select Copy from the Tab Order palette menu and, in the list, select the object above which to place the object you are moving, and then select Paste or Paste After from the menu.

RELATED LINKS:

Changing the tabbing order

To change the tabbing order for a single object using keyboard shortcuts

Changing the tabbing order for a group of objects

To view the tabbing order using visual aids

To change the tabbing order for a single object using keyboard shortcuts

NOTE: You must use the mouse pointer in conjunction with the keyboard to change the tabbing order by using keyboard shortcuts.

You change the tabbing order by selecting the objects and changing their positions in the Tab Order palette list.

To change the order using the menu

1) Press Ctrl+Up Arrow or Ctrl+Down Arrow to move the object one position up or down in the list.

To change the order by editing the number

1) Press F2 to make the number listed beside the object name editable. Then type the new number that indicates the new position of the object in the tabbing order and press Enter.

To change the order using copy and paste

1) Press Ctrl+C to copy the selected object to the clipboard. Then press the Up Arrow or Down Arrow repeatedly until you highlight the object above which you want to place the object, and press Ctrl+V to paste the object from the clipboard.

RELATED LINKS:

Changing the tabbing order

To change the tabbing order for a single object using the mouse

Changing the tabbing order for a group of objects

To view the tabbing order using visual aids

Changing the tabbing order for a group of objects

Instead of moving individual objects to their new location within the tabbing order, you can move a group of objects all at once. This method works only for the objects that are positioned in sequence in the list of objects.

If you have a selection of objects that are scattered throughout the list, you must make them sequential first and then move the group.

After you select a group of sequentially ordered objects, you move the group to a new location in the tabbing order by using the same methods you used for moving a single object.

To select a sequential group of objects using the mouse

1) In the Tab Order palette list, click the first object in the sequence.
2) Press Shift and click the last object in the sequence.

To select a sequential group of objects using the keyboard

1) In the Tab Order palette list, click the first object in the sequence.
2) Press Shift+Down Arrow or Shift+Up Arrow until all objects in the sequence are selected.

To change a group of individually selected objects to a sequential group

1) In the Tab Order palette list, click the first object and then press Ctrl and click each object that you want to select.
2) In the palette menu, select Make Sequential. The selected objects are moved together into a sequentially ordered group.

To change the order to geographic (left-to-right)

1) In the Tab Order palette list, select the objects in the sequence.
2) In the Tab Order palette menu, select, Make Geographic.

To change the geographic order to right-to-left

1) In the Tab Order palette list, select the objects in the sequence.
2) In the Tab Order palette menu, select, Make Geographic Right-To-Left.

RELATED LINKS:

Changing the tabbing order

To change the tabbing order for a single object using the mouse

To change the tabbing order for a single object using keyboard shortcuts

To view the tabbing order using visual aids

Changing the tabbing order on master pages

Although the tabbing order for the objects that are located on a master page is displayed on the Design View tab, you can change the order for these objects only on the Master Pages tab.

The objects on the master page are placed in two areas, one above the content area and one below the content area. The tabbing order is set for objects within each of these two areas. Consequently, you can only change the tabbing order within each area.

NOTE: The upper-left object on each master page is locked in the first tabbing position and cannot be moved.

When you change the tabbing order on a master page, any pages in your form that are using this master page are affected by the change.

RELATED LINKS:

Changing the tabbing order

To change the tabbing order for a single object using the mouse

To change the tabbing order for a single object using keyboard shortcuts

Changing the tabbing order for a group of objects

To change the tabbing order in fragments

If you use fragment references in your form, the tabbing order inside a fragment is visible when viewing the order for the form. To change the tabbing order inside a fragment, you must open the fragment source file for editing, make the change, and save the file. Any forms that use this fragment are affected by this change.

1) Select View > Hide Tab Order.
2) Select the fragment reference.
3) Select Edit > Fragments > Edit Fragment.

4) Select View > Show Tab Order.
5) Change the tabbing order as required.
6) Save and close the fragment source file.

RELATED LINKS:

Changing the tabbing order

To return to the default tabbing order

If you decide that you do not want the customized tabbing order on your form, you can quickly return to the automatic (default) tabbing order. You will lose any changes made to the tabbing order.

1) On the Tab Order palette, select Automatic.
2) In the message box, click Yes to acknowledge that the custom tab order will be removed.

Excluding text and image objects from the tabbing order

You can exclude text and image objects from the tabbing order to improve usability and flow of the tabbing on the form.

For example, on the form below, the user will need to tab through the title of the form (Purchase Order) before reaching the P.O. Date field.

If the text and image objects are excluded from the tabbing order, the tabbing flow on this form will change to include only the fields that require user input or action.

Excluding text and image objects from the tabbing order is not recommended if one or more of the following is true for your form:

- Your form will be used with a screen reader.
- Your form contains hyperlinks inside the text objects.

You can exclude the text and image objects for both the default and the custom tabbing order.

To exclude text and image objects

1) Do one of the following actions:
   – Select Tools > Options > Tab Order and, in the Tab Order panel, select Only Show Tab Order For Fields.
   – In the Tab Order palette menu, select Show Fields Only.

RELATED LINKS:

Viewing the default tabbing order

Changing the tabbing order

To view the tabbing order using visual aids

On a complex and busy form, it may be difficult to see how the tabbing flows from one object to the next. You can use visual aids to help you see the tabbing flow on the form.

With the visual aids turned on, when you hover the pointer over the object, blue arrows show the tabbing flow for the two preceding and two following objects in the tabbing order.

1) Do one of the following actions:
   - Select Tools > Options > Tab Order and, in the Tab Order panel, select Display Additional Visual Aids For Tab Order.
   - In the Tab Order palette menu, select Show Visual Aids.

RELATED LINKS:

Viewing the default tabbing order

Changing the tabbing order

3.6. Building actions in forms

Use the Action Builder dialog box on the Tools menu to build common interactive capabilities in forms, without writing scripts. Using actions, you can control how you present form objects and data and how the objects and data respond to form filler interaction.

Here are examples of what you can do with actions:

- Add buttons that a form filler can click to add or remove sections in the form or rows in a table.
- Set the value of a field, such as prepopulating a date/time field object with the current date or a numeric field object with a specific value.
- Set the background color of fields.
- Hide or show objects or set the focus to a specific field.
- Create custom actions by using scripting objects and the function within the scripts.
- Build actions within fragments.

NOTE: To display the actions within a fragment file in the Action Builder dialog box, you must embed the fragments file within the Adobe XML Form (XDP) document or edit the fragment in Designer. The Action Builder dialog box does not display actions within a fragment that you create in a fragment library.

NOTE: Actions may not work with HTML forms and Guides.

You can add actions to most form fields and objects.

When you click the object link in the Action Builder dialog box, the Select an Object dialog box appears, displaying the objects in the form that you can select to create a condition or result.

You build actions by adding one or more conditions that must be met, and one or more results that occur when the conditions are fulfilled. You can build simple actions with just one condition, where the results begin when that condition is met. Alternatively, you can combine multiple conditions to build more complex actions, where the results can begin at different times depending on the conditions you add.

Designer generates a script for each action and monitors the scripts for changes. If Designer detects that the script has been modified, it performs the following actions:

- Stops monitoring the script, giving the form author ownership of the script.
- Displays a message on the Log tab in the Report palette, indicating that the script is no longer managed and can be edited.

Designer does not modify unmanaged script in any way.

Designer inserts the script that it generates for an action at the beginning of the script in the Script Editor, before any unmanaged scripts.

The Report palette lists warning messages about broken actions. Broken actions occur when an object that was used to create a condition or result is deleted from the form. Broken actions are indicated in the Action list. A missing object link also appears next to the relevant condition or result. Double-click the warning message in the Report palette to open the Action Builder dialog box and highlight the broken action.

NOTE: Designer does not monitor changes that you make to radio buttons and choice lists. If you change the items in the list, reorder items, change the display text or save value, or delete a radio button, the action can break without generating a broken action warning.

Combining conditions

If you combine a trigger condition with other conditions, the trigger condition must be met last. Only the trigger condition can start the action results, after all other conditions are met first. As a result, you can add only one trigger condition to an action. For example, you cannot build an action with two trigger conditions, where the form filler is required to click a button and click a check box at the same time. The Action Builder dialog box displays an error message if you add more than one trigger condition to an action. However, keep in mind that an action does not require a trigger condition. You can build an action without adding a trigger condition. If you do not add a trigger condition to an action, the conditions can be met in any order. Any one of the conditions in the action can display the results, after all other conditions are met.

When you add multiple conditions to an action, the Action Builder dialog box sorts the list of conditions, as shown below. The trigger condition (if you have added one) appears at the top of the list. All other conditions are grouped under the trigger condition. The *and/or* link appears next to the grouped conditions. The and/or link is not available with the trigger condition. When you select *and*, all conditions in the group must be met before the actions results occur. When you select *or*, at least one of the conditions in the group must be met before the action results occur.

A. Trigger condition B. And/or toggle link

Build an action

When you build an action, you add one or more conditions and one or more results. The results occur when the conditions are fulfilled. You can add as many conditions and results as you need. However, you can add only one trigger condition to an action, because the trigger condition must be met last. For example, you cannot build an action where the form filler must click a button and a check box at the same time, to initiate the results. The Action Builder dialog box displays an error message if you add more than one trigger condition to an action.

NOTE: If you build an action using the 'is changed' option for a condition, when the form is viewed in Acrobat or Adobe Reader, the results may not be applied until the field is exited.

For each condition you add, click the object link to open the Select an Object dialog box and choose an object. The Select An Object dialog box only shows the objects in the form that you can use for a condition or result. For each result you add, you select an option in the Select a Result list, and then choose various other options depending on the object. The options available for each result vary depending on the object you select.

NOTE: If you build an action that adds or removes instances of subforms, be sure to name each subform object. If the action references any unnamed subforms, the action could fail.

Keep in mind that the following results are not compatible with HTML forms or guides.

- Attach a File to the Form
- Close the Form
- Go to a Specific Page
- Reset All the Fields in the Form
- Save the Form
- Set the Zoom Level

Designer generates a default name for each action, which consists of the object name followed by the scripting event that the condition is generated within (<name of object>.<name of scripting event>).

For information on events, see Events in Designer ScriptingBasics.

NOTE: If you create an action in the Enter event and run the form in Acrobat 7.1.3, the background or foreground color of the field does not change until the user exits the field.

1) Select Tools > Action Builder.
2) Click the Add A New Action button.
3) In the Condition area, click the Add A Condition button.

4) Click the object link.
5) Select the object for the condition. Repeat steps 3 and 4 as needed. If you add three or more conditions, the *and/or* link appears next to the conditions. Click the link to change the relationship between the conditions as needed.
6) In the Result area, click the Add A Result button.
7) In the Select a Result list, select a result and then choose options as needed. Repeat steps 6 and 7 as needed.

Rename an action

1) Click the action name and type a new name. You can give two or more actions the same name.

Revert to the default name

1) Delete the new name.

Edit an action

Use the Actions dialog box to view and edit the actions in a form. Actions are listed on the left side of the dialog box, and the conditions and results for the selected action appear on the right. Add, delete, and modify condition and results as needed.

For existing conditions, the Select an Object dialog box only shows the object associated with that condition and other objects of the same type. To change the object type for an existing condition, delete the condition and then create another condition.

For existing results, the Select an Object dialog box shows all objects in the form that you can select for a result. To change the object for an existing result, select a different object.

1) Select Tools > Action Builder.
2) Under Actions, select an action, and select options as needed.

Remove an action

You can remove an action from a form at any time.

1) Select Tools > Action Builder.
2) Under Actions, select the action to remove.
3) Click the Remove An Existing Action button.

Build a custom result

You can create a custom result for an action by using a script object that contains a function. A *script object* is an object you can use to store JavaScript functions and values separately from any particular form object.

NOTE: Designer does not validate parameters and return values. Ensure that the parameters you type are correct and the return value is valid.

1) Select Tools > Action Builder.
2) Click the Add A New Action button.
3) In the Condition area, click the Add A Condition button.
4) Click the object link.
5) Select an object for the condition. Repeat steps 3 and 4 as needed. If you add three or more conditions, the *and/or* link appears next to the conditions. Click the link to change the relationship between the conditions as needed.
6) In the Result area, click the Add A Result button.
7) In the Select a Result list, select Call A Script Object Function.
8) Click the Call Script Object link and, under Variables, select the script object.
9) Select a function from the list.
10) Type the parameters in the box. Example: Call script *(script object)* function (function name) with parameters `*(funcParam1, funcParam2)*`and "(No Return Result)".
11) (Optional) Select Assign Return Result To, click the object link, and select an object for the return result.

Setting the zoom level in a PDF form

When a user opens a PDF form in Acrobat or Adobe Reader, they see the initial view of the document. You can build an action to set the initial viewing area of a PDF to a specific zoom level or page magnification. For example, you can set the zoom level to a specific percentage of the page size or to automatically fit the width or height of the page.

1) Select Tools > Action Builder.
2) Click the Add A New Action button.
3) In the Condition area, click the object link.
4) In the Select An Object dialog box, select the name of the form at the top of the form hierarchy. The default name is *form1*.
5) Click OK.
6) In the Condition area, select When Form <`form name`> Has Finished Loading.
7) In the Select a Result list, select Set The Zoom Level.
8) In the Set Zoom To list, select the option you want.
9) Click OK.

3.7. Spell checking in forms

To select the default locale

Before you check the spelling in a form, you should verify that the correct default form locale is selected. The locale option that is selected in the Default Form Locale list in the Form Properties dialog box specifies the default language that Designer uses to check spelling. For quick reference, the current spell-check language is indicated in the upper-right corner of the Check Spelling dialog box, just below the title bar.

If you apply different locale options to individual objects in the form by using the Locale list in the Object palette, those locale settings override the default form locale setting. That is, when the locale setting for an object is different from the locale setting for the form, Designer identifies the words (text) associated with these objects as being misspelled.

When applying different locale options to individual objects, keep in mind that the objects that are within tables and subforms automatically receive the same locale setting as the table and subform object.

NOTE: If you select a locale option (language) that the spell-check feature does not support, a message appears indicating that objects with that particular locale setting will not be spell checked.

1) Select File > Form Properties.
2) Click the Defaults tab and, in the Form Locale list, select the language you want the spell-check feature to use.
3) Click OK.

RELATED LINKS:

To check spelling while you type

To check the spelling in a form

Form Properties dialog box

To check spelling while you type

You can set up Designer to identify spelling errors as you type by selecting the Check Spelling While Typing option in the Spelling panel of the Options dialog box.

When the Check Spelling While Typing option is selected, a wavy red line appears under misspelled words in text objects, caption fields, or the various fields in the Object and Accessibility palettes, regardless of the options selected. You can use the options in the Spell Check list to specify the type of text to spell check when you use the Check Spelling dialog box.

When checking the spelling in a form, Designer always refers to the internal dictionary and your custom dictionary (My Custom Dictionary). You can also add other custom dictionary files (*.clam) to expand the list of correctly spelled words to which Designer refers. For more information, see *Adding a word to the list of ignored words*.

TIP: *To see a list of the dictionaries you currently have available, select Tools > Options, and click Spelling. The Spelling panel displays a list of the installed dictionaries. Notice that My Custom Dictionary is selected at all times and cannot be removed.*

Keep in the mind these other points when you check spelling as you type:

- When the Check Spelling While Typing option is selected, a wavy red line appears under misspelled words in text objects and caption fields only when the insertion point (vertical, flashing bar used for entering text) is placed in or directly next to text.
- You can change the color of the wavy underline by selecting Tools > Options > Spelling > Wavy Underline Color.
- You can quickly correct spelling errors by right-clicking a misspelled word and selecting one of the suggested words from the context menu. Designer displays a maximum of four alternate spelling suggestions on the context menu for each misspelled word. To see more than four suggestions, use the Check Spelling dialog box.

To set up Designer to identify spelling errors as you type

1) Select Tools > Options and, in the Spelling panel, select Check Spelling While Typing.
2) To correct a spelling error while typing, place the insertion point in or directly next to the underlined word, and right-click and perform one of these actions:
   - To correct the misspelled word, select one of suggested words in the context menu. If no spelling suggestions are available for the misspelled word, the text *no spelling suggestions* appears instead.
   - To add the word to My Custom Dictionary, select Add to Dictionary. This selection adds the underlined word to all languages in the dictionary. Designer no longer identifies the word as being misspelled.
   - To disregard all occurrences of the misspelled word, select Ignore All. This selection adds the word to the list of ignored words, and Designer temporarily does not identify the word as being misspelled. All spell checks disregard the word until you restart Designer. The list of ignored words is cleared each time Designer is started.

RELATED LINKS:

To select the default locale

To check the spelling in a form

To edit the custom dictionary

Options dialog box

To check the spelling in a form

You can check the text for spelling errors in selected objects or in an entire form by using the Check Spelling dialog box.

By default, Designer checks the spelling in all of the text associated with the various objects in a form, such as text, captions, tool tips, custom screen reader text, and list box entries. However, you can control the type of text that Designer checks by selecting options in the Spell Check list available in the Spelling panel of the Options dialog box. When you select the Static Text and Captions options, you can also select specific text or captions in the form to spell check. If the Static Text and Captions options are not selected, Designer will not check the spelling in any selected text objects or caption fields.

When checking the spelling in a form, Designer always refers to the internal dictionary and your custom dictionary (My Custom Dictionary). You can also add other custom dictionary files (*.clam) to expand the

list of correctly spelled words to which Designer refers. For more information, see *Adding a word to the list of ignored words*.

TIP: *To see a list of the dictionaries you currently have available, select Tools > Options, and click Spelling. The Spelling panel displays a list of the installed dictionaries. Notice that My Custom Dictionary is selected at all times and cannot be removed.*

1) (Optional) If you want to check the spelling in specific text objects or caption fields, press Ctrl and select the necessary objects.
2) Select Tools > Check Spelling. The Check Spelling dialog box appears with the first misspelled word highlighted in red under Not in Dictionary.
3) To remove spelling errors, perform one of these actions:
   - To correct the misspelled word, select one of the words in the Suggestions list, and then either click Change to replace this occurrence of the misspelled word or click Change All to replace all occurrences of the misspelled word. If the word you want is not in the list of suggested words, you can type the correct spelling directly in the Check Spelling dialog box, and then click Change or Change All as needed. If you clicked Change or Change All and want to revert the changes, click Undo.
   - To disregard this one occurrence of the misspelled word, click Ignore Once.
   - To disregard all occurrences of the misspelled word, click Ignore All. This selection adds the word to the list of ignored words, and Designer temporarily does not identify the word as being misspelled. All spell checks disregard the word until you restart Designer. The list of ignored words is cleared each time Designer is started.
   - To add the word to My Custom Dictionary, click Add to Dictionary. This selection adds the highlighted word to all languages in the My Custom Dictionary and Designer no longer identifies the word as being misspelled.

   After you correct a spelling error, the Check Spelling dialog box automatically advances to the next misspelled word. A message is displayed when Designer completes the spell check.

RELATED LINKS:

To edit the custom dictionary

Check Spelling dialog box

Options dialog box

To add a word to all languages in the custom dictionary

When you spell check a form, Designer may identify a word as being misspelled when the word is actually spelled correctly. For example, Designer generally will not recognize company or industry-specific product terminology and acronyms. As a result, Designer provides each user with a custom dictionary (My Custom Dictionary) where you can create a customized list of properly spelled words. My Custom Dictionary is listed under Installed Dictionaries in the Spelling panel in the Options dialog box and is always selected by default. Designer refers to My Custom Dictionary whenever you spell check a form; therefore, you cannot delete My Custom Dictionary.

The words that you add to the custom dictionary are automatically added to all languages (locales). Designer considers the word to be spelled correctly regardless of the locale that is currently applied to

the form or to a selected object. However, if the word you want to add to the Custom Dictionary does not belong to all languages, you can use the Edit Dictionary dialog box to add the word to selected languages. After you add a word to the Custom Dictionary, Designer recognizes the spelling of the word as correct and removes the wavy red line from under the word.

1) Right-click the misspelled word and select Add to Dictionary.

To add a word to selected languages in the custom dictionary

If the word you want to add to your custom dictionary (My Custom Dictionary) does not belong to all languages, you can use the Edit Dictionary dialog box to add the word to selected languages. After you add a word to My Custom Dictionary, Designer recognizes the spelling of the word as correct for the selected language and removes the wavy red line under the word.

1) Select Tools > Edit Dictionary.
2) In the Language list, select the language you want.
3) In the Word box, type the correct spelling of the word you want to add.
4) Click Add.
5) Repeat steps to add the word to another language.

RELATED LINKS:

To add a word to all languages in the custom dictionary

To edit the custom dictionary

Edit Dictionary dialog box

To edit the custom dictionary

From time to time, you may want to edit your custom dictionary (My Custom Dictionary) or review the contents. In the Edit Dictionary dialog box, you can select options from the Language list to view the words that are associated with all languages or individual languages. You can also change the spelling of any listed word, add words to all or selected languages, and delete words from all or selected languages.

To open the Edit Dictionary dialog box

1) Select Tools > Edit Dictionary.

RELATED LINKS:

To add a word to all languages in the custom dictionary

To add a word to selected languages in the custom dictionary

Options dialog box

Edit Dictionary dialog box

Adding a word to the list of ignored words

Sometimes Designer may identify a word as being misspelled when it is actually spelled correctly. However, in such cases you may not always want to add the word to your custom dictionary. As an alter-

native, you can create a list of words that Designer ignores temporarily when spell checking a document. Designer recognizes the spelling of ignored words as being neither correct nor incorrect, and disregards them when performing a spell check.

You cannot view or edit the list of ignored words. Therefore, it is a good idea to remember the words you add. The list of ignored words remains in place until you close Designer; each time Designer is started, the list is cleared.

You can quickly add a word to the list of ignored words while typing in one of the fields in a form or in the Object or Hierarchy palettes by using the commands on the context menu (right-click). Using the buttons in the Check Spelling dialog box, you can also add a word to the list of ignored words while spell checking in a form.

To add a word to the list of ignored words while typing

1) Right-click a misspelled word in any field in a form or in the Object or Hierarchy palette and select Ignore All. Designer temporarily disregards all instances of the word as being misspelled until Designer is restarted.

To add a word to the list of ignored words while spell checking a form

1) Select Tools > Check Spelling. The Check Spelling dialog box appears with the first misspelled word highlighted in red under Not in Dictionary.
2) Perform one of the these actions:
   – To disregard only this occurrence of the misspelled word, click Ignore Once. Designer ignores the word and advances to the next misspelled word.
   – To disregard all occurrences of the misspelled word, click Ignore All. Designer ignores all instances of the misspelled word and advances to the next misspelled word. All spell checks disregard all occurrences of the misspelled word until you restart Designer. The list of ignored words is cleared each time Designer is started.

RELATED LINKS:

To add a word to selected languages in the custom dictionary

To edit the custom dictionary

Check Spelling dialog box

Edit Dictionary dialog box

To add a dictionary

You can add additional dictionary files to Designer to suit your needs. That is, if you want to expand the list of correctly spelled words that Designer refers to when spell checking a form, you can add one or more custom dictionary files (*.clam) for a specific language or for all supported languages. For example, you can add a custom dictionary of unique terms for specialized industries such as medicine, law, engineering, insurance, or finance. Keep in mind that if you want to add the words in a custom dictionary file to a particular language instead of all languages, you must include the correct language extension in the custom dictionary file name. Let's say you want to add a custom dictionary file to French Canadian only,

you must add *fr_CA* to the file name like this, *[file name]*-fr_CA.clam. If you do not include a language extension in the file name, the words in the file will be considered correct for all languages.

NOTE: You can also add more standard dictionary files (*.lex) for languages that Designer does not already support, by manually adding the files to the folder located at \Program Files\Common Files\Adobe\Linguistics\Providers\Proximity.

You can use the options in the Spelling panel of the Options dialog box to add dictionaries and select which dictionaries are used when performing spell-check operations.

1) Select Tools > Options.
2) Select Spelling from the list on the left and then click Add a Dictionary. The Custom Dictionary File dialog box appears.
3) Browse to the dictionary file you want to add to Designer and click Open. The dictionary is added to the list of installed dictionaries.

RELATED LINKS:

To remove a dictionary

To edit the custom dictionary

Options dialog box

Edit Dictionary dialog box

To remove a dictionary

The Installed Dictionary list in the Spelling panel of the Options dialog box shows the dictionaries currently available for spell checking a form. You can delete one or more of the custom dictionaries you no longer need. When you delete a custom dictionary, it is removed from the list of installed dictionaries. The actual dictionary file (*.clam) is not removed from your computer's file system.

NOTE: You cannot remove My Custom Dictionary because Designer always refers to this dictionary whenever you spell check a form.

1) Select Tools > Options.
2) In the Spelling panel, select the dictionary you want to delete from the Installed Dictionaries list.
3) Click Remove a Dictionary.

RELATED LINKS:

Adding a word to the list of ignored words

To edit the custom dictionary

Options dialog box

3.8. Hyphenate text

Use hyphenation to improve text alignment in a given area by reducing the amount of white (empty) space between the last word on a line and the right margin. If a word is too long to fit entirely on a single line, the word is hyphenated at the proper hyphenation point, which forces a line break as close to the right margin as possible. Hyphenation makes each line of text approximately the same length to give the

text a more uniform layout. Designer uses a hyphenation dictionary, metrics such as line spacing and font size, as well as other linguistic information to determine where various words can be legally and optimally hyphenated.

You can hyphenate the text in text objects, in the caption area of objects such as text fields, decimal fields, and numeric fields, and in the value area of text field objects (default text and text the form filler enters). For example, you can indicate the number of letters to allow in a word before it can be hyphenated; hyphenate capitalized words, such as the first word of a sentence; hyphenate words that are all capital letters, such as acronyms; and add or remove all hyphenation from the form.

You can set default hyphenation options for all new forms or customize hyphenation settings for individual forms.

Set hyphenation options in these areas:

Options dialog box (Formatting panel)

Use the options in this dialog box to specify default hyphenation settings for all new forms. Changing theses options does not affect the currently opened form.

Forms Properties dialog box (Formatting tab)

Use the options in this dialog box to specify hyphenation settings for the currently opened form only.

Paragraph palette (Hyphenation option)

Use this option to enable or disable hyphenation in individual objects.

NOTE: When you create a form, the Form Properties dialog box automatically inherits the default hyphenation settings for new forms from the Tools Options dialog box. To create a form that has different hyphenation settings or to change the default settings in an existing form, use the Form Properties dialog box. The settings in the Form Properties dialog box override the settings in the Options dialog box for the current form. When you change the hyphenation settings in the Forms Properties dialog box for the current form, the settings in the Options dialog box (for new forms) do not change.

RELATED LINKS:

Hyphenation in selected paragraphs

Paragraph properties in the Paragraph palette

Formatting (Options dialog box)

Formatting tab (Form Properties dialog box)

Considerations for setting hyphenation

Keep the following points in mind regarding the various options you must select for objects that contain hyphenated text:

Hyphenate option in the Paragraph palette

Although the hyphenation you select in the Form Properties and Options dialog boxes apply to the entire form, you can use the Hyphenate option in the Paragraph palette to add or remove hyphenation in individual objects. Using the Hyphenate option, you can manually adjust text layout on an object-by-object basis.

The Hyphenate option is available in the Paragraph palette only when you select the Using the Allow Hyphenation in Text and Field Captions option or the Allow Hyphenation in Text Field Values option, or both on the Form Properties (Formatting tab) or the Options (Formatting panel) dialog boxes. The Hyphenate option displays a colored square when these two options are in a mixed state, where either Allow Hyphenation in Text and Field Captions or Allow Hyphenation in Text Field Values is deselected. That is, when you select the Hyphenate option for an object when these two options are in a mixed state, the check box displays a colored arrow, which changes to a colored square to remind you that one of the options is deselected.

TIP: *When Hyphenate New Items is selected in the Options dialog box on the Formatting panel, the Hyphenate option is automatically selected with new objects added to a form.*

Currently Editing palette menu

The commands in the Currently Editing palette menu (Edit Caption or Value, Edit Caption, or Edit Value) determine when the Hyphenate option is available in the Paragraph palette (does not apply to text objects). That is, the Allow Hyphenation in Text and Field Captions option and the Allow Hyphenation in Text Field Values option are applied according to the command you select. For example, if you select the Allow Hyphenation in Text and Field Captions option, you must also select either the Edit Caption or Value command or the Edit Caption command to make the Hyphenate option available.

Use these commands to select the area (caption, value, or both) to hyphenate in each object that contains hyphenated text. For example, to hyphenate the text in text objects and the captions in text field objects, select the Allow Hyphenation in Text and Field Captions option and either the Edit Caption and Value menu command or the Edit Caption command. Alternatively, to hyphenate the default or user input text in the value area of text objects, select the Allow Hyphenation in Text Field Values option and the Edit Value menu command. The default command for text fields, decimal fields, numeric fields, and signature fields is Edit Caption or Value.

Expand to Fit options

It is recommended that you select the Expand to Fit (Height) option for each object that contains hyphenated text. When a word is too long to fit entirely on a line, hyphenation forces a line break and divides the word over two lines. As a result, the object that contains the text needs to expand in height to accommodate additional lines when needed. However, you can also manually enlarge objects to the correct size where hyphenation can occur. Text objects expand in width (as needed) when you enter text, even if you do not select the Expand To Fit (Width) option.

Allow Multiple Lines option

(Applies to the value area of text objects only) Select this option on the Field tab of the Object palette for each text object to hyphenate the text in, in the value area of text field objects. This option enables the text to break onto more than one line where the height of the field permits.

RELATED LINKS:

Hyphenation in selected paragraphs

Paragraph properties in the Paragraph palette

Formatting (Options dialog box)

Formatting tab (Form Properties dialog box)

Setting and removing hyphenation in forms

You can hyphenate the text in, and remove hyphenation from, text objects, the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, and the value area of text field objects (default text and text the form filler enters).

To set hyphenation for new forms

Use the Formatting panel in the Options dialog box. The numbers you enter and the options you select on the Formatting panel are used as default settings in the Form Properties dialog box on the Formatting tab.

To set hyphenation for individual forms

Use the Formatting tab in the Form Properties dialog box. The numbers you enter and the hyphenation options you select on the Formatting tab apply to the current form only. You can change the hyphenation settings for the current form to suit your needs. The default settings that initially appear in the Form Properties dialog box are inherited from the Formatting panel in the Options dialog box, which automatically apply to all new forms. If you change the default settings in the Form Properties dialog box, the new settings override the settings in the Options dialog box for the current form. The hyphenation options in the Options dialog box do not change.

NOTE: Before you set hyphenation for forms, it is recommended that you become familiar with the different options you need to select for objects that contain hyphenated text. (See *Hyphenate text*)

To remove hyphenation from forms

Use the You can remove hyphenation from the text in text objects, in the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, and in the value area of text field objects.

To set hyphenation for a new form

1) Select Tools > Options.
2) Click Formatting, enter the values, and select the options to apply to new forms:
   - In the Words With At Least <*x*> Letters box, enter the minimum number of letters that a word must contain to be hyphenated.
   - In the After First <*x*> Letters box, enter the minimum number of letters in a word that must appear on a line before the hyphen.
   - In the Before Last <*x*> Letters box, enter the minimum number of letters in a word that must appear on the next line, after the hyphen.

- To hyphenate words that begin with a capital (uppercase) letter, such as the first word of a sentence, select Hyphenate Capitalized Words.
- To hyphenate words that are all capital letters, such as acronyms, select Hyphenate Words in ALL CAPS.
- To hyphenate the text in text objects, and in the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, select Allow Hyphenation in Text and Field Captions.
- To hyphenate the text in the value area of text field objects (default text and text entered by the person filling the form), select Allow Hyphenation in Text Field Values.
- Select Hyphenate New Items to hyphenate the text in new objects added to the form.

For descriptions and example usage of the above options, see Formatting (Options dialog box).

3) Click OK.

To set hyphenation for an individual form

1) Select File > Form Properties.
2) Click the Formatting tab, and enter the values and select the options to apply to the current form:
   - In the Words With At Least <*x*> Letters box, enter the minimum number of letters that a word must contain to be hyphenated.
   - In the After First <*x*> Letters box, enter the minimum number of letters in a word that must appear on a line before the hyphenation point.
   - In the Before Last <*x*> Letters box, enter the minimum number of letters in a word that must appear on the next line after the hyphenation point.
   - To hyphenate words that begin with a capital (uppercase) letter, such as the first word of a sentence, select Hyphenate Capitalized Words.
   - To hyphenate words that are all capital letters, such as acronyms, select Hyphenate Words in ALL CAPS.
   - To hyphenate the text in text objects, and in the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, select Allow Hyphenation in Text and Field Captions.
   - To hyphenate the text in the value area of text field objects (default text and text entered by the person filling the form), select Allow Hyphenation in Text Field Values.

   For descriptions and example usage of the above options, see Formatting (Form Properties dialog box).
3) Click Hyphenate All Text.

   TIP: *Click Edit > Undo Hyphenate All text to immediately undo hyphenation.*

To remove hyphenation from a form

1) Select File > Form Properties.
2) Click the Formatting tab and do one of the following actions:

- To remove hyphenation from the text in text objects and in the caption area of objects, deselect Allow Hyphenation in Text and Field Captions.
- To remove hyphenation from the text in the value area of text field objects, deselect Allow Hyphenation in Text Field Values.
- To remove all hyphenation, click Remove All Hyphenation.

3) Click OK.

TIP: *To immediately restore all hyphenation, click Edit > Undo Remove All Hyphenation."*

3.9. Hyperlinks

Use hyperlinks to provide links to external websites, email addresses, and PDF and HTML files. You can insert URL and email hyperlinks within static text objects (including floating fields) or within the caption area of objects such as text field, image field, and drop-down list objects.

When using hyperlinks with Dynamic XML forms, you must select target version as Acrobat and Adobe Reader 9 or later.

NOTE: Designer cannot anticipate the content of run-time data. If you intend to populate a form with rich text that contains hypertext links, you must set the target version to Acrobat and Adobe Reader 9.0 or later.

To render an XDP form design to HTML when the hyperlinks point to resources in a AEM forms application, use the correct notation and check the file in to the forms server. For the hyperlinks, use the http://<AEM forms server name>:<port number>/repository/Applications/<application name>/<version>/<folder>/<filename> notation. In Workbench, check in the form design to the same forms server and location.

NOTE: The hyperlink menu commands are not available with button objects such as Print and Reset, and with read-only text.

To insert a hyperlink

1) Select the text where you want to insert a hyperlink.
2) Click Insert > Hyperlink.
3) Do one of the following tasks:
   - Select URL, and either type or select a valid website address, or click the browse button to select a file located on your computer. When you select a file, the relative path for the document appears in the URL box.
   - Select Email and type one or more valid email addresses and, optionally, a subject line.
4) Click OK. You can also use the Undo Hyperlink and Redo Hyperlink commands in the Edit menu to quickly revert text or a hyperlink to its former state.

To remove hyperlink

1) Place the insertion point within the hyperlink, right-click, and select Remove Hyperlink.

   TIP: *You can also use the Undo Hyperlink and Redo Hyperlink commands in the Edit menu to quickly revert text or a hyperlink to its former state.*

To edit a hyperlink

1) Place the insertion point within the hyperlink, right-click, and select Edit Hyperlink.
2) Make the necessary changes, and click OK.

To test a hyperlink

After you insert a hyperlink, it is recommended that you test it to make sure it opens correctly.

You can test links on either the Design View tab or the PDF Preview tab. On the Design View tab, when you move the pointer over a hyperlink, the name of the associated object and the link are displayed in a tool tip. On the PDF Preview tab, when you move the pointer over a hyperlink, only the associated link is displayed in a tool tip. Each link opens the appropriate program, such as a web browser or an email program.

1) Place the insertion point within the hyperlink, right-click, and select Open Hyperlink.

   TIP: *As a shortcut, you can use Ctrl+click to quickly open links.*

3.10. Prepare for translation

To prepare a form design for translation, generate XLIFF identifiers for the translatable text. In Designer, the identifiers are generated when you select the Create Translation identifiers When Saving option. When selected, the option generates a unique XLIFF ID for each text string the first time you save the form design. The XLIFF identifiers are visible in the XML source when you save a form design in the Adobe XML Form (.xdp) format.

When you modify and save the form design, Designer compares the identifier, and string combinations to determine whether any changes were made since the last time the form design was saved. Changes can include changed text, field deletion, or field insertion. If the text changed since it last saved, Designer updates the text when you copy an object that already has an identifier so that a duplicate identifier is not created.

Designer includes two examples of Extensible Stylesheet Language Transformations (XSLT) files. The *extractstrings.xslt* file extracts the XLIFF IDs and the *mergestrings.xslt* file creates a new version of the form design in the new language. The examples are installed with Designer in the installation directory under \...\FormTranslation.

RELATED LINKS:

Translating Forms using XLIFF

To create translation identifiers

1) Select Tools > Document Handling.
2) In the File Options area, select Create Translation Identifiers When Saving.
3) Create and save the form design.

Extracting the translation identifiers and translatable text

You can use the example extractstrings.xslt file as a starting point to extract the XLIFF identifiers and text that must be translated from the form design.

Pass the form design as the input to the extractstrings.xslt style sheet by using an XSLT processor, such as a free or commercial version of the Saxon XSLT processor. Optional and required arguments are available. The required arguments describe how to execute a given XSLT. The optional arguments contain metadata.

For example, to extract the XLIFF identifiers and text from the `MyForm.xdp` file, enter the following required arguments at the command prompt to generate the `MyForm.s2x` file:

```
java -jar saxon8.jar MyForm.xdp extractstrings.xslt > MyForm.s2x
```

`MyForm.xdp` is the name of the form design that you extract the XLIFF identifiers and text from, and `MyForm.s2x` is the file that you send to be translated.

Make sure that you keep a backup of the XLIFF file. Then, if you must edit the form design, you can easily determine which strings must be translated by comparing the XLIFF backup file with the XLIFF file for the changed form design.

NOTE: The `saxon8.jar` file can have a different name, depending on the version of Saxon being used. An example is `saxon9.jar`.

extractstrings.xslt optional parameters

You can specify the following optional parameters that are included in the S2X metadata.

Parameter	Description
locale	Locale of the form design
xdpFilename	Name of the form design that you extract the XLIFF identifiers and text from
develContact	Email address of the contact for the form design
collection	Name of the package
domain	Domain name

Each parameter has a default:

```
<xslt:param name="locale" select="'en'" />
<xslt:param name="xdpFilename" select="'mytemplate.xdp'" />
<xslt:param name="develContact" select="'developer@mycompany.com'" />
```

```
    <xslt:param name="collection" select="'package'" />
    <xslt:param name="domain" select="'BC'" />
```

Here is an example:

```
java -jar saxon8.jar input.xdp extractstrings.xslt xdpFilename=MyForm.xdp
locale=en_CA > MyForm.s2x
```

Creating the form design with the translated text

After you receive the translated XLIFF file, enter the following string at the command prompt to create the translated form design:

```
java -jar saxon8.jar MyForm.s2x mergestrings.xslt xdpFile=MyForm.xdp >
MyFormTranslated.xdp
```

`MyForm.s2x` is the translated XLIFF file, `MyForm.xdp` is the name of original form design, and `MyFormTranslated.xdp` is the translated form design.

The result is a master form design (`MyForm.xdp`) and a translated form design (`MyFormTranslated.xdp`) that you can make available to users.

If you must change the form design, edit the master form design and then repeat the steps above to produce a new XLIFF file.

We provide the example `mergestrings.xslt` style sheet, which creates a new version of the form design in the new language.

RELATED LINKS:

Prepare for translation

3.11. Macros

Macros provide an external plug-in interface, to extend the functionality of Designer. For example, you can run a macro to rename a field and update all associated script references, or to find scripts that consist entirely of comments.

NOTE: You should only run a macro if you trust the author of the script.

A macro is a JavaScript file (JS). You create JavaScript files in a JavaScript editor and run the scripts in Designer. The JavaScript in the macro has full access to the template model. In addition to the template DOM, there is an object in the root namespace called designer. The designer object provides methods that you can use to communicate directly with Designer. For example, one method allows you to launch a SWF dialog box and exchange strings with it, which allows you to build a custom user interface.

For more information about available scripting methods, see the Scripting Reference.

To set up macros for use in Designer, you create a subfolder structure for JavaScript files in the Designer installation folder, and run the JavaScript files from the Macros menu (Tools > Macros).

To set up macros for use in Designer

1) Do one of the following actions:
   - For common macros (all languages), in the Designer installation folder, create a subfolder called macros. For example, <DesignerInstallationFolder>\macros.
   - For language specific macros, in the Designer installation folder, create a subfolder for each language (locale), and then create a macros folder in each locale folder. For example, <DesignerInstallationFolder>\EN\macros.
2) In each macros folder, create one or more subfolders for JavaScript (JS) and macro.xml files.
3) Save JavaScript and macro.xml files to the appropriate subfolders. Place any SWF files used by the macro JavaScript files in the same subfolder.
4) In Designer, click Tools > Macros. The macros (common and language specific) are listed on the Macros menu.

About macro.xml configuration files

You create an macro.xml configuration file to rename the command that appears on the Macros menu for each JavaScript file in a subfolder. You add one macro.xml file to each subfolder containing one or more JavaScript files.

Notice the label and script tags in the following example macro.xml file. The label tag encloses the name of the command (Merge) that appears on the Macros menu. The script tag encloses the name of the associated JavaScript file (mergenodes.js).

Example macro.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<designerMacros>
<!-- one or more macros that are in the same directory can be specified
in the same macro.xml config file -->
<macro>
<!-- used as menu command text -->
<label>Merge</label>
<!-- A macro key can refer to only one script file -->
<script>mergenodes.js</script>
</macro>
</designerMacros>
```

Using macro.xml configuration files is optional. When Designer refreshes the list of macros, it searches the macros subfolders for macro.xml files. If Designer does not locate a macro.xml file in a subfolder, the names of the JavaScript files appear on the Macros menu.

Organizing the macros subfolder

Create a least one subfolder under the macros folder to contain your JavaScript files. If you save JavaScript files directly in the macros folder, they do not appear on the Macros menu in Designer.

You can organize macros subfolders different ways. The method you choose, depends on your needs. Consider the number of JavaScript files you have, the number of macro.xml files to create, and the order of commands on the Macros menu.

One way to organize the macros subfolders is to save all JavaScript files in one subfolder. This way, you use one macro.xml file to specify the names of all JavaScript files in the folder, and names of the associated Macros menu commands. The order of the commands on the Macros menu, is the same as the order of macros listed in the macro.xml file.

Example macro.xml file containing multiple macros:

```
<?xml version="1.0" encoding="UTF-8"?>
<designerMacros>
<macro>
<label>Get Designer Locale</label>
<!-- Description - Get Designer locale -->
<script>getLocale.js</script>
</macro>
<macro>
<label>Hello World</label>
<!-- Description - Display Hello World -->
<script>helloWorld.js</script>
</macro>
<macro>
<label>Highlight Text</label>
<!-- Description - Highlight text areas -->
<script>HighlightFields.js</script>
</macro>
<macro>
<label>Refactor</label>
<!-- Description - Refactor -->
<script>refactor.js</script>
</macro>
<macro>
<label>Show Flex Sample</label>
<!-- Description - Show Flex Sample -->
<script>showFlexSample.js</script>
</macro>
</designerMacros>
```

Another way to organize the macros subfolders is to save JavaScript files in separate subfolders. You use a macro.xml file in each subfolder to specify the name of the JavaScript files, and name the related Macros menu commands. The order of the commands on the Macros menu, is the same as the order of subfolders and the macros listed in each macro.xml file.

Keep in mind that all the subfolders you create for JavaScript files must be a child of the macros parent folder (macros\MyMacros). Designer does not scan subfolders below the level of the first subfolder.After you add a JavaScript file to a macros subfolder, the name of the file appears under the Macros command on the Tools menu. If you want to display a name other than the JavaScript filename, you can create an XML configuration file named macro.xml to change the name.

Localizing macros

You create a subfolder structure in the Designer installation folder for the JavaScript files associated with macros. The subfolders structure you create depends on whether you plan to localize macros or not.

If you do not need to localize macros, because they are common to all languages, you create a subfolder named macros in the Designer installation folder. For example, c:\program files\Adobe\Designer\macros\. You then create one or more subfolders in the macros subfolder for the JavaScript files you want to run. You can create one subfolder for all JavaScript files or create a separate subfolder for each JavaScript file. Designer loads the JavaScript files in these subfolders, regardless of the locale option selected in the Form Locale list (Form Properties dialog box > Defaults panel).

If you need to localize macros, because they are locale (language) specific, you create a subfolder in the Designer installation folder for each locale. You then create a macros subfolder in each locale subfolder, and create one or more subfolders in the locale subfolder for the Javascript files. For example: c:\program files\Adobe\Designer\EN\macros\. Designer loads the JavaScript files in these folders based on Designer's application language.

Macro Logging

If you are unsure which macros Designer is loading, you can check the macros log file.

Designer creates a log file called MacrosLog.log in the Designer application data folder located here: C:\Documents and Settings\<username>\Application Data\Adobe\Designer\<Designer version number>.

The MacrosLog.log file lists the macros loaded in Designer.

4. Test and troubleshoot

4.1. Previewing and testing forms

In the Preview PDF tab, you can view and test a form design or template as a PDF form by using Acrobat or Adobe Reader.

The Preview PDF tab appears only when Acrobat or Adobe Reader is installed. By default, if you have both Acrobat and Adobe Reader installed, Designer starts Acrobat automatically to preview the form. To use Adobe Reader to preview the form, you must start it before you click the Preview PDF tab.

If you change the File > Form Properties > Defaults > Target Version setting to a specific version of Acrobat and Adobe Reader, you must ensure that you have the version of Acrobat installed that matches the target version that is saved; otherwise, you may get an error or a warning message when you click the Preview PDF tab.

The preview will use the same format as the saved form. You can right-click the PDF Preview tab to see which default file type option is currently selected. To indicate the format of an unsaved form, you must change the Tools > Options > Document Handling setting.

If you are designing an interactive or printable form, you can change the File > Form Properties > Preview > Preview Type setting to set up your Preview PDF correctly.

If you are designing forms for use with Forms, the same form design can be used to render PDF forms. Although you can preview the form design on the Preview PDF tab, the HTML form may not appear the same as it does when the form is rendered by using Forms. If you are creating an HTML form, render the form by using Forms and preview it in a web browser.

RELATED LINKS:

Addressing warning messages in the Report palette

To preview and test forms in the Preview HTML tab

Besides previewing the PDF rendition of forms, you can also preview the format in HTML rendition while designing the form in Designer. Preview HTML tab can be used to preview form as it would appear in a browser. For more information, see Preview your XDP form in HTML.

To preview your forms in HTML, provide configuration details of your forms server:

1) Go to Tools > Options.
2) In the Options window, select Server Options page and provide following details:
   - **Server URL:** URL of the AEM forms Server.
   - **HTTP port number:** Port number at which the AEM forms server is running. The default value is 8080.

- **HTML Preview Context:** Path of the profile to use for rendering XFA forms. The default value is `lc/content/xfaforms/profiles/default.html`. The default profile is used to preview the form in designer and to use any other custom profile, provide appropriate path of the custom profile.
- **forms manager Context:** Context path at which Forms Manager UI is deployed. The default value is `lc/fm`.

NOTE: Ensure that AEM forms is available at the server URL configured above. The HTML preview connects to the CRX server to generate a preview.

To preview and test forms in the Preview PDF tab

Use the Preview PDF tab to preview a form design as it would appear in Acrobat or Adobe Reader. Before previewing a form design, ensure that you have set the appropriate options in the Form Properties dialog box for previewing the specific type of form.

You can right-click the Preview PDF tab to see which default preview type option is currently selected.

To display the Acrobat or Adobe Reader toolbars in the Preview PDF tab, press ALT+F8.

NOTE: To preview a form in the Preview PDF tab, ensure that the Display In Browser option in Acrobat is selected. Ensure that you have the version of Acrobat or Adobe Reader installed that matches the target version saved; otherwise, you may get an error or warning message when you click the Preview PDF tab.

1) Choose one of the following ways to display the Preview PDF tab:
   - Select View > Preview PDF.
   - In the Layout Editor, click the Preview PDF tab.
2) If the form is interactive, test the objects on the form to ensure they are functioning as expected.
3) (Optional) If the form is non-interactive and being merged with data, you may also want to test the form with a sample data file to ensure that fields are appropriately mapped to the data source.

NOTE: You must use Acrobat to test the Web Services and Database Connectivity features. These features require a rights-enabled PDF form, but you cannot set usage rights for a PDF form for previewing in Designer.

RELATED LINKS:

Addressing warning messages in the Report palette

To set preview options for an interactive form

To preview an interactive form, you need to set the appropriate options in the Form Properties dialog box.

1) Select File > Form Properties.
2) Click the Preview tab and, in the Preview Type list, select Interactive Form.
3) (Optional) To test the form you are previewing by using a data source that you created, enter the full path to your test data file in the Data File box. You can also use the browse button to navigate to the file.

4) (Optional) To test the form you are previewing by using an automatically generated data source, click Generate Preview Data. You can use the browse button to navigate to the location where you want the file saved. If the form contains repeating subforms or subform sets, indicate the number of times each subform or subform set will repeat in the data file.
5) In the Preview Adobe XML Form As list, select either Static PDF Form or Dynamic XML Form.

To set preview options for a non-interactive form

To preview a non-interactive form, you need to set the appropriate options in the Form Properties dialog box.

1) Select File > Form Properties.
2) (Optional) To test the form that you are previewing as if it were printed on one side of the paper, click the Preview tab and, in the Preview Type list, select Print Form (One-sided).
3) (Optional) To test the form that you are previewing as if it were printed on both sides of the paper, click the Preview tab and, in the Preview Type list, select Print Form (Two-sided).

   NOTE: *When you select Print Form, all objects are non-interactive.*
4) (Optional) To test the form that you are previewing by using a data source, enter the full path to your test data file in the Data File box. You can also use the browse button to navigate to the file.
5) (Optional) To test the form that you are previewing by using an automatically generated data source, click Generate Preview Data. You can use the browse button to navigate to the location where you want the file saved. If the form contains repeating subforms or subform sets, indicate the number of times each subform or subform set will repeat in the data file.
6) In the Preview Adobe XML Form As list, select either Static PDF Form or Dynamic XML Form.

To preview a form using sample data

Designer lets you preview and test your form by using sample XML data. It is recommended that you frequently test your form with sample data to ensure that the form renders correctly.

If you do not have sample data, Designer can create it, or you can create it yourself. (See *To automatically generate sample data to preview your form* and *To create sample data to preview your form*.)

Testing your form by using a sample data source ensures that the data and fields are mapped and that repeating subforms repeat as you expected. You can create a balanced form layout that provides the appropriate space for each object to display the merged data.

1) Select File > Form Properties.
2) Click the Preview tab and, in the Data File box, type the full path to your test data file. You can also use the browse button to navigate to the file.
3) Click OK. The next time you preview the form in the Preview PDF tab, the data values from the sample XML file will appear in the respective objects.

For detailed information about each option in the Preview tab, see *Preview (Form Properties dialog box)*.

RELATED LINKS:

To create a data connection to an XML schema

To automatically generate sample data to preview your form

You can generate sample data to preview and test your form instead of creating a sample data file. Also, if your form contains repeating subforms or subform sets, you can specify the number of times the data will be repeated when you preview the form.

Designer generates sample data that is valid for the corresponding objects in the form, with a few exceptions:

- Sample data is not generated according to any validation scripts that might be specified for an object.
- The minimum and maximum count for a subform will restrict the number of repeating subforms that you specify for the generated sample data file.
- The default value you select for a 2D barcode is retained in the generated sample data file.

After you generate the sample data file, you can edit the file, if required.

Specify which data file to use when you preview the form. The next time you preview the form in Designer, the sample data will appear in the respective objects.

1) In Designer, select File > Form Properties > Preview tab.
2) (Optional) To test the form that you are previewing as an interactive form, in the Preview Type list, select Interactive Form
3) (Optional) To test the form that you are previewing for single-sided printing, in the Preview Type list, select Print Form (One-sided).
4) (Optional) To test the form that you are previewing for double-sided printing, in the Preview Type list, select Print Form (Two-sided).
5) Click Generate Preview Data.
6) In the Generate Preview Data dialog box, type the full path, including a file name, for the test data file. You can also use the browse button to navigate to the location in the Data File box.
7) In the Repeating Elements list, select the number to the left of a subform and type the number of times it will repeat in the data file.
8) Click Generate.

To create sample data to preview your form

If you do not want to use a sample data file that Designer automatically creates, you can create a sample data file manually.

1) Save the form design as a PDF file.
2) Open the PDF file and enter values in the fields you want to test.
3) In Acrobat, select Advanced > Forms > Export Form Data.
4) In the Export Form Data As dialog box, name and save the file as type XML Data Package (*.xdp).
5) In Designer, select File > Form Properties > Preview tab.
6) (Optional) To test the form that you are previewing as an interactive form, in the Preview Type list, select Interactive Form

7) (Optional) To test the form that you are previewing for single-sided printing, in the Preview Type list, select Print Form (One-sided).
8) (Optional) To test the form that you are previewing for double-sided printing, in the Preview Type list, select Print Form (Two-sided).
9) Click the folder button next to the Data File box and browse to the XML file.
10) Select the XML file and click OK.

Considerations for testing form designs with data

When setting up forms to support merged data, analyze the data-merging requirements of the form against the input data. The form design should be created based on the structure of the input data.

Data binding attempts to match each new form node with a data node. When you use explicit bindings, the targets that are defined in the form design take precedence over implicit bindings. When you use implicit (normal) bindings, the following rules apply:

- The relative order of same-named data values or groups is significant.
- The relative order of uniquely named data values or groups is not significant.
- The hierarchy of structure described by data values or groups is significant.

Review the input data and consider preparing a sample data file, or have Designer automatically generate a sample data file for testing purposes.

Sample data may be used to determine the behavior of a form and should not be considered a replacement for thorough testing. To obtain the best results, the form design should be tested with system-generated data. You can generate the sample data as suggested in the following list to determine whether the layout, formatting, content, and behavior of a form responds as expected:

- To verify whether data formatting (for example, font type, font size, and paragraph alignment) is correct, generate data for every field in the form. The data will also let you verify any calculated field values.
- To verify whether field objects are large enough to accommodate all data values, generate data to reach the maximum number of characters permitted per field.
- To verify the operation of multiple-line and expand-to-fit settings, generate multiple lines of data.
- If your form design contains repeating subforms, you can generate repeating data groups to test the rendering of those subforms in the sample data file that you create. If you are using an automatically generated sample data file, you can specify the number of times you want a data group to repeat.
-
- If your form design contains overflow leaders or trailers, generate the data necessary to test every overflow leader or trailer and their occurrence settings. You should generate enough repeating data groups to flow over three pages, which will also let you verify page numbering. Use the sample data (add one repeating data group at a time) to verify how a repeating subform looks when it flows onto a new page.

- To test the data pattern setting for bound data, ensure that all data values are in the same format generated by the system, especially if the syntax of the source data does not match Designer defaults.
- Generate enough data to employ the layout of every master page in the rendered form.

IMPORTANT: Ensure that the form data does not contain hexadecimal values between 0x00 and 0x20, except for carriage return and horizontal tab. These values are invalid XML characters that Forms does not recognize.

Creating a sample data file

The following guidelines will help you to create a simple test file that contains representative input data:

- The input data file must be a valid XML file. For example, a flat file would have elements of this format:

```
<root_node>
<first_node>value</first_node>
<second_node>value</second_node>
...
<last_node>value</last_node>
</root_node>
```

- Compare the flow of the input data to the physical layout of the form. If you are using implicit binding, the names of the data nodes must match the corresponding containers and fields in the form and be presented in the same order as the fill order in the form.
- If the input data file has more levels of nesting compared to the items in the Hierarchy palette, data bindings for all of the nested objects must be set explicitly through the Binding tab in the Object palette.
- In the input data file, look for data that is repeated but not part of every record. This information could possibly be handled on master pages as boilerplate objects or in subforms that repeat the data for unique records only.

RELATED LINKS:

Working with Data Sources

About form designs

Form design layouts

To test for accessibility

You should test your forms by using a variety of assistive technologies to ensure that the forms are accessible to users.

Download demonstration versions of the screen reader software. As the form author, your familiarity with the form may make it difficult to determine whether the information read by the screen reader is sufficient and understandable. If possible, have someone else test your form in this way. To test screen reader results, turn your monitor off and use only the screen reader to navigate and fill the form.

Check the Internet for demonstration versions of screen magnification software.

Ensure that you can fill the form by using only the keyboard, and keep in mind the following questions:

- Are there any operations that cannot be performed?
- Are any operations awkward or difficult to perform?
- Are keyboard mechanisms well-documented?
- Do all controls and menu items have underlined access keys?

When filling the form, take note of these issues:

- Any parts of the form that become invisible, unrecognizable, or difficult to use
- Areas that continue to appear black on a white background
- Form objects that are improperly sized or truncated

4.2. Addressing warning messages in the Report palette

The Report palette displays different warning and error messages relating to aspects of the form design. When a warning message is generated, these details appear in the Warnings tab in the Report palette:

- The affected object
- A description of the warning
- The type of warning
- The target version that supports the object (if applicable)
- The warning code number

The messages in the Warnings tab are automatically updated as you work in the form design. The messages disappear as you fix the errors.

Designer generates these types of warning messages:

Target

Target warning messages appear when you try to use a feature that is not supported in the Acrobat and Adobe Reader target version. For information on addressing target warning messages, see *Target version warning messages*.

Marker

Marker warning messages appear when a problem occurs with an object. These messages are associated with warning symbols, such as a yellow triangle or white “X” within a red circle. For information on addressing marker warning messages, see *Action warning messages*.

Scripting

Scripting error messages occur when issues occur with scripts in the form design. When you click Tools > Check Script Syntax, scripting error messages appear in the Warning tab in the Report palette. For more information about scripting error messages, see *Scripting error messages*.

Target version warning messages

When you use a feature that the selected target version does not support, warning messages appear in the Warnings tab in the Report palette.

If you require a feature, use one of these solutions:

- Update to a target version that supports the feature.
- Save the form as a static PDF form. Earlier versions of Acrobat and Adobe Reader support some features when the form is saved as a static PDF form.
- Fix the warning message.

This table indicates the code for each target warning message, the affected file types, the target warning message that appears in the Description column on the Warning tab of the Report palette, and ways to remove the warning message.

Code	File type	Description	To remove the warning message
2405	Dynamic XML Form	Target version does not support vertical paragraph alignment for field values.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Paragraph palette and reset the paragraph vertical alignment to these default values: • Text Field (single line) - Align Middle *Text Field (multiple line) - Align Top* *Date/Time Field - Align Middle* *Numeric Field - Align Middle* *Decimal Field - Align Middle* *Drop-down List - Align Middle* *Password Field - Align Middle* *List Box - Align Top*
2406	Dynamic XML Form	Legacy (Version 6) Text Formatting does not support vertical paragraph alignment for field values.	Select File > Form Properties click the Compatibility tab and update the text formatting to Version 7, or click the Paragraph palette and reset the paragraph vertical alignment to these default values: • Text Field (single line) - Align Middle *Text Field (multiple line) - Align Top* *Date/Time Field - Align Middle* *Numeric Field - Align Middle* *Decimal Field - Align Middle* *Drop-down List - Align Middle* *Password Field - Align Middle* *List Box - Align Top*

Code	File type	Description	To remove the warning message
2407	Dynamic XML Form	Target version does not support the preOpen event.	Increase the target version to Acrobat and Adobe Reader 9.0 or later, or move the script to a supported event, or delete the script.
2500	Static PDF Form Dynamic XML Form	Target version does not support the check mark shapes for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Object palette, click the Field tab and select Default from the Check Style list.
2501	Static PDF Form Dynamic XML Form	Target version does not support button highlighting for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Object palette, click the Field tab and select Inverted from the Highlighting list.
2502	Static PDF Form Dynamic XML Form	Target version does not support comb fields for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Object palette, click the Field tab, and deselect Comb Of.
2503	Static PDF Form Dynamic XML Form	Target version does not support the Limit Length to Visible Area option for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Object palette, click the Field tab, and deselect Limit Length to Visible Area.
2505	Dynamic XML Form	Target version does not support the Preserve Scripting Changes To Form When Saved option.	Select File > Form Properties, click the Defaults tab and, under Preserve Scripting Changes to Form When Saved, select Manually.
2506	Dynamic XML Form	Target version does not support locking fields after signing.	Select the Signature Field object, click the Object palette, click the Signature tab, and deselect Lock Fields After Signing.
2507	Dynamic XML Form	Target version does not support the indexChange event.	Move the script to a supported event or deleting the script.
2508	Dynamic XML Form	Target version does not support the signature settings.	Select the Signature Field object, click the Object palette, click the Signature tab, and deselect the previously selected options.

Code	File type	Description	To remove the warning message
2509	Static PDF Form Dynamic XML Form	Target version does not support the Visible (Screen Only) option for the current file type and/or object type.	For Static PDF Form (buttons), increase the target version to Acrobat and Adobe Reader 7.0.5 or later. For Dynamic XML Form (all objects except buttons that have right-hand borders), increase the target version to Acrobat and Adobe Reader 9.0 or later.
2510	Static PDF Form Dynamic XML Form	Target version does not support the Visible (Print Only) option for the current file type.	Increase the target version to Acrobat and Adobe Reader 9.0 or later.
2514	Dynamic XML Form	Target version does not support the Enforce Strict Scoping Rules in JavaScript option.	Select File > Form Properties, click the Defaults tab and deselect Enforce Strict Scoping Rules in JavaScript.
2515	Dynamic XML Form	Target version does not support XML data signatures.	Increase the target version to Acrobat and Adobe Reader 8.0 or later, or select the object, click the Object palette, click the Field tab or the Submit tab, and deselect Sign Submission.
2516	Dynamic XML Form	Target version does not support the Duplex Mode option.	Select File > Form Properties, click the PDF Print Options tab and deselect Use These Print Settings For Printing This PDF Form, or select File > Form Properties, click the PDF Print Options tab, and select Simplex.
2517	Dynamic XML Form	Target version does not support row shading in tables. If you change the target version after adding row shading, reapply it to correspond to the new target version.	Increase the target version to Acrobat and Adobe Reader 8.0 or later.
2518	Static PDF Form Dynamic XML Form	Target version does not support tooltips.	Increase the target version to Acrobat and Adobe Reader 8.0 or later, or select the static image, click the Accessibility palette, and remove the tooltip text.

Code	File type	Description	To remove the warning message
2519	Dynamic XML Form	Target version does not support the Number of Copies option.	Select File > Form Properties, click the PDF Print Options tab, select Default in the Number Of Copies list, or deselect Use These Print Settings For Printing This PDF Form.
2520	Dynamic XML Form	Target version does not support the No Page Scaling option.	Select File > Form Properties, click the PDF Print Options tab, select Use These Print Settings For Printing This PDF Form and then select Use Adobe Acrobat/Reader Setting or deselect Use These Print Settings For Printing This PDF Form.
2521	Dynamic XML Form	Target version does not support the Select Paper Source by Page Size option.	Select File > Form Properties, click the PDF Print Options tab, select Default in the Number Of Copies list, or deselect Use These Print Settings For Printing This PDF Form.
2600	Static PDF Form Dynamic XML Form	Target version does not support the Print on Both Sides or Print on Front Side Only options for the current file type.	In the Hierarchy palette, click Master Pages, click the Object palette, click the Page Set tab, and select Page Occurrence from the Printing list.
2601	Dynamic XML Form	Target version does not support the One-sided Printing Only and Two-sided Printing Only options.	Increase target version to Acrobat and Adobe Reader 8.1 or later.
2800	Static PDF Form (captions and text objects only) Dynamic XML Form	Target version does not support font scaling for the current file type.	For Static PDF Form (field values) and Dynamic XML Form, increase target version to Acrobat and Adobe Reader 9.0, or click the Font palette and type 100% in the Vertical Scale and Horizontal Scale boxes.
2801	Static PDF Form (captions and text objects only) Dynamic XML Form	Target version does not support letterspacing for the current file type.	For Static PDF Form and Dynamic XML Form, increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Font palette and type 0 in the Letter Spacing box.

Code	File type	Description	To remove the warning message
2802	Static PDF Form (captions and text objects only) Dynamic XML Form	Target version does not support kerning for the current file type.	For Static PDF Form (field values) and Dynamic XML Form, increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Font palette and deselect Kerning.
2803	Static PDF Form (captions and text objects only) Dynamic XML Form	Target version does not support text hyphenation for the current file type.	For Static PDF Form (field values) and Dynamic XML Form, increase target version to Acrobat and Adobe Reader 9.0 or later, or select Select File > Form Properties, click the Formatting tab, and click Remove All Hyphenation.
2804	Dynamic XML Form	Target version does not support GB18030 submit data encoding.	For Static PDF Form (button objects) and Dynamic XML Form, increase target version to Acrobat and Adobe Reader 9.0 or later, click the Object palette, click the Submit tab and select another Encoding option from the Data Encoding list.
2805	Static PDF Form Dynamic XML Form	Target version does not support the Widow and Orphan Control option for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or save the form as a Static PDF Form, or select Select File > Form Properties, click the Formatting tab, and deselect Widow and Orphan Control.
2806	Static PDF Form Dynamic XML Form	Target version does not support the Allow Page Breaks Within Content option for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Object palette, click the Subform tab, and deselect Allow Page Breaks Within Content.
2807	Static PDF Form Dynamic XML Form	Target version does not support the Keep With Next option for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or deselect Keep With Next.
2808	Static PDF Form (captions and text objects only) Dynamic XML Form	Target version does not support hyperlinks for the current file type.	For Static PDF Form (text objects and field values) and Dynamic XML Form, increase target version to Acrobat and Adobe Reader 9.0 or later, or remove hyperlinks.

Code	File type	Description	To remove the warning message
2809	Dynamic XML Form	Target version does not support the preSign event.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or remove the script event.
2810	Dynamic XML Form	Target version does not support the postSign event.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or remove the script event.
2811	Dynamic XML Form	Target version does not support the postOpen event.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or remove the script event.
2812	Dynamic XML Form	Target version does not support the postSubmit event.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or remove the script event.
2813	Dynamic XML Form	Target version does not support web service authentication.	Increase target version to Acrobat and Adobe Reader 9.0 or later.
2814	Static PDF Form Dynamic XML Form	Target version does not support this use of leaders for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later.
2815	Static PDF Form Dynamic XML Form	Target version requires the Allow Page Breaks Within Content option for the current file type.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or click the Object palette, click the Subform tab, and select Allow Page Breaks Within Content.
2816	Dynamic XML Form	Target version does not support the Prevent User from Changing No Page Scaling option.	Select File > Form Properties, click the PDF Print Options tab, deselect Prevent User from Changing, or select Use Adobe Acrobat/Reader Setting, or deselect Use These Print Settings For Printing This PDF Form.
2817	Static PDF Form Dynamic XML Form	Target version does not support the "%s" action.	Increase target version to Acrobat and Adobe Reader 9.0 or later.
2818	Static PDF Form Dynamic XML Form	Target version does not support the "%s" action.	Increase target version to Acrobat and Adobe Reader 9.0 or later, or delete the action, or unmanage the action's script, or undo the last action, or fix the action by picking another trigger condition.

Code	File type	Description	To remove the warning message
3000	Dynamic XML Form	Target version does not support Inactive presence.	Select the object, click the Object palette, click the Field tab, and select a supported presence.
3001	Dynamic XML Form	Target version does not support the Event Propagation option.	Select the object, click the Script Editor, and deselect Enable Event Propagation.
3002	Dynamic XML Form	The current target version does not support the US Postal Intelligent Mail Barcode.	Remove the object or increase the target version to Acrobat and Adobe Reader 9.1 or later.
3003	Dynamic XML Form	Target version does not support the Show Dialog Message.	Select File > Form Properties, click the Form Validation tab, select Show Dialog Message from the List of Options, and deselect Configure How Acrobat Displays Validation Message Boxes.
3004	Dynamic XML Form	Target version does not support the validationState event.	Increase target version to Acrobat and Adobe Reader 8.0 or later, or move the script to a supported event, or delete the script.
3005	Dynamic XML Form	Target version does not support the Set Focus option for form validation.	Select File > Form Properties, click the Form Validation tab, select Set Focus from the list of options, and deselect Set Focus To The First Field That Fails To Validate.
3006	Dynamic XML Form	Target version does not support the Color Failed Fields option for form validation.	Select File > Form Properties, click the Form Validation tab, select Color Failed Fields from the list of options, and deselect Color Fields That Fail Their Validations.
3007	Dynamic XML Form	Target version does not support the Color Mandatory Fields option for form validation.	Select File > Form Properties, click the Form Validation tab, select Color Mandatory Fields from the list of options, and deselect Color Mandatory Fields That Are Not Filled-in.
90000	Static PDF Form Dynamic XML Form	Target version does not support the %1 attribute or element.	A general message for problems that other target messages do not cover. It also appears when you edit the XML source and introduce unsupported functionality.

RELATED LINKS:

Report palette menu

Form Validation (Form Properties dialog box)

Action warning messages

The Warnings tab in the Report palette lists messages associated with actions.

The following table lists and describes the action warning messages.

Code	Type	Description
40000	Actions	The "action name" action is broken. It references one or more missing objects.

RELATED LINKS:

Building actions in forms

Form Validation (Form Properties dialog box)

Warning marker messages

The Warnings tab in the Report palette lists messages associated with warning markers that appear when a problem occurs with an object.

The following table lists and describes the warning marker messages.

Code	Type	Description
20000	Miscellaneous	At least two radio buttons have the same value. The group is meant to be mutually exclusive; all values must be unique.
20001	Miscellaneous	You may have to provide a return URL. See the Help documentation for more information about using the submit button.
20002	Miscellaneous	No email address specified. Use the Object palette to specify an email address.
20003	Miscellaneous	Invalid email address. Enter the email address of the form: name@address.com.
20004	Miscellaneous	Although this object allows rich text to be entered, its Data Format accepts only plain text. Use the Object palette to make these properties compatible.
20005	Miscellaneous	Although this object allows only plain text to be entered, its Data Format accepts rich text. Use the Object palette to make these properties compatible.
20006	Miscellaneous	The list box is set to support multiple selection and its Commit On property is set to 'select' rather than the recommended value of 'exit'.

Code	Type	Description
20007	Miscellaneous	The field's value area does not fit the maximum length value permitted. Increase the field's value area or decrease its value length to ensure that all possible values will fit.
20008	Miscellaneous	A border around a barcode field can affect its scanning.
20009	Miscellaneous	Background fill behind a barcode field can affect its scanning.
20010	Miscellaneous	Limit to visible area is ignored on fields that are set to expand to fit.
20011	Miscellaneous	For a screen reader to read a list properly, list items must be contained within a parent List element.
20012	Miscellaneous	No subform is present in the content area or the first subform is too large for the content area.
20013	Miscellaneous	A content area should be in the print-only view and in the screen-only view.
20014	Miscellaneous	A page area should be in the print-only view and in the screen-only view.
20015	Picture Clause	The specified Data pattern "%s" is invalid. Define a valid Data pattern.
20016	Picture Clause	The Data pattern "%s" is incompatible with the object's data format. Define a compatible Data pattern.
20017	Picture Clause	The specified Display pattern "%s" is invalid. Define a valid Display pattern.
20018	Picture Clause	The Display pattern "%s" is incompatible with the object's data format. Define a compatible Display pattern.
20019	Picture Clause	The specified Validation pattern "%s" is invalid. Define a valid Validation pattern.
20020	Picture Clause	The Validation pattern "%s" is incompatible with the object's data format. Define a compatible Validation pattern.
20021	Picture Clause	The specified Edit pattern "%s" is invalid. Define a valid Edit pattern.
20022	Picture Clause	The Edit pattern "%s" is incompatible with the object's data format. Define a compatible Edit pattern.

Code	Type	Description
20029	Data Binding	This field has the same name as another global field with an incompatible type.
20030	Data Binding	Using *Use Name*data binding when a default data connection is defined may produce undesirable results.
20031	Data Binding	Default binding value '%1' does not correspond to a data connection.
20032	Data Binding	Direct binding references to multiple levels of repeating data may not produce preferred results. Form may require relative binding references for repeating subform containers.
20033	Locale	The value may not be presented as expected when using Viewer's System Locale for this field representing currency.
20034	Locale	The spelling in this object, or in any child objects that inherit this locale, will not be checked.
20035	Miscellaneous	This fragment reference has local overrides to one or more properties of the source fragment.
20036	Miscellaneous	Cannot externally reference fragments in forms that are saved as PDF files. Save the form as an XDP file.
20037	Miscellaneous	The signature field cannot be repeatable.
20038	Picture Clause	The number of symbols in the specified Data pattern %s is not equal to the number of maximum allowable characters. Define a new Data pattern or change the maximum number of characters of the field.
20039	Miscellaneous	The number of symbols in the specified Display pattern %s is not equal to the number of maximum allowable characters. Define a new Display pattern or change the maximum number of characters of the field.
20040	Picture Clause	The number of symbols in the specified Validation pattern %s is not equal to the number of maximum allowable characters. Define a new Validation pattern or change the maximum number of characters of the field.
20041	Picture Clause	The number of symbols in the specified Edit pattern %s is not equal to the number of maximum allowable characters. Define a new Edit pattern or change the maximum number of characters of the field.
20042	Miscellaneous	The minimum number of letters after a hyphen must be less than the minimum number of letters of a hyphenated word.

Code	Type	Description
20043	Miscellaneous	The minimum number of letters before a hyphen must be less than the minimum number of letters of a hyphenated word.
20044	Miscellaneous	The content and caption cannot fit in the space provided. Press the indicator to automatically expand the object.
20045	Data Binding	Data connection ‘%s1' does not match the cached data description. Regenerate the data description using Connection Properties.
20046	Miscellaneous	Although the object is allowed to break, deselecting the Allow Page Break Within Content option of the parent object restricts this object from breaking between pages.
20047	Miscellaneous	Subform encoding is deprecated. Use the Collection encoding option or provide a custom script.
20048	Miscellaneous	Form contains two or more signature field objects. At least one signature field object has Lock Fields After Signing selected on the Signature tab. See the Help documentation for more information about using multiple signature field objects in a form.
20049	Data Binding	Export binding value '%1' does not correspond to a data connection.
20050	Data Binding	Import binding value '%1' does not correspond to a data connection.
20051	Data Binding	The maximum repeat value for this subform is set to a higher value than the data connection allows.
20052	Data Binding	The data node named ‘%1’ cannot be bound to this node type.
20053	Data Binding	The '%1' used to create the data connection has been modified. You may need to update the connection using Connection Properties.

RELATED LINKS:

Report palette menu

Form Validation (Form Properties dialog box)

Scripting error messages

The Warnings tab in the Report palette lists the following types of error messages associated with scripting errors in the form design:

JavaScript Scripting Errors

The following table lists and describes JavaScript scripting error messages.

Code	Error type	Error text	Error description
7001	Error	%1	A placeholder for any unexpected or unknown error. %1 can potentially contain any text.
7002	ReferenceError	Error %1 is undefined	An unqualified variable lookup failed.
7003	ReferenceError	Cannot assign value	An assignment attempts to access a constant such as a number, string, or XML.
7004	SyntaxError	Unterminated string constant	The closing quotation character (") was omitted at the end of a string constant. Example: `var sName = "Name;` To correct this error, locate the affected line and add the missing quotation character (").
7005	SyntaxError	Unterminated comment	The closing comment characters (*/) were omitted at the end of a comment string. Example: `/* The old fashioned comment style is still useful var i = 0;` To correct this error, locate the affected line and add the missing closing characters. In this example, add */ after the word useful. Note that when using the single-line comment characters (//), there is no need to terminate the comment string with matching closing characters.
7006	SyntaxError	Bad digit in number	Contains a character that is not a number or a valid separator (a period or a space). Examples: `123u8 123,8`
7007	SyntaxError	Language feature %1 is not supported	Currently, only property getter and setter methods are unsupported, as defined in Mozilla SpiderMonkey. Getter and setter methods are not part of the JavaScript standard. This error also occurs when compiling JavaScript without XML support and attempting to use XML.
7008	SyntaxError	Syntax error	A generic (catch-all) syntax error.

Code	Error type	Error text	Error description
7009	SyntaxError	Illegal use of reserved word '%1'	A keyword was used out of context. Example: `var for = 56;` The word `for` is a reserved word and cannot be used as a variable name. To correct this error, change the keyword to a non-reserved word.

7010	SyntaxError	Break or Continue outside a loop	The keywords `break` and `continue` are meant to be used inside a `for` loop or a `while` loop. The keyword `break` is also valid inside a `switch` statement. Using these keywords outside these structures is not permitted.
			Examples:
			Correct:
			`for (i = 0; i < 20; i++)`
			`{`
			`if (a == i)`
			`break;`
			`}`
			Incorrect:
			`var sName = "Nicole";`
			`break;`
			`var sAnimal = "cat”;`
			To correct this error, remove the line with the `break` statement.

7011	SyntaxError	Label not found	JavaScript does not support `goto`, but it supports labels for `continue` and `break` statements.
			Example:
			`outer: for (I = 0; I < 100; i++)`
			`{`
			`inner: while (condition)`
			`{`
			`if (bad)`
			`break outer;`
			`}`
			`}`
			To correct this error, ensure that any label used in a program is defined and that spelling is consistent where labels are referred to.

Code	Error type	Error text	Error description

7013	SyntaxError	Too many closing braces	The program contains an unmatched closing brace.
			Example:
			`if {sSoftware == "Designer"}`
			`{`
			`// Heh.`
			`sDesc = "Form Design Software";`
			`}}`
			To correct this error, remove the extra closing brace.

7014	SyntaxError	No matching closing brace found	A closing brace is missing somewhere in the program.
			Example:
			The closing brace of the `for` loop is missing in this program:
			`if {sSoftware == "Designer"}`
			`{`
			`for {i = 0; i < 7; i++}`
			`{`
			`nCount = nVer + 1;`
			`}`
			Despite the indentation that indicates the `for` loop is missing a closing brace, the error line usually indicates that the `if` statement is missing a brace. Generally, with a missing brace error, the outermost statement is reported to be missing the closing brace.

7015	SyntaxError	Try without catch/finally	The `try` statement cannot be used unless it is paired with the `catch/finally` statement. To correct this error, remove the `try` statement or add a `catch/finally` statement.
7016	SyntaxError	Catch/finally without try	The `catch`/`finally` statement cannot be used unless it is paired with the `try` statement. To correct this error, remove the `catch/finally` statement or add a `try` statement.

7017	TypeError	Variable expected	A variable name is expected.
			Examples:
			`function f{1} {} try {} catch {"hi"} {}`
			`var 5;`
			The characters in bold are where a variable name is expected rather than a constant or a number.

Code	Error type	Error text	Error description

7018	TypeError	Variable or value expected, but found %1	Rarely displayed.
			The conversion of an object to a primitive value failed.
			Example:
			`o={toString:function(){return this}};`
			`o+"test";`

7019	TypeError	Bad argument %1	A function argument cannot be converted to a suitable data type. For example, a function is expecting a number, but a string is passed in instead.
7020	TypeError	Bad argument list	There is a problem with the function's argument list, and the arguments cannot be used.
7021	TypeError	%1 is not an object	An invalid object is being used to perform an operation. An object is needed for an operation, but the data cannot be converted to an object. Example: `var obj = null; obj.toString{};` Calling a method on `obj` fails because `obj` is null and not an object.
7022	ReferenceError	%1 does not have a constructor	Host objects that cannot be created, such as the Application object, have a dummy constructor function so that the prototype object can be accessed. An example is Application.prototype where an attempt was made to use this function as a constructor.
7023	ReferenceError	%1 does not have a value	The conversion of an object to a primitive value failed. Example: `o={toString:function{}{return this}};` o+"test";

7024	ReferenceError	%1 is not a function	Something is invoked as a function, and it does not exist.
			Example:
			`var f = "No function";`
			`f{};`

7025	SyntaxError	Expected: %1	Parser expected a certain symbol but did not locate it. The missing symbol is often a single character but can be more than one character. Example: `<xml>{javascript]</xml> // expected: }` The parser was expecting the closing symbol `}` after the word `javascript` to match the opening symbol.

Code	Error type	Error text	Error description

7026	Error	%1 cannot work with this class	Methods were moved from one class to another. Only the `String` and `Array` methods are generic enough to work with different classes.
			Example:
			`s = new String {'test'};`
			`s.getTime = Date.prototype.getTime;`
			`s.getTime{};`

7030	SyntaxError	Illegal 'return' outside a function body	Using the keyword `return` outside a function definition is not permitted. To correct this error, remove the `return`statement.
7037	SyntaxError	Conversion error	A conversion from one character encoding to another fails. This is not a JavaScript error. It occurs, for example, when trying to read a malformed Shift-JIS file.
7038	SyntaxError	Partial multibyte	This error occurs during the conversion from one character encoding to another. This is not a JavaScript error. It occurs, for example, when the last byte of a UTF-8 file is missing when the file is read.

7039	SyntaxError	More than one switch default	The `switch` statement has a special label, `default`, which is the code that runs if no other case in the switch is chosen. Only one `default` label per `switch` statement is allowed.
			Example:
			`switch {nVersion}`
			`{`
			`case 1:`
			`// ...`
			`break;`
			`case 2:`
			`// ...`
			`break;`
			`default:`
			`// ...`
			`break;`
			`default:`
			`// ...`
			`break;`
			`};`
			To correct this error, delete all but one of the `default` labels.

Code	Error type	Error text	Error description

7040	TypeError	%1 redeclared	A constant cannot be declared more than once. A variable can be declared more than once.
			Example:
			`const a = 5;`
			`const a = 6;`

7041	RangeError	%1 is out of range	An argument, index, or value exceeds the allowable numeric range. Example: `Number {5}.toFixed {111}; // 100 is max`

7042	SyntaxError	Catch after unconditional catch	JavaScript supports multiple catch clauses, but the last catch clause must be unconditional.
			Example:
			`try {}`
			`catch {e if e instanceof String} {}`
			`catch {e if e > 5} {}`
			`catch {e} {}`
			`catch {e if typeof e == "object"} {}`
			To correct this error, move the illegal conditional catch clause before the unconditional catch clause, or delete it.

FormCalc Scripting Errors

The following table lists and describes FormCalc scripting error messages.

Code	Error Type	Error Text	Error Description

7008	SyntaxError	Syntax error near token ‘%1’ on line %2, column %3.	A generic {catch-all} FormCalc error. Generally, %1 contains the token (word) nearest to the error. The token may not be associated with the error, other than proximity to the problem.
			Example:
			`var b = abc{1}`
			`if {b ne 1} then`
			`//comment`
			The error in this example is that the `endif` token is missing from the script. The last correct token is `then`. Comments do not count as tokens.
			To correct this error, add an `endif`statement to the end of the script.

Code	Error Type	Error Text	Error Description

7100	SyntaxError	Function '%1'on line %2, column %3 is built in.	A user-defined function used the same name as a built-in function.
			Line and column numbers provide information to locate where the error appears on a line.
			Example:
			`func sum{}`
			`do`
			`x = 1`
			`endfunc`

7101	SyntaxError	Function '%1'on line %2, column %3 is unknown.	A script attempted to invoke a function that is not defined. Example: `read{}`

RELATED LINKS:

Report palette menu

Form Validation (Form Properties dialog box)

4.3. Displaying validation errors in Acrobat

Use the form validation options to control how Acrobat displays data validation errors in a PDF form.

The form validation options apply to the entire PDF form and work with other validation features like validation patterns and scripts. Before you select form validation options, apply a validation pattern or script to each field object that you want Acrobat to verify. For example, you can configure a numeric field object with a validation pattern that validates user-entered values to ensure the proper format of the number. If a user enters an invalid number, the field fails to validate. The validation pattern options define how data must be entered into a field for it to validate. The form validation options control the appearance and behavior of validation errors in Acrobat.

Designer generates a validation script for each of the validation options, except for Show Dialog Message option. The Show Dialog Message option does not generate script.

Designer monitors the validation script for any changes. If Designer detects the script is modified, it performs the following actions:

- Stops monitoring the script and all associated scripts for that action.
- Deselects the associated validation option on the Validation tab in the Form Properties dialog box.
- Adds a message to the log file indicating that the script is no longer monitored and can be edited.

As long as the validation script is unchanged and managed, Designer removes the script from the form if you deselect a validation option.

Designer generates different form validation script for static and dynamic forms. As a result, when you save a form as an Adobe XML Form (*.xdp), Designer determines which type of script to generate based

on the option you select in the PDF Render Format list (Static PDF Form or Dynamic XML Form) on the Defaults tab in the Form Properties dialog box.

NOTE: The script that the form validation options generate overwrites any existing script that changes the appearance of field objects. The overwrite occurs when the field objects become valid or invalid. It is recommended that you customize existing scripts to store the current appearance, so that you can revert to that appearance if necessary.

It is recommended that you use the form validation options with dynamic PDF forms. However, if you are working with a static PDF form, review the recommendations you must keep in mind.

Recommendations for using form validation with static PDF forms

If you use the Form Validation feature in a static PDF form, to color mandatory fields that are not filled or fields that fail validation, Designer may change the appearance of validated fields.

As a result, when you use the Form Validation feature with a static form that contains objects with borders, it is recommended that you use only these border options on the Border tab:

- Border Edges: Solid, Lowered-3D, Raised-3D, Dash
- Corners: Rectangle corner
- Background Fill Style: Solid

To configure how Acrobat displays validation error message boxes

You can configure how Acrobat displays validation error message boxes when a user fills or submits a PDF form. For example, you can choose to show each validation error message in its own box or combine all validation error messages into a single box. Alternatively, you can show only the first validation error message or no messages.

NOTE: Form validation options do not apply to field objects that are not configured to validate.)

1) Select File > Form Properties and click Form Validation.
2) Under List of Options, click Show Dialog Message.
3) Select Configure How Acrobat Displays Validation Messages Boxes and perform one of these actions:
   - To display each new validation error message after the user closes the current message, select Show Every Message In Its Own Message Box One After The Other.
   - To combine validation errors into a single list, select Combine The Messages Of All The Failed Fields Into One Message Box.
   - To display only the first validation error message, select Show the First Failed Field’s Message And Suppress Any Other Messages.
   - To stop all validation messages, select Don’t Show Any Messages Boxes At All.
4) Click OK.

To color fields that fail validation

You can choose different colors to highlight the borders, the background fill, or both of field objects that fail to validate in Acrobat.

NOTE: Form validation options do not apply to field objects that are not configured to validate.)

1) Select File > Form Properties and click Form Validation.
2) Under List of Options, click Color Failed Fields.
3) Select Color Fields That Fail Their Validations and perform one or both of these actions:
   – To select a color to apply to the borders of field objects, click Border Color and select a color from the palette.
   – To select a color to apply to the background of fields objects, click Background Color and select a color from the palette.
4) Click OK.

To color mandatory fields that are not filled

You can choose different colors to highlight the borders, the background fill, or both of mandatory field objects that the user did not fill in Acrobat.

If a mandatory field is highlighted and another script makes the field optional, highlighting is not removed.

NOTE: Form validation options do not apply to field objects that are not configured to validate.

1) Select File > Form Properties and click Form Validation.
2) Under List of Options, click Color Mandatory Fields.
3) Select Color Mandatory Fields That Are Not Filled-In and perform one or both of these actions:
   – To select a color to apply to the borders of field objects, click Border Color and select a color from the palette.
   – To select a color to apply to the background of fields objects, click Background Color and select a color from the palette.
4) Click OK.

To set the focus to the first field that fails validation

The Set Focus To The First Field That Fails To Validate option sets the focus to the first non-validated field in Acrobat.

The focus is set to the first invalid field in the Hierarchy palette, not the first invalid field in the geographic order. It is recommended that you order the field objects in the Hierarchy palette to match the order of the field objects in the form design. Otherwise, the focus could be set on an invalid field at the bottom of the page before one at the top of the page.

NOTE: Form validation options do not apply to field objects that are not configured to validate.

1) Select File > Form Properties and click Form Validation.
2) Under List of Options, click Set Focus.
3) Select Set Focus To The First Field That Fails To Validate and then click OK.

RELATED LINKS:

Formatting field values and using patterns

Addressing warning messages in the Report palette

Objects that support scripting and calculations

Defaults (Form Properties dialog box)

Document Handling (Options dialog box)

To validate user input

4.4. Guidelines for forms

When creating forms in Designer, you can save your form designs in several different formats. Form author can save a form design as a PDF form so that form fillers can easily access the form using Adobe Reader®. When saving a form design as a PDF form in Designer, you can select between two different types of PDF forms: Adobe Static PDF Form (*.pdf) and Adobe Dynamic XML Form (*.pdf). Form designs saved as static and dynamic PDF forms can be interactive or non-interactive.

RELATED LINKS:

Creating interactive forms that have a flowable layout

Creating non-interactive forms that have a flowable layout

Rendering PDF forms

To understand the difference between static and dynamic PDF forms, it is important to understand what the term render means. Rendering a form is the process of creating the precise final layout and formatting from the form design. The form may or may not be merged with data, depending on the form design and data. Rendering can be done by AEM forms on the server or by Adobe Reader on the client. A PDF form must be rendered before it can be displayed to the end user or printed. Therefore, the rendering is the final image.

Characteristics of static PDF forms

Form designs saved as static PDF forms render once on the server and are displayed on the client in the Acrobat or Adobe Reader target version. They are not rerendered in response to user interaction. The form may have been designed with a flowable layout; however, when the static PDF form is created, its layout is fixed and the resultant PDF form will not rerender on the client.

File size

In general, forms saved as static PDF forms render to larger file sizes than the equivalent files saved as dynamic PDF forms.

Rendering location

Static PDF forms render once and are displayed on the client in Acrobat or Adobe Reader. They are not rerendered in response to user interaction. Because rendering is performed on the server, only small changes to the final output are possible on the client. For example, the value area may show a different color after the user clicks a button or exits a field.

Display speed

In general, static PDF forms require more time to render on the server than dynamic PDF forms but are displayed faster in the client software on the end-user's computer. Static PDF forms require more time on the server because the server performs the page layout operations, data merging, and final rendering. In the case of dynamic PDF forms, the server performs only data merging. The client software on the end-user's computer performs the page layout operations and final rendering. As a result, the rendering time for dynamic PDF forms relies on the processing power of the end-user's computer, especially when the PDF file size is large.

Characteristics of dynamic PDF forms

Forms saved as dynamic PDF forms render on the client in Acrobat or Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of objects is possible in Acrobat or Adobe Reader because Acrobat or Adobe Reader have enough information to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If the end user clicks a button that adds a new row to a table, the form is rerendered in Acrobat or Adobe Reader. Dynamic PDF forms were first introduced in Adobe Reader, Acrobat Professional and Acrobat Standard 7.0.

File size

In general, forms saved as dynamic forms render to smaller file sizes than the equivalent files saved as static PDF forms.

Rendering location

Form designs saved as dynamic PDF forms render on the client in Acrobat or Adobe Reader. Because rendering is performed on the client, the form can rerender several times based on end-user interactions.

Display speed

In the case of dynamic PDF forms, the server performs only data merging. The client software on the end-user's computer performs the page layout operations and final rendering. As a result, the rendering time for dynamic PDF forms relies on the processing power of the end-user's computer, especially when the PDF file size is large.

RELATED LINKS:

Choosing the type of PDF form

Choosing the type of PDF form

You can create many kinds of forms as either static or dynamic with little difference to the end-user experience. However, some forms work as designed only if they are created as a static or dynamic PDF forms. In general, the choice of static over dynamic is determined by the following considerations:

- If the form works as either a static PDF form or a dynamic PDF form, use a dynamic PDF form to reduce server-side processing, which results in a greater number of transactions per second.
- If the form relies on client-side scripts to change the layout (for example, it uses scripts to add or remove rows from a table or to make text fields grow), use a dynamic PDF form.
- If end users will need to add annotations or comments to the PDF form, use a static PDF form.
- If the form must work with Acrobat installations earlier than version 7.0, use a static PDF form.

Additionally, there are a number of specific issues to consider when choosing to create a static or dynamic PDF form. These issues are differences in the behavior between the form types that may be critical in making your decision:

- *Applying formatting by using client-side scripts*
- *Saving object formatting*
- *Unavailable commandsfor dynamic forms in Adobe Reader*
- *Font behavior*
- *Objects and properties for static PDF forms*
- If your form includes digital signatures, there are limitations to the way that you can use dynamic PDF forms with the Signature service. For more information about these limitations, see AEM forms Services Reference.
-

As you design a form, you can see how the form behaves as a static or dynamic PDF form in the Preview PDF tab.

Applying formatting by using client-side scripts

In a static PDF form, only the value area of the field can be updated on the client. Everything else on the form is frozen or fixed. In a dynamic PDF form, the entire field object can be updated on the client. The examples that follow are provided to give you an idea of what happens.

Changing the field border color

If you create a client-side script to apply borders to objects, the results differ in static and dynamic PDF forms.

For example, you can write a script on the exit event of a field to turn the border color red:

```
TextField1.border.edge.color.value="255,0,0"
```

In a static PDF form, the outline of only the value area turns red after the end user exits the field.

In a dynamic PDF form, the outline of the entire field object turns red after the end user exits the field.

Applying field shading

If you create a client-side script to apply shading to objects, the results differ in static and dynamic PDF forms.

For example, you can write a script on the exit event of a field to turn the shading color in the field to red:

```
TextField1.fillColor = "255,0,0"
```

In a static PDF form, only the value area turns red after the end user exits the field.

In a dynamic PDF form, the entire object, including the value area, turns red after the end user exits the field.

Hiding objects

If you create a client-side script to hide objects, the results differ in static and dynamic PDF forms.

When you are designing a static or dynamic PDF form, you can hide objects on the form by setting the presence property to either “invisible” or ”hidden”. In both cases, the objects do not appear in the final output; however, when you set the presence value to “hidden”, objects do not occupy any space in the layout. If the objects are inside flowed subform objects, the subform itself will shrink in response to the hidden objects.

The difference between static and dynamic PDF forms is that on a static PDF form you cannot change the presence value of an object by using an interactive scripting event, such as the click event of a button. Static PDF forms cannot rerender on the client; therefore, scripts executed on interactive events cannot change the visibility of form objects. Dynamic PDF forms do not have the same limitation because they can rerender on the client.

You can change the visibility of form objects on static PDF forms by using non-interactive events that trigger during form rendering, such as the initialize event. For example, on a dynamic PDF form, you can write a script on the click event of a button to hide a text field:

```
TextField1.presence = "invisible"
```

Alternatively, you can use this script to completely remove the text field from the layout:

```
TextField1.presence = "hidden"
```

In both cases, to achieve the same results on a static PDF form, you use the same scripts but write them on the initialize event of either the text field or the button.

RELATED LINKS:

Saving object formatting

Saving object formatting

Prior to Acrobat 8.0, in a static PDF form, the object's formatting is saved when the end user saves, closes, and reopens the form. In a dynamic PDF form, the object's formatting is not saved.

IMPORTANT: *Although it is possible to save an object's formatting on a dynamic PDF form by using client-side scripting, it is not a recommended practice and should be avoided. If saving an object's formatting is a requirement, you need to use a static PDF form.*

Beginning with Acrobat 8.0, the state of all the objects is saved and can be restored automatically or manually upon reopening. This is controlled by the `restoreState` scripting property on the root subform.

Object formatting, as well as the layout of a dynamic PDF form, can change when any of the following form actions occur:

- The user opens the form
- The user imports new data
- The user adds new data to the form
- A client-side script is run

The following table compares the end-user experience using the examples in the topic *Applying formatting by using client-side scripts* when saving as either a static PDF form or a dynamic PDF form in Adobe Reader.

Form content	Static PDF form	Dynamic PDF form
A client-side script to change the border color of a field. (See Changing the field border color.)	The outline of the value area remains red when the form is saved, closed, and reopened.	The outline of the object does not remain red when the form is saved, closed, and reopened.
A client-side script to apply shading. (See Applying field shading.)	The fillable value area remains red when the form is saved, closed, and reopened.	The object, including the value area, does not remain red when the form is saved, closed, and reopened.
A client-side script to hide objects. (See Hiding objects.)	The value in the text field remains invisible when the form is saved, closed, and reopened.	The text field does not remain hidden when the form is saved, closed, and reopened.

NOTE: If `restoreState` is set to `auto`, all of these changes are maintained if the document is saved and then reopened in Acrobat 8.0.

For more information, see *restore State* in the *Scripting Reference*.

Unavailable commands for dynamic forms in Adobe Reader

When a PDF form is opened in Adobe Reader, some Adobe Reader commands that are available for static PDF forms are not available for dynamic PDF forms. The reason is because the final layout of the form can change the position of objects, which in turn affects the number of form pages.

The following Adobe Reader commands are available for static PDF forms but not for dynamic PDF forms:

- Comment and Markup tools
- Import Comments and Export Comments
- Insert Pages, Extract Pages, Replace Pages, Delete Pages, Crop Pages, and Rotate Pages

RELATED LINKS:

Choosing the type of PDF form

Target version warning messages

Font behavior

When you create PDF forms, it is possible to include only those characters of a font that are actually used in the form. This technique is called font subsetting. You can also embed fonts into the form so that end users have all of the fonts they need to use the form. In this case, the size of the PDF file is larger.

NOTE: The manufacturer of the font can specify the level of embedding that is allowed.

Fonts can be subset in static PDF forms

In static PDF forms, fonts can be subset for text and field captions. Subsetting reduces the size of the PDF file. To subset fonts into the form, the font you select must be set to the Print & Preview Embedding Allowed level.

NOTE: The fonts used in the fillable area of fields cannot be subset on static PDF forms because the user input can contain any character from the font.

Fonts are embedded in dynamic PDF forms

In dynamic PDF forms, fonts are fully embedded or linked. Embedded fonts are never subset. When a form that has embedded fonts is opened, even if the fonts are already on the system, Adobe Reader uses the embedded fonts. To embed fonts into the form, the embedding settings in the fonts you select must allow embedding for edit.

RELATED LINKS:

Choosing fonts for performance

Objects and properties for static PDF forms

Here is the list of scripting objects that form authors can use to make changes to fields in static PDF forms in Adobe Reader. For more information about scripting objects, see Scripting Objects.

Object	Description
border	Describes the border surrounding an object
color	Describes a unique color on an object
fill	Applies a color and optional rendered designs to the region enclosed by an object

Here is the list of scripting properties that form authors can change for fields in static PDF forms in Adobe Reader.For more information about scripting properties, see Scripting Properties.

Property	Description	Syntax
access	Controls user access to the contents of a container	*SOMexpression*.access = "open | readOnly|protected"
fontColor	The font color value for the text	*SOMexpression*.fontColor = "[0-255], [0-255], [0-255]"
hAlign	Specifies the horizontal text alignment	*SOMexpression*.hAlign = "left | center | right"
maxChars	Specifies the maximum number of characters that this text value can enclose	*SOMexpression*.maxChars = "0 | integer"
maxLength	Specifies the maximum (inclusive) permitted length of the content or -1 to indicate that no maximum length is imposed	*SOMexpression*.maxLength = "0 | integer"
multiLine	Specifies whether the text may span multiple lines	*SOMexpression*.multiLine = "0 | 1"
presence	Specifies the visibility of an object's value	*SOMexpression*.presence = "visible | invisible|hidden"
textEntry	Determines whether a user can type a value into a drop-down list	*SOMexpression*.textEntry = "0 | 1"

4.5. Import documents

You can import forms into Designer that were created in another form authoring application or an XForm XML application, or import spreadsheet data from Microsoft Excel 2002 or later by copying and pasting the content of the spreadsheet cells into a form. You can also import legacy forms as a starting point for creating new Designer forms. You can import files from these applications into Designer.

Application	File types
Acrobat	PDF
Adobe Form Designer 5.0 or later	XFT
Adobe Output Designer	IFD To import Output Designer files (IFD) into Designer, you must install Output Designer 5.5 or later on the same computer as Designer.
Microsoft Word	DOC, DOT, RTF
Microsoft Excel	XLS, XML, HTML, XLT, TXT
Microsoft InfoPath	XSN
XForms Model	XHTML, XML, HTML, HTM, XFDL

Designer includes a number of options to handle imported files and render the best possible output. Designer preserves the layout of the imported form as much as possible and converts the elements into Designer objects.

Importing PDF files

PDF files can come from many sources, and each authoring application can define the contents of the PDF file in a unique manner.

Before importing a PDF file into Designer, you should understand the different options available for fine-tuning the results of the import process.

NOTE: PDF files created by using Designer do not go through the import process. They open directly in Designer and should appear as designed without modifications.

You use the New Form Assistant to import PDF files. Using the New Form Assistant simplifies the process. It guides you through a number of steps in which you choose the PDF file to import, how you want to work with the imported PDF content, and how the form is distributed and returned.

When Designer is integrated with Workbench, you can also use Workbench to import PDF files. In Workbench, select File > New > Form and then follow the onscreen instructions. On the Getting Started panel of the New Form Assistant, select the Import A PDF Document option.

If you are using the stand-alone version of Designer, you can also use the Import Options dialog box to import a PDF file. The Import Options dialog box is not available when you use Designer with Workbench. Using the Import Options dialog box gives you more control over the way Designer imports the contents of the PDF file when creating an interactive form with a flowable layout. For example, you can select custom options that control how Designer processes the text, paragraphs, and images in the document. Use one of these methods to display the Import Options dialog box:

- Click the Do Not Use the Assistant link in the New Form Assistant to open the Assistant Options dialog box. Select the option that best suits your needs.

- Select Tools > Options to open the Options dialog box, select Wizards and Tips and, under Form Assistance, deselect Show When Importing Documents. (See *Wizards and Tips (Options dialog box)*.).

Before you import a PDF, consider the following scenarios:

- If you have a PDF document that was created in Acrobat or an application other than Designer, and you want to import the contents as background artwork and maintain the original layout and appearance of the document, select Create an Interactive Form with Fixed Pages. You can edit any interactive objects and place additional library objects on top of the artwork.
- If you want to edit and change the content of the PDF document in Designer, select Create an Interactive Form with a Flowable Layout. Word-wrapping and line breaks may not be preserved.

If you are not satisfied with the initial conversion, you can experiment with different combinations of options to achieve the best results.

NOTE: When importing a PDF file that has security permissions set, Designer prompts you to enter the correct password to prevent unauthorized access.

To import a PDF file by using the New Form Assistant in the stand-alone version of Designer

1) Select File > New.
2) In the Getting Started panel of the New Form Assistant, select Import a PDF Document and click Next.
3) In the Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
4) In the Document Setup: Import Options panel, select the import option you want to use:
   – To import the PDF file as background artwork, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
   – To import the PDF file so that you can edit it in Designer, select Create an Interactive Form With a Flowable Layout.
5) Click Next.
6) [Optional] In the Form Return Setup: Adding Buttons panel, select how the form is distributed and how the form data is returned.
7) Click Finish.

To import a PDF file without the New Form Assistant in the stand-alone version of Designer

1) Select File > Open.
2) Navigate to the PDF file that you want to import, select the file, and click Open. The Import Options dialog box appears. However, if you have not disabled the New Form Assistant by deselecting the Show When Importing Documents option in the Options dialog box (Wizards and Tips panel), the New Form Assistant appears. If necessary, click Do Not Use Assistant to open the Import Options dialog box.
3) In the Import Options dialog box, select the option that best suits how you want to work with the imported PDF content:

- To preserve the layout and appearance of the PDF content, select Create an Interactive Form with Fixed Pages.
- To edit the content of the PDF document in Designer, select Create an Interactive Form with a Flowable Layout, and then select custom options for importing the PDF, as required.

4) Click OK. If you selected the Display A Summary Report option, a dialog box lists any conversion issues, such as unavailable fonts and unsupported objects, before the form opens.

   NOTE: The Temporary folder is your temporary directory that is configured by Windows (which is typically \Documents and Settings*[username]**Local Settings**Temp). The log file is named the same as the PDF file with a .log file name extension. If a file with that name already exists in the Temporary Folder, it is overwritten.*

To use File > New to import a PDF file with Designer and Workbench

When you use File > New to import a PDF file, you can specify a name for the form. The form will be saved in the Applications view in Workbench.

1) Select File > New. The New Form dialog box opens in Workbench.
2) Follow the onscreen instructions until the New Form Assistant > Getting Started panel opens in Designer.
3) Select the Import A PDF Document option, and then click Next.
4) In the Document Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
5) In the Document Setup: Import Options panel, select the import option you want to use:
   - To import the PDF file as background artwork, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
   - To import the PDF file so that you can edit it in Designer, select Create an Interactive Form With a Flowable Layout.

     NOTE: *By default, images are embedded in medium quality JPEG format. If an image extraction problem occurs, a message appears in the Conversion Summary and the image is embed in BMP format.*

6) Click Next.
7) Click Finish.

To use File > Open to import a PDF file with Designer and Workbench

When you use File > Open to import a PDF file, you specify the AEM forms application where you want to save the form.

1) Select File > Open.
2) Navigate to the PDF file to import, select the file, and click Open.
3) In the Document Setup: Import Options panel, select the import option you want to use:
   - To import the PDF file as background artwork, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.

- To import the PDF file so that you can edit it in Designer, select Create an Interactive Form With a Flowable Layout.

  NOTE: *By default, images are embedded in medium quality JPEG format. If an image extraction problem occurs, a message appears in the Conversion Summary and the image is embed in BMP format.*

4) Click Finish.
5) Select File > Save As.
6) Navigate to the Workbench folder on your local system:
   - If you are using Windows XP, the Workbench folder is located in \Documents and Settings\<user name>.
   - If you are using Windows Vista, the Workbench folder is located in \Desktop\<user name>.
7) Select the application folder where you want to save the form.
8) Type a name for the form, and then click Save.

RELATED LINKS:

Saving forms

Import Options dialog box

Wizards and Tips (Options dialog box)

Reviewing the results of a PDF file import

The form design that results from an imported PDF file depends on the different combinations of options that you selected during the import process. If you are not satisfied with the initial conversion, you might want to experiment with different combinations of options to achieve the best results, or you can edit the objects in the form design by using the Object Editor.

The Object Editor appears when you select an object, and it provides quick access to commands that are commonly used when editing objects.

Initially, object boundaries appear in the form design so that you can see how the PDF file elements, especially text, have been grouped into objects. This can make the form design look busy and difficult to read. Using the Object Boundaries command in the View menu, you can hide the object boundaries while you examine the form design. You will find the object boundaries useful if you need to make changes to objects in the form design. For example, you use the object boundaries as a guide when you resize or move objects without borders. Also, if a body of text is divided over two or more objects in the form design, the object boundaries will show it. You can easily merge these objects into one text object by using the Merge Selected Text Objects command in the Layout menu.

Title and instruction text may be divided into multiple text objects and field objects. You can create a caption for a field object that does not have a caption by merging it with a text object using the Merge as Caption command in the Layout menu.

You might also find the locking commands in the Edit menu useful. Locking objects prevents you from selecting and editing them in the Layout Editor. For example, you can lock text so that you must

double-click the required text or caption to edit it. This makes it somewhat difficult to modify the text. You can also lock static objects such as text, lines, images, and shapes, and you can lock field objects.

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

About reducing PDF conversion problems

Adding a new mapping to the font-mapping table in Designer

Character-mapping table in ConvertPDF_CharMap.txt

How Designer converts PDF objects

The conversion of PDF contents into XDP format is highly dependent on the source of the PDF file itself. The PDF specification allows for a wide variety of styling and structural layout, some of which lends itself to good migration. Often, PDF is used as a final form layout; that is, the display of the contents is the only thing that matters, not the contents themselves.

In many situations, Distiller® or PDFMaker actually generates the PDF file, but they both can only render what is provided to them in the order that it is provided.

This table identifies key elements in a PDF file and indicates how Designer treats them when they are imported.

Item	Converted	Not converted
Actions (from an Acrobat form)	Designer only converts the following actions that may be associated with a form created in Acrobat: • Import form data *Launch a URL* *Reset a form (full and partial field list)* *Show/hide a field* *Submit a form for complete field list*	All remaining actions
Calculations	Average Maximum Minimum Sum Range	
Comments/ Annotations		Not converted

Item	Converted	Not converted
Document	Document-level JavaScript script is converted to event scripts that are commented out. You must verify and update the scripts to match the Designer model. Pages and names Events other than JavaScript scripts Security options Document metadata, including PDF version, title, author, subject, keywords, creator, producer, creation date, modification date Accessibility information of Tagged PDFs is preserved.	Tab order
Fields	Date, Text, Numeric, Check Box, Radio Buttons and groups, Button, Combo box, Signature (only identified), Drop-down List, and List Box field types Default text Dimensions Events Font name, size, bold, italic JavaScript script is converted to event scripts that are commented out. You must verify and update the scripts to match the Designer model. Name Picture formatting Position, orientation, alignment, and visibility Printable Read Only, Required, Maximum Characters, and Multiline attributes RGB and grey color definitions for background color, border color, and color. Other color specifications are mapped to RGB values. Tool tip is preserved	All fonts used in a PDF file must exist on your system or font substitution rules must be defined in the font-mapping table. Designer uses a cross (x) to indicate when a check box is selected. If a check box in the PDF file uses a different character, the character is converted to a cross that may not be perfectly aligned inside the fillable area.
Images	All 1-bit, 4-bit, 8-bit, and 24-bit Dimensions Position	Vector based 6-bit 32-bit Some TIFF formats, including tiled TIFF format
Lines	Color (both edge and fill) Dimensions Position Width Some dotted and dashed styles Style	

Item	Converted	Not converted
Rectangles	Color (both edge and fill) Dimensions Position Width Some dotted and dashed line styles Rounded-corner rectangles are converted when they are detected. The conversion depends on how the PDF file was generated.	Style
Scripts	Document, page, and field-level JavaScripts scripts are converted to event scripts that are commented out. You must verify and update the scripts to match the Designer model.	
Text	Content Dimensions Font name, size, bold, italic Mixed formatting (rich text) Position RGB and grey color definitions for background color, border color, and color. Other color specifications are mapped to RGB values.	All fonts used in a PDF file must exist on your system or font substitution rules must be defined in the font-mapping table. Symbol-based characters are not always displayed correctly. You can define character-mapping rules in the character-mapping table. Designer does not support font kerning. When PDF-based kerning is detected, the font size is decreased so that the text occupies the same horizontal space.

About reducing PDF conversion problems

Some PDF files may be difficult to import into Designer. Conversion problems can depend both on the application used to produce the PDF file and on choices made when the source document was authored. For example, these original sources of PDF files have proven difficult to import into Designer:

- Digipath files, which are converted to one large image that cannot be edited
- Microsoft Word, when the document has not been specifically designed as a form

You can reduce possible conversion problems by being aware of these issues:

- Fonts are substituted if a font used on the form is not installed when you import the form. If a font is substituted, text will not appear exactly as in the original form.

- If you are using Acrobat to create PDF files, do not build the form by using drawing annotations only.
- When an image in a PDF form contains a large number of lines, such as a vector-based image, Designer needs a significant amount of time to process the lines during an import operation. For faster processing, remove the image from the source file that is used to create the PDF. If required, you can reinsert the image after you save the form design in Designer. If the source file is not available, deleting the image from the form in Designer will improve performance of the form.
- If you are having conversion problems and you have access to the source files for the original form, try re-creating the PDF by using Distiller. If the original PDF was not created correctly, re-creating it may resolve the issue.

Importing tagged PDF files

When importing a tagged PDF (an accessible PDF), the accessibility information of the page contents is preserved. In addition, organizing the form by adding or changing subforms may cause the accessibility information to change.

To ensure that the form still performs in a manner acceptable to your users, test the form thoroughly. Pay close attention to the following areas and adjust them if they are not behaving as expected:

- Tool tips
- Custom screen reader text
- Screen reader reading order

RELATED LINKS:

About accessible forms

Importing PDF files

Matching unavailable fonts

During import, Designer attempts to match fonts in the PDF file with fonts that are available on your computer. If the file contains an unavailable font, Designer displays the Missing Fonts dialog box showing the missing font and a suggested replacement. You can accept the replacement font or change it. This font substitution is not permanent. Designer must map the unavailable fonts every time the form opens.

After importing the PDF, the Warnings tab of the Report palette lists all unavailable fonts and explains how they were converted. If the font-mapping table includes the unavailable font, the Warnings tab states that the font was mapped to the new font. If the unavailable font is not in the font-mapping table, Designer attempts to select an appropriate substitution. The Warnings tab then states that the unavailable font was changed to the new font. In this case, you may want to permanently substitute the font or add a new mapping to the font table.

To make the font substitution permanent

1) Import the PDF. See *Importing PDF files*.
2) If the Missing Fonts dialog box appears, specify the desired substitute fonts.

3) To apply permanent font substitution immediately, select Permanently Replace Unavailable Fonts and click OK.
4) To apply permanent font substitution at later time in the session before closing the form design, select Tools > Missing Fonts and select Permanently Replace Unavailable Fonts.
5) Save the form design.

Adding a new mapping to the font-mapping table in Designer

Designer uses the font-mapping table stored in the Designer.xci file to speed up the substitution of an unavailable font with one of the fonts on your computer. It contains several default entries for font mapping.

The Designer.xci file is located in the installation directory. However, as soon as Designer starts, the XCI file is copied to the user directory.

You can modify the default entries font-mapping table in the Designer.xci file.

Syntax

```
<equate from='input_font_*_*' to='Designer_font_*_*' force="0"/>
```

Parameters

input_font

The name of the font used in the input PDF file.

Designer_font

The name of the font that is installed on your computer.

Asterisk (*)

Wildcard character. For example, the string `*Cour*` refers to all font names that contain the `Cour` string.

Question mark (?)

A single-character wildcard.

_*_*

Bold and italic attributes. Valid values after the first underscore are:`*`,`normal`, and`bold`. Valid values after the second underscore are `*`, `normal`, and `italic`.

force="0"

Instructs Designer to search for the font on the system and use font mapping only if the font is unavailable.

force="1"

Instructs Designer to map the font whether it is installed or not.

Example

The following line instructs Designer to map Courier font to Courier New when Courier is not available:

```
<equate from='Courier_*_*' to='Courier New_*_*' force="0"/>
```

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

Font-mapping table in ConvertPDF_FontMap.txt

Character-mapping table in ConvertPDF_CharMap.txt

Font-mapping table in ConvertPDF_FontMap.txt

You can set up a font mapping table in the ConvertPDF_FontMap.txt file.

If you add a font-mapping rule to this file, all occurrences of the substituted font are changed to the new font when you import a PDF, regardless of the font-mapping table in the Designer.xci file and regardless of whether the substituted font is available on your computer.

The ConvertPDF_FontMap.txt file is located in the installation directory and is empty by default.

Syntax

```
input_font=Designer_font
```

input_font

The name of the font used in the input PDF file.

Designer_font

The name of the font that is installed on your computer.

Asterisk (*)

Wildcard character. For example, the string `*Cour*` refers to all font names that contain the string `Cour`. Adding wildcard characters before and after the *input_font* name accounts for any slight variations in the name of the font in the form.

Question mark (?)

Single-character wildcard.

Example

```
*Arial*=Arial
*Helv*=Arial
*Cour*=Courier New
*=Times New Roman
```

Processing occurs in the same order as in the file. The last line in the example converts all remaining fonts to Times New Roman.

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

Adding a new mapping to the font-mapping table in Designer

Character-mapping table in ConvertPDF_CharMap.txt

Character-mapping table in ConvertPDF_CharMap.txt

PDF files may contain characters that are mapped differently in Designer. To deal with this issue, you can use character-mapping rules in the character-mapping table in the ConvertPDF_CharMap.txt file.

Designer relies on the character-mapping table to display input PDF symbols correctly in its native character set. The table contains several default entries.

If any of the lines in the ConvertPDF_CharMap.txt file do not reflect your setup, you can remove them. You can add additional lines if required.

The ConvertPDF_CharMap.txt file is located in the installation directory.

Syntax

```
input_char,input_font=Designer_char,Designer_font
```

input_char

The input PDF symbol decimal value.

input_font

The specific font name in the PDF that the character must belong to. An asterisk (*) indicates any font.

Designer_char

The UTF-8 decimal value of the same character.

Designer_font

The appropriate font that contains the character. An asterisk (*) indicates to leave the original font name as is.

Example

The following line converts a PDF double quotation mark (“) in any font to the UTF-8 equivalent in the same font:

```
0144,*=0039,*
```

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

Adding a new mapping to the font-mapping table in Designer

Font-mapping table in ConvertPDF_FontMap.txt

Importing PDF documents as artwork

If you have a PDF document that was created in Acrobat or some other application, which you would like to use as the basis for creating a new PDF form in Designer, you can use the Create an Interactive Form with Fixed Pages option in the New Form Assistant to import the content of a PDF document like this as the background artwork for a new form.

If the PDF document was created in Acrobat and contains form fields, Designer converts those fields to the corresponding library objects. After you import a PDF as artwork, you can place additional library objects on top of the artwork to augment the new form design.

If you want to change the background artwork at any point, you can use the Replace Artwork command in the Edit menu to import a substitute PDF document.

Keep in mind the following factors when importing a PDF document as artwork:

- You can only import PDF documents as artwork if the document was created in Acrobat or some other application capable of generating PDF documents. If the PDF document was created using Designer, you cannot import the contents as artwork; you can only import the contents as editable form objects.
- After you import a PDF document as artwork, you must save the resulting form as an Adobe Static PDF Form. You cannot save the form as an Adobe Dynamic XML Form or Adobe XML Form. The default file type for new forms that contain PDF artwork is Adobe Static PDF Form (*.pdf).
- You can import unstructured or structured PDF documents that contain tags and have a defined tabbing and reading order for keyboard access and screen readers. For more information, see *About importing structured PDF documents as artwork*.
- The following library objects are unavailable in the Standard and Custom groups when importing a PDF document as artwork: circle, content area, image, line, rectangle, subform, table, text, masked field, masked field - partial, page n of m, sheet n of m, signature - print and sign, and survey question.
- Buttons that are labeled with icon images in the imported PDF document are not supported.
- If you open a PDF document that contains background art in an earlier version of Designer than the version used to create the document, the image data in the document may not be displayed correctly.
- Document, page, and field-level JavaScript script (in the PDF document imported as artwork) is converted into event scripts that are commented out. You must verify and update the scripts to match the Designer model. You can no longer edit this script in Acrobat.

After importing a PDF document as artwork, you can work with the resulting document the same as with any static PDF form.

To import a PDF file as artwork in the stand-alone version of Designer

1) Select File > New.
2) In the Getting Started panel of the New Form Assistant, select Import a PDF Document and click Next.
3) In the Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
4) In the Document Setup: Import Options panel, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
5) Click Next.
6) [Optional] In the Form Return Setup: Adding Buttons panel, select how the form is distributed and how the form data is returned.
7) Click Finish.

To use File > New to import a PDF file as artwork with Designer and Workbench

When you use File > New to import a PDF file, you can specify a name for the form. The form will be saved in the Applications view in Workbench.

1) Select File > New. The New Form dialog box opens in Workbench.
2) Follow the onscreen instructions until the New Form Assistant > Getting Started panel opens in Designer.
3) Select the Import A PDF Document option, and then click Next.
4) In the Document Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
5) In the Document Setup: Import Options panel, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
6) Click Next.
7) Click Finish.

To use File > Open to import a PDF file as artwork with Designer and Workbench

When you use File > Open to import a PDF file, you specify the AEM forms application where you want to save the form.

1) Select File > Open.
2) Navigate to the PDF file to import, select the file, and click Open.
3) In the Document Setup: Import Options panel, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
4) Click Finish.
5) Select File > Save As.
6) Navigate to the Workbench folder on your local system:

- If you are using Windows XP, the Workbench folder is located in \Documents and Settings\<user name>.
- If you are using Windows Vista, the Workbench folder is located in \Desktop\<user name>.

7) Select the application folder where you want to save the form.
8) Type a name for the form, and then click Save.

To replace PDF artwork

You can replace the background artwork in a PDF form by using the Replace Artwork command in the Edit menu. Designer replaces the current background artwork with the new background artwork, while preserving any field objects that were placed on top of the previous background artwork. Any Acrobat form fields in the replacement PDF artwork are ignored and removed.

Keep in mind that replacing the PDF artwork affects the PDF structure. For more information, see *About importing structured PDF documents as artwork*.

IMPORTANT: *You cannot undo replacing artwork. As a result, it is recommended that you save your PDF form before replacing the artwork.*

1) Select Edit > Replace Artwork.
2) Select File > Save and save the form in one of the file formats in the Save As Type list.

RELATED LINKS:

Saving forms for Acrobat and Adobe Reader

Import documents

About importing structured PDF documents as artwork

You can import structured PDF documents as artwork, which contains tags and structural information to provide a defined tabbing and reading order for assistive technologies such as keyboard access and screen readers.

The PDF Structure palette in Designer provides a view of the hierarchical structure of a tagged PDF and displays three types of structural tags:

<content>

Content tags apply to the content within the document. Content tags are similar to HTML tags and are displayed as <P>, <H1>, <Sect>, <Figure>, and so on. The content of the document is displayed below these tags. Content tags are provided for reference only and cannot be modified in Designer.

<form>

Form tags act as containers for field tags. All structure for library objects is contained under the form tags.

<field>

Field tags correspond to the actual objects in the document. Field tags can only exist under form tags. Each field tag can be linked to a single object. When a field tag is linked to an object, the tag

displays the name of the object along with an icon indicating the object type in the document. When a field tag is unlinked from any object, the tag displays the words *unlinked field*.

When you import a structured PDF document as artwork, Designer maintains the structure and automatically associates any tagged Acrobat form fields that are converted to library objects with the correct form tags in the document structure. However, you may need to manually add field tags under the correct form tags for any untagged fields that remain in the form. You can also place additional library objects on top of PDF artwork and add field tags to the document structure for each object.

Using the PDF Structure palette, you can see which objects are associated with the structure tags in the form. When you click a content tag in the PDF Structure palette, the corresponding object in the form is highlighted in the Design View tab. For example, if you click a paragraph content tag <P> displayed in the PDF Structure palette, the corresponding content, usually text, is highlighted in the form.

After you import a structured PDF document as artwork, you can use the commands on the PDF Structure palette menu (and context menu) to add field tags to and remove them from the document structure, navigate the structure, move fields up and down in the structure, and show the content tags within the structure.

NOTE: Keep in mind that you can only edit the structure for fields in Designer. If you want to edit the structure tags and content structure in a form, you must do so in Acrobat.

Replacing PDF artwork affects document structure

Keep in mind that when you replace the background artwork in a structured PDF document, the current structure is replaced with the structure in the new background artwork. All content tags for the old artwork are replaced with the content tags for the new artwork. The form and field tags remain and are merged with the new content tags. Any content in the field objects is preserved and merged with the new content structure. You may need to move some of the form and field tags if the merged tags are not in the intended locations.

Note that any form fields that were added to the replacement PDF artwork in Acrobat are not converted to the corresponding editable library objects in Designer.

If the replacement PDF artwork is not structured, the structure of the current PDF artwork is removed and the resulting PDF form no longer has structure.

To add a field tag to the document structure

You can select a field object in a form and then add a field tag under a selected form tag in the document structure. You can add one field tag under each form tag.

You can also create a field tag by dragging a field in the form onto a structure tag in the PDF Structure palette.

1) In the Design View tab, select the field in the form to add to the document structure.
2) In the PDF Structure palette, right-click the form tag under which to add the field tag and select Add Field to Structure.

To remove a field tag from the document structure

1) In the PDF Structure palette, right-click the field tag to delete and select Remove Field from Structure.

To move a field tag up or down in the document structure

1) In the PDF Structure palette, right-click field tag to move and select Move Field Up or Move Field Down as needed.

To find untagged fields in the document structure

1) Right-click in the PDF Structure palette and select Find Untagged Fields.

To show the context tags in the document structure

1) Right-click in the PDF Structure palette and select Show Content tags.

RELATED LINKS:

Import documents

About importing structured PDF documents as artwork

PDF Structure palette menu

Importing Word files

You can import a Microsoft Word XP or Microsoft Word 2003 file into Designer. The Word file can be a document (DOC), a template (DOT), or a rich text format (RTF) file.

Word objects, including paragraphs, tables, images, line art, form fields, headers, and footers, are converted into Designer XML objects, and all rich text formatting is preserved.

NOTE: If the Word file contains contiguous underlined tab characters (where underscore is used as the tab character), Designer attempts to convert the information into a text field.

Designer displays the file import options each time you import a Word file.

1) Select File > Open.
2) From the Files Of Type list, select Microsoft Word Document (DOC, DOT, RTF).
3) Navigate to the file that you want to open, select the file, and click Open.
4) In the File Import Options dialog box, type a password if it is needed to open or unprotect the file.
5) To substitute any missing fonts and suppress all font substitution error messages, select Ignore Missing Fonts.
6) To convert and embed any images into the form design, select Convert Images.
7) Select a logging option from the Generate Log File list.
8) Click OK.

NOTE: Because Designer accesses the Windows clipboard to convert data, the clipboard cannot be accessed by other applications while Designer is importing a Word file.

How Designer converts Word objects and settings

When Designer converts a Word file, it preserves the layout. However, you may need to do some manual repositioning of objects afterward to make the form look as good as possible.

This table identifies the key objects in a Word file and indicates how Designer treats them when they are imported.

Item	Converted	Not Converted
File properties	Built-in and custom properties are converted to form properties.	
Paragraphs	Content (for example, text), including rich text formatting Font effects such as underline, strikethrough, superscript, and subscript Borders and shading applied to paragraphs	Hyperlinks Color underline is converted to black underline. All Caps font effect Font color of list numbers if it is different from the text color Shading for Empty paragraphs Borders applied to text
Tables	Tables are converted to subforms that position content, and table cells are converted to child subforms.	Some border styles are not supported.
Shapes	Text boxes Images Lines, rectangles, and circles are rasterized. Auto shapes OLE objects Frames ActiveX® controls are converted to images.	Arrowheads are converted to straight lines. Drawn objects are converted to images. Line slope Drawing Shape Rotation other than multiples of 90 are not supported. Blank lines in frames ActiveX control functionality
Inline shapes	Images and OLE objects are treated like characters and are positioned as if they were a character in a line of text. ActiveX controls are converted to images.	ActiveX control functionality

Item	Converted	Not Converted
Form Fields	Checkbox fields are converted to check boxes. The default state is carried over. Drop-down fields are converted to drop-down lists. All list items are carried over. Text input fields are converted to text fields and the default value is retained. HelpText and StatusText contents are converted to tooltips.	Formatting of Textinput values Web Tools objects
Headers and Footers	If all headers and footers are the same, they are created on the master page and are positioned according to the Word document's header and footer layout settings. If any headers and footers are different, they are created on individual pages.	Embedded field codes such as date and time
Page Setup settings	The Margins settings are used to specify the layout of the default subform for the page. Multiple pages are converted to single individual pages. The Paper Size and Orientation settings determine the basic page layout properties.	Gutter settings
Scripts	Any scripting associated with a form field is converted into comments in the Designer XML source code.	
Page borders	If specified, this attribute is converted to a border on the master page.	
Line numbers		Not converted
Comments		Not converted
Web page attributes		Not converted

About reducing Word conversion problems

Microsoft Word XP or Microsoft Word 2003 must be installed on the same computer as Designer.

If you are importing a Word XP file that is configured to be sent as an email message, ensure you have installed Word XP Service Pack 1. Otherwise, Word XP will crash after the import process has completed.

To obtain the best results, the input file should contain no more than ten pages.

Copying spreadsheet data from Microsoft Excel

You can copy spreadsheet data from Microsoft Excel 2002 or later and paste the cells into a form in Designer. Copying and pasting data from an earlier version of Excel may produce unexpected results.

Designer creates a table object in the form with the same dimensions, number of cells, and data content as the selection in the spreadsheet. Designer also preserves some of the original formatting of the spreadsheet cells, including the text, fonts, colors, background colors, alignment, and borders. If the selected cells in the spreadsheet contain column headers, Designer maintains the column header formatting in a header row at the top of the table.

Keep in mind that Designer does not maintain any calculations, scripts, images, and embedded objects, such as graphs that are in the Excel spreadsheet.

1) In Excel, copy the spreadsheet cells you want, and keep Excel open.
2) In Designer, select Edit > Paste.
3) Select Paste Cells As a Table and click OK.

RELATED LINKS:

Creating a form from the content of a Microsoft Excel spreadsheet

Spreadsheet Paste Options dialog box

Creating a form from the content of a Microsoft Excel spreadsheet

This feature is available only in the stand-alone version of Designer.

You can use a Microsoft Excel spreadsheet as the basis to create a new PDF form, which you can distribute to recipients by email to collect the same type of data as in the spreadsheet.

Using the New Form Assistant, you can create a PDF form with interactive text fields based on the column headers in the Excel spreadsheet. The New Form Assistant creates one text field object for each column in the spreadsheet. Text field objects are positioned in the form from left to right and wrapped onto additional lines if needed. The caption and binding name of each text field object corresponds to the column header text in the spreadsheet.

If you include an email submit button object in the form, you can distribute the form to recipients to fill and return. When you select the Distribute Form command from the File menu, the form closes in Designer and the Acrobat Distribute wizard appears to guide you through the distribution process. As you receive the filled forms from the recipients, you can organize the collected data into data sets in Acrobat. After you receive multiple sets of form data, you can reorganize the form data into a spreadsheet.

Keep in mind that Designer does not maintain any calculations, scripts, images, and embedded objects such as graphs that are in the spreadsheet document.

1) In Excel, open the spreadsheet you want to use to create a new PDF form.
2) In Designer, select File > New.
3) Select Based on a Spreadsheet, and click Next.

4) Follow the on-screen instructions in the New Form Assistant. You must leave Excel running to complete the process in the New Form Assistant.

RELATED LINKS:

Copying spreadsheet data from Microsoft Excel

Spreadsheet Paste Options dialog box

Importing InfoPath files

You can import Microsoft InfoPath files (XSN) into Designer. During the conversion process, Designer extracts these components from the input file and uses them to create Designer XML elements:

- An XML file describing the contents of the package
- An XML file describing the InfoPath form
- XSL files describing each view
- A schema file for the InfoPath form
- A script file, which is saved as comments in the Designer XML source code

Designer displays the file import options each time you import an InfoPath file.

1) Select File > Open.
2) From the Files Of Type list, select InfoPath (XSN).
3) Navigate to the file that you want to open, select the file, and click Open.
4) Do one of the following actions:
   – If you want to be prompted for a view, select the Prompt To Select View During Conversion option. If the InfoPath file has only one view, a prompt will not display even if you select this option.
   – If you know the name of the view that you want to import, type the name of the view in the View box. To convert the default view, leave the box blank.
5) Do one of the following actions:
   – Select the Import Print Settings option to import headers and footers, page orientation, and page margins of the InfoPath form.
   – If you know the page size of the InfoPath form, select one from the list or select Default.
   – If you know the page orientation of the InfoPath form, select one from the list.

   NOTE: If you select the Import Print Settings option, the page size and orientation are taken from the InfoPath form and you do not have to select the page size and orientation.
6) (Optional) Select an option from the Generate A Log File list.
7) (Optional) To convert and embed any images into the form design, select Embed Images In XDP.
8) Click OK.

   NOTE: *Designer uses Microsoft Internet Explorer 5.0 or later to transfer data while importing an InfoPath file. If you do not have Internet Explorer installed, you cannot import an InfoPath file into Designer.*

How Designer converts Microsoft InfoPath objects

Designer preserves the layout when converting a Microsoft InfoPath file. However, you may need to do some manual repositioning of objects afterward to make the form look as good as possible.

The table identifies the key objects in an InfoPath form and indicates how Designer treats them when they are imported.

Item	Converted	Not Converted
Controls	Text boxes, including rich text boxes, are converted to text fields. Drop-down list boxes are converted to drop-down list boxes. Values are included only if the control is filled manually. List boxes are converted to list boxes. Date picker controls are converted to Date/Time fields unless the data type for the date picker in InfoPath is set to “Text (String)”. In this case, it is converted to a Text field. Check boxes are converted to check boxes. Option buttons are converted to radio buttons. Buttons are converted to buttons.	Validation rules Scrolling Images embedded in rich text boxes A group of option buttons may not be converted to a single radio button group. Submit actions for buttons are not handled Padding properties of check boxes and radio buttons Attachments Custom controls
Expression boxes		Not converted
Lists	Numbered lists are converted to text. For bulleted lists, the text only is converted to text.	Bullet symbols
Pictures	Pictures are converted to images.	Ink pictures
Schema	If the InfoPath form has been designed based on a schema, the schema is imported and binding is done automatically in Designer.	
Scripts		Not converted
Scrolling regions	Scrolling regions are converted to subforms.	

Item	Converted	Not Converted
Sections	Sections are converted to subforms. Repeating sections are converted to repeating subforms in a form. Add and delete buttons are provided to add or delete the instance of the section when filling the form.	Include/exclude properties Instructional text Only one instance of any optional or repeating sections is converted
Tables	Tables are converted to tables. Repeating tables are converted to repeating tables in a form. Add and delete buttons are provided to add or delete the instance of the section when filling the form.	Some border styles are not supported. Only the visible rows in a repeating table are converted. Rows that have been merged vertically are converted to subforms in a positioned layout.
Text	Text is converted to text. Any highlight color is converted to background color.	Hyperlinks

Differences between Designer and InfoPath objects

Designer and InfoPath do not support the same features and object properties. The most obvious differences include these:

- Images cannot be placed in Designer text fields.
- Designer does not support editable sequential numbered or bulleted lists.

Importing Adobe Output Designer Form files

To import Output Designer files (IFD) into Designer, you must install Output Designer 5.5 or later on the same computer as Designer.

Designer displays the File Import Options dialog box each time you import an Adobe Output Designer Form file.

1) Select File > Open.
2) In the Files Of Type list, select Adobe Output Designer Form (IFD).
3) Select the file to open, and click Open.
4) In the File Import Options dialog box, select options as needed:
   – To generate a temporary log file, in the Generate a Trace Log of the Conversion Process list select one of the following:
     - **\TEMP\ConvertIFD.log** Creates a log file in your systems's temporary folder that records information about the conversion process.
     - **\TEMP\inputfile.log** Generates a log file in your system's temporary folder using the same name as the input file. Generates a log file for each imported file.

- To use the Output Designer configuration file to generate PDF form file, select Use Output Designer PDF Configuration. Uses an Output Designer configuration file (pdf.ics) and a compiled version of the configuration file (pdf.icf) to generate a PDF form file.
- To synchronize field objects with matching names to use global data binding, select sync Like Named Fields As Global. Applies a global value to all field objects in the form with the same name.
- To remove all un-named subforms from the hierarchy, select Import Fields Only.

5) Click OK.

RELATED LINKS:

Adobe Output Designer form (File Import Options dialogbox)

Import documents

Importing XForms Model files

You can import XForms Model files into Designer. Each time you import an XForms Model file, the File Import Options dialog box appears.

1) Select File > Open.
2) In the Files Of Type list, select XForms Model.
3) Navigate to the file to open, select the file, and click Open.
4) (Optional) To embed a schema within the output XDP file if a schema is set for the XForms document, select Embed Schemas Inline.
5) (Optional) In the Set Working Directory box, type the path to the folder where you want Designer to check for relative file path locations, such as instance data XML files or XML schema files, which might be included in the XForms document. If you do not specify this folder, the default working directory will be the same location as the XForm that is being imported. This directory is not where the log files or temporary instance data files are created.
6) In the Generate log file list, select one of the following options:
   - If you do not want to generate a log file to capture conversion output messages, select Do Not Log.
   - To generate a log file in the Temp folder located in Documents and Settings\[*username*]\Local Settings\Temp (Windows XP) or Users\[*username*]\AppData\Local\Temp (Windows Vista), select \TEMP\ConvertXF.log. Each log file created is prefixed with ConvertXF and appended with a unique number (for example, ConvertXF38512.log).
7) Click OK.

NOTE: Designer forms do not support an equivalent of the `xforms-valid` event. Any scripting associated with the `xforms-valid` event is not imported.

RELATED LINKS:

XForms Model (File Import Options dialog box)

Import documents

5. Guidelines for forms

When creating forms in Designer, you can save your form designs in several different formats. Form author can save a form design as a PDF form so that form fillers can easily access the form using Adobe Reader®. When saving a form design as a PDF form in Designer, you can select between two different types of PDF forms: Adobe Static PDF Form (*.pdf) and Adobe Dynamic XML Form (*.pdf). Form designs saved as static and dynamic PDF forms can be interactive or non-interactive.

RELATED LINKS:

Creating interactive forms that have a flowable layout

Creating non-interactive forms that have a flowable layout

5.1. Rendering PDF forms

To understand the difference between static and dynamic PDF forms, it is important to understand what the term render means. Rendering a form is the process of creating the precise final layout and formatting from the form design. The form may or may not be merged with data, depending on the form design and data. Rendering can be done by AEM forms on the server or by Adobe Reader on the client. A PDF form must be rendered before it can be displayed to the end user or printed. Therefore, the rendering is the final image.

Characteristics of static PDF forms

Form designs saved as static PDF forms render once on the server and are displayed on the client in the Acrobat or Adobe Reader target version. They are not rerendered in response to user interaction. The form may have been designed with a flowable layout. However, when the static PDF form is created, its layout is fixed and the resultant PDF form will not rerender on the client.

File size

In general, forms saved as static PDF forms render to larger file sizes than the equivalent files saved as dynamic PDF forms.

Rendering location

Static PDF forms render once and are displayed on the client in Acrobat or Adobe Reader. They are not rerendered in response to user interaction. Because rendering is performed on the server, only small changes to the final output are possible on the client. For example, the value area may show a different color after the user clicks a button or exits a field.

Display speed

In general, static PDF forms require more time to render on the server than dynamic PDF forms but are displayed faster in the client software on the end-user's computer. Static PDF forms require more time on the server because the server performs the page layout operations, data merging,

and final rendering. In the case of dynamic PDF forms, the server performs only data merging. The client software on the end-user's computer performs the page layout operations and final rendering. As a result, the rendering time for dynamic PDF forms relies on the processing power of the end-user's computer, especially when the PDF file size is large.

Characteristics of dynamic PDF forms

Forms saved as dynamic PDF forms render on the client in Acrobat or Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of objects is possible in Acrobat or Adobe Reader because Acrobat or Adobe Reader have enough information to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If the end user clicks a button that adds a new row to a table, the form is rerendered in Acrobat or Adobe Reader. Dynamic PDF forms were first introduced in Adobe Reader, Acrobat Professional and Acrobat Standard 7.0.

File size

In general, forms saved as dynamic forms render to smaller file sizes than the equivalent files saved as static PDF forms.

Rendering location

Form designs saved as dynamic PDF forms render on the client in Acrobat or Adobe Reader. Because rendering is performed on the client, the form can rerender several times based on end-user interactions.

Display speed

In the case of dynamic PDF forms, the server performs only data merging. The client software on the end-user's computer performs the page layout operations and final rendering. As a result, the rendering time for dynamic PDF forms relies on the processing power of the end-user's computer, especially when the PDF file size is large.

RELATED LINKS:

Choosing the type of PDF form

5.2. Choosing the type of PDF form

You can create many kinds of forms as either static or dynamic with little difference to the end-user experience. However, some forms work as designed only if they are created as a static or dynamic PDF forms. In general, the choice of static over dynamic is determined by the following considerations:

- If the form works as either a static PDF form or a dynamic PDF form, use a dynamic PDF form to reduce server-side processing, which results in a greater number of transactions per second.
- If the form relies on client-side scripts to change the layout (for example, it uses scripts to add or remove rows from a table or to make text fields grow), use a dynamic PDF form.
- If end users will need to add annotations or comments to the PDF form, use a static PDF form.

- If the form must work with Acrobat installations earlier than version 7.0, use a static PDF form.

Additionally, there are a number of specific issues to consider when choosing to create a static or dynamic PDF form. These issues are differences in the behavior between the form types that may be critical in making your decision:

- *Applying formatting by using client-side scripts*
- *Saving object formatting*
- *Unavailable commands for dynamic forms in Adobe Reader*
- *Font behavior*
- *Objects and properties for static PDF forms*
- If your form includes digital signatures, there are limitations to the way that you can use dynamic PDF forms with the Signature service. For more information about these limitations, see AEM forms Services Reference.

As you design a form, you can see how the form behaves as a static or dynamic PDF form in the Preview PDF tab.

5.3. Applying formatting by using client-side scripts

In a static PDF form, only the value area of the field can be updated on the client. Everything else on the form is frozen or fixed. In a dynamic PDF form, the entire field object can be updated on the client. The examples that follow are provided to give you an idea of what happens.

Changing the field border color

If you create a client-side script to apply borders to objects, the results differ in static and dynamic PDF forms.

For example, you can write a script on the exit event of a field to turn the border color red:

```
TextField1.border.edge.color.value="255,0,0"
```

In a static PDF form, the outline of only the value area turns red after the end user exits the field.

In a dynamic PDF form, the outline of the entire field object turns red after the end user exits the field.

Applying field shading

If you create a client-side script to apply shading to objects, the results differ in static and dynamic PDF forms.

For example, you can write a script on the exit event of a field to turn the shading color in the field to red:

```
TextField1.fillColor = "255,0,0"
```

In a static PDF form, only the value area turns red after the end user exits the field.

In a dynamic PDF form, the entire object, including the value area, turns red after the end user exits the field.

Hiding objects

If you create a client-side script to hide objects, the results differ in static and dynamic PDF forms.

When you are designing a static or dynamic PDF form, you can hide objects on the form by setting the presence property to either “invisible” or ”hidden”. In both cases, the objects do not appear in the final output; however, when you set the presence value to “hidden”, objects do not occupy any space in the layout. If the objects are inside flowed subform objects, the subform itself will shrink in response to the hidden objects.

The difference between static and dynamic PDF forms is that on a static PDF form you cannot change the presence value of an object by using an interactive scripting event, such as the click event of a button. Static PDF forms cannot rerender on the client; therefore, scripts executed on interactive events cannot change the visibility of form objects. Dynamic PDF forms do not have the same limitation because they can rerender on the client.

You can change the visibility of form objects on static PDF forms by using non-interactive events that trigger during form rendering, such as the initialize event. For example, on a dynamic PDF form, you can write a script on the click event of a button to hide a text field:

```
TextField1.presence = "invisible"
```

Alternatively, you can use this script to completely remove the text field from the layout:

```
TextField1.presence = "hidden"
```

In both cases, to achieve the same results on a static PDF form, you use the same scripts but write them on the initialize event of either the text field or the button.

RELATED LINKS:

Saving object formatting

5.4. Saving object formatting

Prior to Acrobat 8.0, in a static PDF form, the object’s formatting is saved when the end user saves, closes, and reopens the form. In a dynamic PDF form, the object’s formatting is not saved.

IMPORTANT: *Although it is possible to save an object's formatting on a dynamic PDF form by using client-side scripting, it is not a recommended practice and should be avoided. If saving an object's formatting is a requirement, you need to use a static PDF form.*

Beginning with Acrobat 8.0, the state of all the objects is saved and can be restored automatically or manually upon reopening. This is controlled by the `restoreState` scripting property on the root subform.

Object formatting, as well as the layout of a dynamic PDF form, can change when any of the following form actions occur:

- The user opens the form
- The user imports new data
- The user adds new data to the form
- A client-side script is run

The following table compares the end-user experience using the examples in the topic *Applying formatting by using client-side scripts* when saving as either a static PDF form or a dynamic PDF form in Adobe Reader.

Form content	Static PDF form	Dynamic PDF form
A client-side script to change the border color of a field.	The outline of the value area remains red when the form is saved, closed, and reopened.	The outline of the object does not remain red when the form is saved, closed, and reopened.
A client-side script to apply shading.	The fillable value area remains red when the form is saved, closed, and reopened.	The object, including the value area, does not remain red when the form is saved, closed, and reopened.
A client-side script to hide objects.	The value in the text field remains invisible when the form is saved, closed, and reopened.	The text field does not remain hidden when the form is saved, closed, and reopened.

NOTE: If `restoreState` is set to `auto`, all of these changes are maintained if the document is saved and then reopened in Acrobat 8.0.

For more information, see *restoreState* in the *Scripting Reference*.

5.5. Unavailable commands for dynamic forms in Adobe Reader

When a PDF form is opened in Adobe Reader, some Adobe Reader commands that are available for static PDF forms are not available for dynamic PDF forms. The reason is because the final layout of the form can change the position of objects, which in turn affects the number of form pages.

The following Adobe Reader commands are available for static PDF forms but not for dynamic PDF forms:

- Comment and Markup tools

- Import Comments and Export Comments
- Insert Pages, Extract Pages, Replace Pages, Delete Pages, Crop Pages, and Rotate Pages

RELATED LINKS:

Choosing the type of PDF form

Target version warning messages

5.6. Font behavior

When you create PDF forms, it is possible to include only those characters of a font that are actually used in the form. This technique is called font subsetting. You can also embed fonts into the form so that end users have all of the fonts they need to use the form. In this case, the size of the PDF file is larger.

NOTE: The manufacturer of the font can specify the level of embedding that is allowed.

Fonts can be subset in static PDF forms

In static PDF forms, fonts can be subset for text and field captions. Subsetting reduces the size of the PDF file. To subset fonts into the form, the font you select must be set to the Print & Preview Embedding Allowed level.

NOTE: The fonts used in the fillable area of fields cannot be subset on static PDF forms because the user input can contain any character from the font.

Fonts are embedded in dynamic PDF forms

In dynamic PDF forms, fonts are fully embedded or linked. Embedded fonts are never subset. When a form that has embedded fonts is opened, even if the fonts are already on the system, Adobe Reader uses the embedded fonts. To embed fonts into the form, the embedding settings in the fonts you select must allow embedding for edit.

RELATED LINKS:

Choosing fonts for performance

5.7. Objects and properties for static PDF forms

Here is the list of scripting objects that form authors can use to make changes to fields in static PDF forms in Adobe Reader. For more information about scripting objects, see Scripting Objects.

Object	Description
border	Describes the border surrounding an object
color	Describes a unique color on an object
fill	Applies a color and optional rendered designs to the region enclosed by an object

Here is the list of scripting properties that form authors can change for fields in static PDF forms in Adobe Reader.For more information about scripting properties, see Scripting Properties.

Property	Description	Syntax
access	Controls user access to the contents of a container	*SOMexpression*.access = "open | readOnly|protected"
fontColor	The font color value for the text	*SOMexpression*.fontColor = "[0-255], [0-255], [0-255]"
hAlign	Specifies the horizontal text alignment	*SOMexpression*.hAlign = "left | center | right"
maxChars	Specifies the maximum number of characters that this text value can enclose	*SOMexpression*.maxChars = "0 | integer"
maxLength	Specifies the maximum (inclusive) permitted length of the content or -1 to indicate that no maximum length is imposed	*SOMexpression*.maxLength = "0 | integer"
multiLine	Specifies whether the text may span multiple lines	*SOMexpression*.multiLine = "0 | 1"
presence	Specifies the visibility of an object's value	*SOMexpression*.presence = "visible | invisible|hidden"
textEntry	Determines whether a user can type a value into a drop-down list	*SOMexpression*.textEntry = "0 | 1"

6. Import documents

You can import forms into Designer that were created in another form authoring application or an XForm XML application, or import spreadsheet data from Microsoft Excel 2002 or later by copying and pasting the content of the spreadsheet cells into a form. You can also import legacy forms as a starting point for creating new Designer forms. You can import files from these applications into Designer.

Application	File types
Acrobat	PDF
Adobe Form Designer 5.0 or later	XFT
Adobe Output Designer	IFD To import Output Designer files (IFD) into Designer, you must install Output Designer 5.5 or later on the same computer as Designer.
Microsoft Word	DOC, DOT, RTF
Microsoft Excel	XLS, XML, HTML, XLT, TXT
Microsoft InfoPath	XSN
XForms Model	XHTML, XML, HTML, HTM, XFDL

Designer includes a number of options to handle imported files and render the best possible output. Designer preserves the layout of the imported form as much as possible and converts the elements into Designer objects.

6.1. Importing PDF files

PDF files can come from many sources, and each authoring application can define the contents of the PDF file in a unique manner.

Before importing a PDF file into Designer, you should understand the different options available for fine-tuning the results of the import process.

NOTE: PDF files created by using Designer do not go through the import process. They open directly in Designer and should appear as designed without modifications.

You use the New Form Assistant to import PDF files. Using the New Form Assistant simplifies the process. It guides you through a number of steps in which you choose the PDF file to import, how you want to work with the imported PDF content, and how the form is distributed and returned.

When Designer is integrated with Workbench, you can also use Workbench to import PDF files. In Workbench, select File > New > Form and then follow the onscreen instructions. On the Getting Started panel of the New Form Assistant, select the Import A PDF Document option.

If you are using the stand-alone version of Designer, you can also use the Import Options dialog box to import a PDF file. The Import Options dialog box is not available when you use Designer with Workbench. Using the Import Options dialog box gives you more control over the way Designer imports the contents of the PDF file when creating an interactive form with a flowable layout. For example, you can select custom options that control how Designer processes the text, paragraphs, and images in the document. Use one of these methods to display the Import Options dialog box:

- Click the Do Not Use the Assistant link in the New Form Assistant to open the Assistant Options dialog box. Select the option that best suits your needs.
- Select Tools > Options to open the Options dialog box, select Wizards and Tips and, under Form Assistance, deselect Show When Importing Documents. (See *Wizards and Tips (Options dialog box)*.).

Before you import a PDF, consider the following scenarios:

- If you have a PDF document that was created in Acrobat or an application other than Designer, and you want to import the contents as background artwork and maintain the original layout and appearance of the document, select Create an Interactive Form with Fixed Pages. You can edit any interactive objects and place additional library objects on top of the artwork.
- If you want to edit and change the content of the PDF document in Designer, select Create an Interactive Form with a Flowable Layout. Word-wrapping and line breaks may not be preserved.

If you are not satisfied with the initial conversion, you can experiment with different combinations of options to achieve the best results.

NOTE: When importing a PDF file that has security permissions set, Designer prompts you to enter the correct password to prevent unauthorized access.

To import a PDF file by using the New Form Assistant in the stand-alone version of Designer

1) Select File > New.
2) In the Getting Started panel of the New Form Assistant, select Import a PDF Document and click Next.
3) In the Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
4) In the Document Setup: Import Options panel, select the import option you want to use:
   - To import the PDF file as background artwork, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
   - To import the PDF file so that you can edit it in Designer, select Create an Interactive Form With a Flowable Layout.
5) Click Next.
6) [Optional] In the Form Return Setup: Adding Buttons panel, select how the form is distributed and how the form data is returned.
7) Click Finish.

To import a PDF file without the New Form Assistant in the stand-alone version of Designer

1) Select File > Open.
2) Navigate to the PDF file that you want to import, select the file, and click Open. The Import Options dialog box appears. However, if you have not disabled the New Form Assistant by deselecting the Show When Importing Documents option in the Options dialog box (Wizards and Tips panel), the New Form Assistant appears. If necessary, click Do Not Use Assistant to open the Import Options dialog box.
3) In the Import Options dialog box, select the option that best suits how you want to work with the imported PDF content:
   – To preserve the layout and appearance of the PDF content, select Create an Interactive Form with Fixed Pages.
   – To edit the content of the PDF document in Designer, select Create an Interactive Form with a Flowable Layout, and then select custom options for importing the PDF, as required.
4) Click OK. If you selected the Display A Summary Report option, a dialog box lists any conversion issues, such as unavailable fonts and unsupported objects, before the form opens.

   NOTE: The Temporary folder is your temporary directory that is configured by Windows (which is typically \Documents and Settings*[username]**Local Settings**Temp). The log file is named the same as the PDF file with a .log file name extension. If a file with that name already exists in the Temporary Folder, it is overwritten.*

To use File > New to import a PDF file with Designer and Workbench

When you use File > New to import a PDF file, you can specify a name for the form. The form will be saved in the Applications view in Workbench.

1) Select File > New. The New Form dialog box opens in Workbench.
2) Follow the onscreen instructions until the New Form Assistant > Getting Started panel opens in Designer.
3) Select the Import A PDF Document option, and then click Next.
4) In the Document Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
5) In the Document Setup: Import Options panel, select the import option you want to use:
   – To import the PDF file as background artwork, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
   – To import the PDF file so that you can edit it in Designer, select Create an Interactive Form With a Flowable Layout.

     NOTE: *By default, images are embedded in medium quality JPEG format. If an image extraction problem occurs, a message appears in the Conversion Summary and the image is embed in BMP format.*

6) Click Next.
7) Click Finish.

To use File > Open to import a PDF file with Designer and Workbench

When you use File > Open to import a PDF file, you specify the AEM forms application where you want to save the form.

1) Select File > Open.
2) Navigate to the PDF file to import, select the file, and click Open.
3) In the Document Setup: Import Options panel, select the import option you want to use:
   – To import the PDF file as background artwork, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
   – To import the PDF file so that you can edit it in Designer, select Create an Interactive Form With a Flowable Layout.

     NOTE: *By default, images are embedded in medium quality JPEG format. If an image extraction problem occurs, a message appears in the Conversion Summary and the image is embed in BMP format.*
4) Click Finish.
5) Select File > Save As.
6) Navigate to the Workbench folder on your local system:
   – If you are using Windows XP, the Workbench folder is located in \Documents and Settings\<user name>.
   – If you are using Windows Vista, the Workbench folder is located in \Desktop\<user name>.
7) Select the application folder where you want to save the form.
8) Type a name for the form, and then click Save.

RELATED LINKS:

Saving forms

Import Options dialog box

Wizards and Tips (Options dialog box)

6.2. Reviewing the results of a PDF file import

The form design that results from an imported PDF file depends on the different combinations of options that you selected during the import process. If you are not satisfied with the initial conversion, you might want to experiment with different combinations of options to achieve the best results, or you can edit the objects in the form design by using the Object Editor.

The Object Editor appears when you select an object, and it provides quick access to commands that are commonly used when editing objects.

Initially, object boundaries appear in the form design so that you can see how the PDF file elements, especially text, have been grouped into objects. This can make the form design look busy and difficult to read. Using the Object Boundaries command in the View menu, you can hide the object boundaries while you examine the form design. You will find the object boundaries useful if you need to make changes to objects in the form design. For example, you use the object boundaries as a guide when you resize or move objects without borders. Also, if a body of text is divided over two or more objects in the form design, the object boundaries will show it. You can easily merge these objects into one text object by using the Merge Selected Text Objects command in the Layout menu.

Title and instruction text may be divided into multiple text objects and field objects. You can create a caption for a field object that does not have a caption by merging it with a text object using the Merge as Caption command in the Layout menu.

You might also find the locking commands in the Edit menu useful. Locking objects prevents you from selecting and editing them in the Layout Editor. For example, you can lock text so that you must double-click the required text or caption to edit it. This makes it somewhat difficult to modify the text. You can also lock static objects such as text, lines, images, and shapes, and you can lock field objects.

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

About reducing PDF conversion problems

Adding a new mapping to the font-mapping table in Designer

Character-mapping table in ConvertPDF_CharMap.txt

How Designer converts PDF objects

The conversion of PDF contents into XDP format is highly dependent on the source of the PDF file itself. The PDF specification allows for a wide variety of styling and structural layout, some of which lends itself to good migration. Often, PDF is used as a final form layout; that is, the display of the contents is the only thing that matters, not the contents themselves.

In many situations, Distiller® or PDFMaker actually generates the PDF file, but they both can only render what is provided to them in the order that it is provided.

This table identifies key elements in a PDF file and indicates how Designer treats them when they are imported.

Item	Converted	Not converted
Actions (from an Acrobat form)	Designer only converts the following actions that may be associated with a form created in Acrobat: • Import form data • Launch a URL • Reset a form (full and partial field list) • Show/hide a field • Submit a form for complete field list	All remaining actions
Calculations	Average Maximum Minimum Sum Range	
Comments/ Annotations		Not converted
Document	Document-level JavaScript script is converted to event scripts that are commented out. You must verify and update the scripts to match the Designer model. Pages and names Events other than JavaScript scripts Security options Document metadata, including PDF version, title, author, subject, keywords, creator, producer, creation date, modification date Accessibility information of Tagged PDFs is preserved.	Tab order

Item	Converted	Not converted
Fields	Date, Text, Numeric, Check Box, Radio Buttons and groups, Button, Combo box, Signature (only identified), Drop-down List, and List Box field types Default text Dimensions Events Font name, size, bold, italic JavaScript script is converted to event scripts that are commented out. You must verify and update the scripts to match the Designer model. Name Picture formatting Position, orientation, alignment, and visibility Printable Read Only, Required, Maximum Characters, and Multiline attributes RGB and grey color definitions for background color, border color, and color. Other color specifications are mapped to RGB values. Tool tip is preserved	All fonts used in a PDF file must exist on your system or font substitution rules must be defined in the font-mapping table. Designer uses a cross (x) to indicate when a check box is selected. If a check box in the PDF file uses a different character, the character is converted to a cross that may not be perfectly aligned inside the fillable area.
Images	All 1-bit, 4-bit, 8-bit, and 24-bit Dimensions Position	Vector based 6-bit 32-bit Some TIFF formats, including tiled TIFF format
Lines	Color (both edge and fill) Dimensions Position Width Some dotted and dashed styles Style	
Rectangles	Color (both edge and fill) Dimensions Position Width Some dotted and dashed line styles Rounded-corner rectangles are converted when they are detected. The conversion depends on how the PDF file was generated.	Style

Item	Converted	Not converted
Scripts	Document, page, and field-level JavaScripts scripts are converted to event scripts that are commented out. You must verify and update the scripts to match the Designer model.	
Text	Content Dimensions Font name, size, bold, italic Mixed formatting (rich text) Position RGB and grey color definitions for background color, border color, and color. Other color specifications are mapped to RGB values.	All fonts used in a PDF file must exist on your system or font substitution rules must be defined in the font-mapping table. Symbol-based characters are not always displayed correctly. You can define character-mapping rules in the character-mapping table. Designer does not support font kerning. When PDF-based kerning is detected, the font size is decreased so that the text occupies the same horizontal space.

About reducing PDF conversion problems

Some PDF files may be difficult to import into Designer. Conversion problems can depend both on the application used to produce the PDF file and on choices made when the source document was authored. For example, these original sources of PDF files have proven difficult to import into Designer:

- Digi path files, which are converted to one large image that cannot be edited
- Microsoft Word, when the document has not been specifically designed as a form

You can reduce possible conversion problems by being aware of these issues:

- Fonts are substituted if a font used on the form is not installed when you import the form. If a font is substituted, text will not appear exactly as in the original form.
- If you are using Acrobat to create PDF files, do not build the form by using drawing annotations only.
- When an image in a PDF form contains a large number of lines, such as a vector-based image, Designer needs a significant amount of time to process the lines during an import operation. For faster processing, remove the image from the source file that is used to create the PDF. If required, you can reinsert the image after you save the form design in Designer. If the source file is not available, deleting the image from the form in Designer will improve performance of the form.

- If you are having conversion problems and you have access to the source files for the original form, try re-creating the PDF by using Distiller. If the original PDF was not created correctly, re-creating it may resolve the issue.

6.3. Importing tagged PDF files

When importing a tagged PDF (an accessible PDF), the accessibility information of the page contents is preserved. In addition, organizing the form by adding or changing subforms may cause the accessibility information to change.

To ensure that the form still performs in a manner acceptable to your users, test the form thoroughly. Pay close attention to the following areas and adjust them if they are not behaving as expected:

- Tool tips
- Custom screen reader text
- Screen reader reading order

RELATED LINKS:

About accessible forms

Importing PDF files

6.4. Matching unavailable fonts

During import, Designer attempts to match fonts in the PDF file with fonts that are available on your computer. If the file contains an unavailable font, Designer displays the Missing Fonts dialog box showing the missing font and a suggested replacement. You can accept the replacement font or change it. This font substitution is not permanent. Designer must map the unavailable fonts every time the form opens.

After importing the PDF, the Warnings tab of the Report palette lists all unavailable fonts and explains how they were converted. If the font-mapping table includes the unavailable font, the Warnings tab states that the font was mapped to the new font. If the unavailable font is not in the font-mapping table, Designer attempts to select an appropriate substitution. The Warnings tab then states that the unavailable font was changed to the new font. In this case, you may want to permanently substitute the font or add a new mapping to the font table.

To make the font substitution permanent

1) Import the PDF. See *Importing PDF files*.
2) If the Missing Fonts dialog box appears, specify the desired substitute fonts.
3) To apply permanent font substitution immediately, select Permanently Replace Unavailable Fonts and click OK.
4) To apply permanent font substitution at later time in the session before closing the form design, select Tools > Missing Fonts and select Permanently Replace Unavailable Fonts.
5) Save the form design.

6.5. Adding a new mapping to the font-mapping table in Designer

Designer uses the font-mapping table stored in the Designer.xci file to speed up the substitution of an unavailable font with one of the fonts on your computer. It contains several default entries for font mapping.

The Designer.xci file is located in the installation directory. However, as soon as Designer starts, the XCI file is copied to the user directory.

You can modify the default entries font-mapping table in the Designer.xci file.

Syntax

```
<equate from='input_font_*_*' to='Designer_font_*_*' force="0"/>
```

Parameters

input_font

The name of the font used in the input PDF file.

Designer_font

The name of the font that is installed on your computer.

Asterisk (*)

Wildcard character. For example, the string `*Cour*` refers to all font names that contain the `Cour` string.

Question mark (?)

A single-character wildcard.

_*_*

Bold and italic attributes. Valid values after the first underscore are:`*`,`normal`, and `bold`. Valid values after the second underscore are `*`, `normal`, and `italic`.

force="0"

Instructs Designer to search for the font on the system and use font mapping only if the font is unavailable.

force="1"

Instructs Designer to map the font whether it is installed or not.

Example

The following line instructs Designer to map Courier font to Courier New when Courier is not available:

```
<equate from='Courier_*_*' to='Courier New_*_*' force="0"/>
```

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

Font-mapping table in ConvertPDF_FontMap.txt

Character-mapping table in ConvertPDF_CharMap.txt

6.6. Font-mapping table in ConvertPDF_FontMap.txt

You can set up a font mapping table in the ConvertPDF_FontMap.txt file.

If you add a font-mapping rule to this file, all occurrences of the substituted font are changed to the new font when you import a PDF, regardless of the font-mapping table in the Designer.xci file and regardless of whether the substituted font is available on your computer.

The ConvertPDF_FontMap.txt file is located in the installation directory and is empty by default.

Syntax

```
input_font=Designer_font
```

input_font

The name of the font used in the input PDF file.

Designer_font

The name of the font that is installed on your computer.

Asterisk (*)

Wildcard character. For example, the string `*Cour*` refers to all font names that contain the string `Cour`. Adding wildcard characters before and after the *input_font* name accounts for any slight variations in the name of the font in the form.

Question mark (?)

Single-character wildcard.

Example

```
*Arial*=Arial
*Helv*=Arial
*Cour*=Courier New
*=Times New Roman
```

Processing occurs in the same order as in the file. The last line in the example converts all remaining fonts to Times New Roman.

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

Adding a new mapping to the font-mapping table in Designer

Character-mapping table in ConvertPDF_CharMap.txt

6.7. Character-mapping table in ConvertPDF_CharMap.txt

PDF files may contain characters that are mapped differently in Designer. To deal with this issue, you can use character-mapping rules in the character-mapping table in the ConvertPDF_CharMap.txt file.

Designer relies on the character-mapping table to display input PDF symbols correctly in its native character set. The table contains several default entries.

If any of the lines in the ConvertPDF_CharMap.txt file do not reflect your setup, you can remove them. You can add additional lines if required.

The ConvertPDF_CharMap.txt file is located in the installation directory.

Syntax

```
input_char,input_font=Designer_char,Designer_font
```

input_char

The input PDF symbol decimal value.

input_font

The specific font name in the PDF that the character must belong to. An asterisk (*) indicates any font.

Designer_char

The UTF-8 decimal value of the same character.

Designer_font

The appropriate font that contains the character. An asterisk (*) indicates to leave the original font name as is.

Example

The following line converts a PDF double quotation mark (“) in any font to the UTF-8 equivalent in the same font:

```
0144,*=0039,*
```

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

Adding a new mapping to the font-mapping table in Designer

Font-mapping table in ConvertPDF_FontMap.txt

6.8. Importing PDF documents as artwork

If you have a PDF document that was created in Acrobat or some other application, which you would like to use as the basis for creating a new PDF form in Designer, you can use the Create an Interactive Form with Fixed Pages option in the New Form Assistant to import the content of a PDF document like this as the background artwork for a new form.

If the PDF document was created in Acrobat and contains form fields, Designer converts those fields to the corresponding library objects. After you import a PDF as artwork, you can place additional library objects on top of the artwork to augment the new form design.

If you want to change the background artwork at any point, you can use the Replace Artwork command in the Edit menu to import a substitute PDF document.

Keep in mind the following factors when importing a PDF document as artwork:

- You can only import PDF documents as artwork if the document was created in Acrobat or some other application capable of generating PDF documents. If the PDF document was created using Designer, you cannot import the contents as artwork; you can only import the contents as editable form objects.
- After you import a PDF document as artwork, you must save the resulting form as an Adobe Static PDF Form. You cannot save the form as an Adobe Dynamic XML Form or Adobe XML Form. The default file type for new forms that contain PDF artwork is Adobe Static PDF Form (*.pdf).
- You can import unstructured or structured PDF documents that contain tags and have a defined tabbing and reading order for keyboard access and screen readers. For more information, see *About importing structured PDF documents as artwork*.
- The following library objects are unavailable in the Standard and Custom groups when importing a PDF document as artwork: circle, content area, image, line, rectangle, subform, table, text, masked field, masked field - partial, page n of m, sheet n of m, signature - print and sign, and survey question.
- Buttons that are labeled with icon images in the imported PDF document are not supported.
- If you open a PDF document that contains background art in an earlier version of Designer than the version used to create the document, the image data in the document may not be displayed correctly.
- Document, page, and field-level JavaScript script (in the PDF document imported as artwork) is converted into event scripts that are commented out. You must verify and update the scripts to match the Designer model. You can no longer edit this script in Acrobat.

After importing a PDF document as artwork, you can work with the resulting document the same as with any static PDF form.

To import a PDF file as artwork in the stand-alone version of Designer

1) Select File > New.
2) In the Getting Started panel of the New Form Assistant, select Import a PDF Document and click Next.
3) In the Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
4) In the Document Setup: Import Options panel, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
5) Click Next.
6) [Optional] In the Form Return Setup: Adding Buttons panel, select how the form is distributed and how the form data is returned.
7) Click Finish.

To use File > New to import a PDF file as artwork with Designer and Workbench

When you use File > New to import a PDF file, you can specify a name for the form. The form will be saved in the Applications view in Workbench.

1) Select File > New. The New Form dialog box opens in Workbench.
2) Follow the onscreen instructions until the New Form Assistant > Getting Started panel opens in Designer.
3) Select the Import A PDF Document option, and then click Next.
4) In the Document Setup: Import a PDF panel, browse to and select the PDF file you want to import, click Open, and then click Next.
5) In the Document Setup: Import Options panel, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
6) Click Next.
7) Click Finish.

To use File > Open to import a PDF file as artwork with Designer and Workbench

When you use File > Open to import a PDF file, you specify the AEM forms application where you want to save the form.

1) Select File > Open.
2) Navigate to the PDF file to import, select the file, and click Open.
3) In the Document Setup: Import Options panel, select Create an Interactive Form with Fixed Pages. When you import a PDF file as background artwork, Designer preserves the appearance of the original document and retains existing interactive fields.
4) Click Finish.

5) Select File > Save As.
6) Navigate to the Workbench folder on your local system:
   – If you are using Windows XP, the Workbench folder is located in \Documents and Settings\<user name>.
   – If you are using Windows Vista, the Workbench folder is located in \Desktop\<user name>.
7) Select the application folder where you want to save the form.
8) Type a name for the form, and then click Save.

To replace PDF artwork

You can replace the background artwork in a PDF form by using the Replace Artwork command in the Edit menu. Designer replaces the current background artwork with the new background artwork, while preserving any field objects that were placed on top of the previous background artwork. Any Acrobat form fields in the replacement PDF artwork are ignored and removed.

Keep in mind that replacing the PDF artwork affects the PDF structure. For more information, see *About importing structured PDF documents as artwork*.

IMPORTANT: *You cannot undo replacing artwork. As a result, it is recommended that you save your PDF form before replacing the artwork.*

1) Select Edit > Replace Artwork.
2) Select File > Save and save the form in one of the file formats in the Save As Type list.

RELATED LINKS:

Saving forms for Acrobat and Adobe Reader

Import documents

About importing structured PDF documents as artwork

You can import structured PDF documents as artwork, which contains tags and structural information to provide a defined tabbing and reading order for assistive technologies such as keyboard access and screen readers.

The PDF Structure palette in Designer provides a view of the hierarchical structure of a tagged PDF and displays three types of structural tags:

<content>

Content tags apply to the content within the document. Content tags are similar to HTML tags and are displayed as <P>, <H1>, <Sect>, <Figure>, and so on. The content of the document is displayed below these tags. Content tags are provided for reference only and cannot be modified in Designer.

<form>

Form tags act as containers for field tags. All structure for library objects is contained under the form tags.

<field>

Field tags correspond to the actual objects in the document. Field tags can only exist under form tags. Each field tag can be linked to a single object. When a field tag is linked to an object, the tag displays the name of the object along with an icon indicating the object type in the document. When a field tag is unlinked from any object, the tag displays the words *unlinked field*.

When you import a structured PDF document as artwork, Designer maintains the structure and automatically associates any tagged Acrobat form fields that are converted to library objects with the correct form tags in the document structure. However, you may need to manually add field tags under the correct form tags for any untagged fields that remain in the form. You can also place additional library objects on top of PDF artwork and add field tags to the document structure for each object.

Using the PDF Structure palette, you can see which objects are associated with the structure tags in the form. When you click a content tag in the PDF Structure palette, the corresponding object in the form is highlighted in the Design View tab. For example, if you click a paragraph content tag <P> displayed in the PDF Structure palette, the corresponding content, usually text, is highlighted in the form.

After you import a structured PDF document as artwork, you can use the commands on the PDF Structure palette menu (and context menu) to add field tags to and remove them from the document structure, navigate the structure, move fields up and down in the structure, and show the content tags within the structure.

NOTE: Keep in mind that you can only edit the structure for fields in Designer. If you want to edit the structure tags and content structure in a form, you must do so in Acrobat.

Replacing PDF artwork affects document structure

Keep in mind that when you replace the background artwork in a structured PDF document, the current structure is replaced with the structure in the new background artwork. All content tags for the old artwork are replaced with the content tags for the new artwork. The form and field tags remain and are merged with the new content tags. Any content in the field objects is preserved and merged with the new content structure. You may need to move some of the form and field tags if the merged tags are not in the intended locations.

Note that any form fields that were added to the replacement PDF artwork in Acrobat are not converted to the corresponding editable library objects in Designer.

If the replacement PDF artwork is not structured, the structure of the current PDF artwork is removed and the resulting PDF form no longer has structure.

To add a field tag to the document structure

You can select a field object in a form and then add a field tag under a selected form tag in the document structure. You can add one field tag under each form tag.

You can also create a field tag by dragging a field in the form onto a structure tag in the PDF Structure palette.

1) In the Design View tab, select the field in the form to add to the document structure.
2) In the PDF Structure palette, right-click the form tag under which to add the field tag and select Add Field to Structure.

To remove a field tag from the document structure

1) In the PDF Structure palette, right-click the field tag to delete and select Remove Field from Structure.

To move a field tag up or down in the document structure

1) In the PDF Structure palette, right-click field tag to move and select Move Field Up or Move Field Down as needed.

To find untagged fields in the document structure

1) Right-click in the PDF Structure palette and select Find Untagged Fields.

To show the context tags in the document structure

1) Right-click in the PDF Structure palette and select Show Content tags.

RELATED LINKS:

Import documents

About importing structured PDF documents as artwork

PDF Structure palette menu

6.9. Importing Word files

You can import a Microsoft Word XP or Microsoft Word 2003 file into Designer. The Word file can be a document (DOC), a template (DOT), or a rich text format (RTF) file.

Word objects, including paragraphs, tables, images, line art, form fields, headers, and footers, are converted into Designer XML objects, and all rich text formatting is preserved.

NOTE: If the Word file contains contiguous underlined tab characters (where underscore is used as the tab character), Designer attempts to convert the information into a text field.

Designer displays the file import options each time you import a Word file.

1) Select File > Open.
2) From the Files Of Type list, select Microsoft Word Document (DOC, DOT, RTF).
3) Navigate to the file that you want to open, select the file, and click Open.
4) In the File Import Options dialog box, type a password if it is needed to open or unprotect the file.
5) To substitute any missing fonts and suppress all font substitution error messages, select Ignore Missing Fonts.

6) To convert and embed any images into the form design, select Convert Images.
7) Select a logging option from the Generate Log File list.
8) Click OK.

NOTE: Because Designer accesses the Windows clipboard to convert data, the clipboard cannot be accessed by other applications while Designer is importing a Word file.

How Designer converts Word objects and settings

When Designer converts a Word file, it preserves the layout. However, you may need to do some manual repositioning of objects afterward to make the form look as good as possible.

This table identifies the key objects in a Word file and indicates how Designer treats them when they are imported.

Item	Converted	Not Converted
File properties	Built-in and custom properties are converted to form properties.	
Paragraphs	Content (for example, text), including rich text formatting Font effects such as underline, strikethrough, superscript, and subscript Borders and shading applied to paragraphs	Hyperlinks Color underline is converted to black underline. All Caps font effect Font color of list numbers if it is different from the text color Shading for Empty paragraphs Borders applied to text
Tables	Tables are converted to subforms that position content, and table cells are converted to child subforms.	Some border styles are not supported.
Shapes	Text boxes Images Lines, rectangles, and circles are rasterized. Auto shapes OLE objects Frames ActiveX® controls are converted to images.	Arrowheads are converted to straight lines. Drawn objects are converted to images. Line slope Drawing Shape Rotation other than multiples of 90 are not supported. Blank lines in frames ActiveX control functionality
Inline shapes	Images and OLE objects are treated like characters and are positioned as if they were a character in a line of text. ActiveX controls are converted to images.	ActiveX control functionality

Item	Converted	Not Converted
Form Fields	Checkbox fields are converted to check boxes. The default state is carried over. Drop-down fields are converted to drop-down lists. All list items are carried over. Text input fields are converted to text fields and the default value is retained. HelpText and StatusText contents are converted to tooltips.	Formatting of Textinput values Web Tools objects
Headers and Footers	If all headers and footers are the same, they are created on the master page and are positioned according to the Word document's header and footer layout settings. If any headers and footers are different, they are created on individual pages.	Embedded field codes such as date and time
Page Setup settings	The Margins settings are used to specify the layout of the default subform for the page. Multiple pages are converted to single individual pages. The Paper Size and Orientation settings determine the basic page layout properties.	Gutter settings
Scripts	Any scripting associated with a form field is converted into comments in the Designer XML source code.	
Page borders	If specified, this attribute is converted to a border on the master page.	
Line numbers		Not converted
Comments		Not converted
Web page attributes		Not converted

About reducing Word conversion problems

Microsoft Word XP or Microsoft Word 2003 must be installed on the same computer as Designer.

If you are importing a Word XP file that is configured to be sent as an email message, ensure you have installed Word XP Service Pack 1. Otherwise, Word XP will crash after the import process has completed.

To obtain the best results, the input file should contain no more than ten pages.

6.10. Copying spreadsheet data from Microsoft Excel

You can copy spreadsheet data from Microsoft Excel 2002 or later and paste the cells into a form in Designer. Copying and pasting data from an earlier version of Excel may produce unexpected results.

Designer creates a table object in the form with the same dimensions, number of cells, and data content as the selection in the spreadsheet. Designer also preserves some of the original formatting of the spreadsheet cells, including the text, fonts, colors, background colors, alignment, and borders. If the selected cells in the spreadsheet contain column headers, Designer maintains the column header formatting in a header row at the top of the table.

Keep in mind that Designer does not maintain any calculations, scripts, images, and embedded objects, such as graphs that are in the Excel spreadsheet.

1) In Excel, copy the spreadsheet cells you want, and keep Excel open.
2) In Designer, select Edit > Paste.
3) Select Paste Cells As a Table and click OK.

RELATED LINKS:

Creating a form from the content of a Microsoft Excel spreadsheet

Spreadsheet Paste Options dialog box

6.11. Creating a form from the content of a Microsoft Excel spreadsheet

This feature is available only in the stand-alone version of Designer.

You can use a Microsoft Excel spreadsheet as the basis to create a new PDF form, which you can distribute to recipients by email to collect the same type of data as in the spreadsheet.

Using the New Form Assistant, you can create a PDF form with interactive text fields based on the column headers in the Excel spreadsheet. The New Form Assistant creates one text field object for each column in the spreadsheet. Text field objects are positioned in the form from left to right and wrapped onto additional lines if needed. The caption and binding name of each text field object corresponds to the column header text in the spreadsheet.

If you include an email submit button object in the form, you can distribute the form to recipients to fill and return. When you select the Distribute Form command from the File menu, the form closes in Designer and the Acrobat Distribute wizard appears to guide you through the distribution process. As you receive the filled forms from the recipients, you can organize the collected data into data sets in Acrobat. After you receive multiple sets of form data, you can reorganize the form data into a spreadsheet.

Keep in mind that Designer does not maintain any calculations, scripts, images, and embedded objects such as graphs that are in the spreadsheet document.

1) In Excel, open the spreadsheet you want to use to create a new PDF form.
2) In Designer, select File > New.
3) Select Based on a Spreadsheet, and click Next.
4) Follow the on-screen instructions in the New Form Assistant. You must leave Excel running to complete the process in the New Form Assistant.

RELATED LINKS:

Copying spreadsheet data from Microsoft Excel

Spreadsheet Paste Options dialog box

6.12. Importing InfoPath files

You can import Microsoft InfoPath files (XSN) into Designer. During the conversion process, Designer extracts these components from the input file and uses them to create Designer XML elements:

- An XML file describing the contents of the package
- An XML file describing the InfoPath form
- XSL files describing each view
- A schema file for the InfoPath form
- A script file, which is saved as comments in the Designer XML source code

Designer displays the file import options each time you import an InfoPath file.

1) Select File > Open.
2) From the Files Of Type list, select InfoPath (XSN).
3) Navigate to the file that you want to open, select the file, and click Open.
4) Do one of the following actions:
   - If you want to be prompted for a view, select the Prompt To Select View During Conversion option. If the InfoPath file has only one view, a prompt will not display even if you select this option.
   - If you know the name of the view that you want to import, type the name of the view in the View box. To convert the default view, leave the box blank.
5) Do one of the following actions:
   - Select the Import Print Settings option to import headers and footers, page orientation, and page margins of the InfoPath form.
   - If you know the page size of the InfoPath form, select one from the list or select Default.
   - If you know the page orientation of the InfoPath form, select one from the list.

   NOTE: If you select the Import Print Settings option, the page size and orientation are taken from the InfoPath form and you do not have to select the page size and orientation.
6) (Optional) Select an option from the Generate A Log File list.

7) (Optional) To convert and embed any images into the form design, select Embed Images In XDP.

8) Click OK.

NOTE: *Designer uses Microsoft Internet Explorer 5.0 or later to transfer data while importing an InfoPath file. If you do not have Internet Explorer installed, you cannot import an InfoPath file into Designer.*

How Designer converts Microsoft InfoPath objects

Designer preserves the layout when converting a Microsoft InfoPath file. However, you may need to do some manual repositioning of objects afterward to make the form look as good as possible.

The table identifies the key objects in an InfoPath form and indicates how Designer treats them when they are imported.

Item	Converted	Not Converted
Controls	Text boxes, including rich text boxes, are converted to text fields. Drop-down list boxes are converted to drop-down list boxes. Values are included only if the control is filled manually. List boxes are converted to list boxes. Date picker controls are converted to Date/Time fields unless the data type for the date picker in InfoPath is set to "Text (String)". In this case, it is converted to a Text field. Check boxes are converted to check boxes. Option buttons are converted to radio buttons. Buttons are converted to buttons.	Validation rules Scrolling Images embedded in rich text boxes A group of option buttons may not be converted to a single radio button group. Submit actions for buttons are not handled Padding properties of check boxes and radio buttons Attachments Custom controls
Expression boxes		Not converted
Lists	Numbered lists are converted to text. For bulleted lists, the text only is converted to text.	Bullet symbols
Pictures	Pictures are converted to images.	Ink pictures
Schema	If the InfoPath form has been designed based on a schema, the schema is imported and binding is done automatically in Designer.	
Scripts		Not converted

Item	Converted	Not Converted
Scrolling regions	Scrolling regions are converted to subforms.	
Sections	Sections are converted to subforms. Repeating sections are converted to repeating subforms in a form. Add and delete buttons are provided to add or delete the instance of the section when filling the form.	Include/exclude properties Instructional text Only one instance of any optional or repeating sections is converted
Tables	Tables are converted to tables. Repeating tables are converted to repeating tables in a form. Add and delete buttons are provided to add or delete the instance of the section when filling the form.	Some border styles are not supported. Only the visible rows in a repeating table are converted. Rows that have been merged vertically are converted to subforms in a positioned layout.
Text	Text is converted to text. Any highlight color is converted to background color.	Hyperlinks

Differences between Designer and InfoPath objects

Designer and InfoPath do not support the same features and object properties. The most obvious differences include these:

- Images cannot be placed in Designer text fields.
- Designer does not support editable sequential numbered or bulleted lists.

6.13. Importing Adobe Output Designer Form files

To import Output Designer files (IFD) into Designer, you must install Output Designer 5.5 or later on the same computer as Designer.

Designer displays the File Import Options dialog box each time you import an Adobe Output Designer Form file.

1) Select File > Open.
2) In the Files Of Type list, select Adobe Output Designer Form (IFD).
3) Select the file to open, and click Open.
4) In the File Import Options dialog box, select options as needed:
   – To generate a temporary log file, in the Generate a Trace Log of the Conversion Process list select one of the following:
     - **\TEMP\ConvertIFD.log** Creates a log file in your system's temporary folder that records information about the conversion process.

- **\TEMP\inputfile.log** Generates a log file in your system's temporary folder using the same name as the input file. Generates a log file for each imported file.

– To use the Output Designer configuration file to generate PDF form file, select Use Output Designer PDF Configuration. Uses an Output Designer configuration file (pdf.ics) and a compiled version of the configuration file (pdf.icf) to generate a PDF form file.

– To synchronize field objects with matching names to use global data binding, select sync Like Named Fields As Global. Applies a global value to all field objects in the form with the same name.

– To remove all un-named subforms from the hierarchy, select Import Fields Only.

5) Click OK.

RELATED LINKS:

Adobe Output Designer form (File Import Options dialog box)

Import documents

6.14. Importing XForms Model files

You can import XForms Model files into Designer. Each time you import an XForms Model file, the File Import Options dialog box appears.

1) Select File > Open.
2) In the Files Of Type list, select XForms Model.
3) Navigate to the file to open, select the file, and click Open.
4) (Optional) To embed a schema within the output XDP file if a schema is set for the XForms document, select Embed Schemas Inline.
5) (Optional) In the Set Working Directory box, type the path to the folder where you want Designer to check for relative file path locations, such as instance data XML files or XML schema files, which might be included in the XForms document. If you do not specify this folder, the default working directory will be the same location as the XForm that is being imported. This directory is not where the log files or temporary instance data files are created.
6) In the Generate log file list, select one of the following options:
   – If you do not want to generate a log file to capture conversion output messages, select Do Not Log.
   – To generate a log file in the Temp folder located in Documents and Settings\[*username*]\Local Settings\Temp (Windows XP) or Users\[*username*]\AppData\Local\Temp (Windows Vista), select \TEMP\ConvertXF.log. Each log file created is prefixed with ConvertXF and appended with a unique number (for example, ConvertXF38512.log).
7) Click OK.

NOTE: Designer forms do not support an equivalent of the `xforms-valid` event. Any scripting associated with the `xforms-valid` event is not imported.

RELATED LINKS:

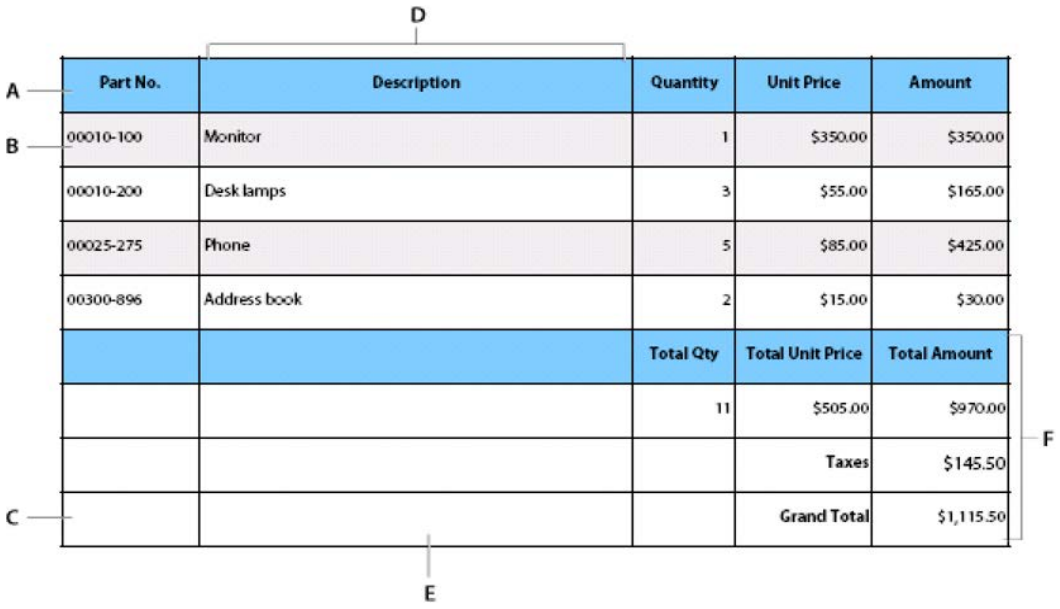
XForms Model (File Import Options dialog box)

Import documents

7. Using tables

7.1. About tables

A table is made up of rows and columns of cells that you can fill with form fields or merge with data. This example shows what the various parts of a table are called.



A.

Header Row

B.

Row

C.

Footer Row

D.

Column

E.

Cell

F.

Section

Types of tables

You can create two types of tables in Designer form designs:

- Tables that have a fixed number of rows and columns. For example, this table is a simple four-column, four-row table with a header and footer row.

Header	Header	Header	Header
Footer	Footer	Footer	Footer

- Tables in which the number of columns are fixed, but the number of rows will change depending on how much information is in the data source.

  A. This is how the table looks in the Layout Editor **B.***This is how the table looks in the Preview PDF tab when merged with data from a data source*

  Or, interactive tables can grow or shrink when a user clicks a button to add or delete a row.

  A. Buttons that add a row or delete a row **B.***When the user clicks the Add Row button twice, Designer adds two rows.*

Ways to create tables

Here are some of the common methods for creating tables.

- Create an empty table that has a fixed number of rows and columns. See *To create a simple table*.
- Create an empty table that has a fixed number of rows and columns. See *To create a simple table*.
- Create a table whose number of rows changes to accommodate the amount of data that displays. See *To create a table from a data source*.
- Create an empty table in which the number of rows changes to accommodate the amount of data. See *To create a table using the Table Assistant*.
- Create a table from existing objects. See *To create a table from existing objects*.
- Create a table nested within a table. See *To create a table within a table*.

- Import a table from another application. See *To import a table from Microsoft Word*.

7.2. To create a simple table

You can create a simple table with a range of columns and rows with or without header or footer rows. For example, here is a four-column, four-row table with a header and footer row.

1) In the Object Library palette, click the Standard category and select the Table object.
2) Click where you want the table to appear.

   NOTE: *If you selected Don't Show This Again in the Insert Table dialog box, when you select the Table object in the Object Library palette, Designer automatically inserts a table with the same number of columns and rows that you inserted the last time you used the Insert Table dialog box.*
3) In the Insert Table dialog box, enter the number of columns and rows.

   You can enter a maximum of 20 columns and 50 rows. You can add more columns and rows after the table is created by using the Insert commands in the Table menu.
4) (Optional) To add a header row, select Include Header Row In Table.

   NOTE: Select Repeat Row for Each Data Item and Include Header Row In Subsequent Pages table properties to ensure that the header row displays on each page for the table.
5) (Optional) To add a footer row, select Include Footer Row In Table.
6) Click OK.

RELATED LINKS:

To repeat a header or footer row on subsequent pages

7.3. To create a table using the Table Assistant

The Table Assistant simplifies and streamlines the process of creating a table by letting you easily define some of the basic elements in a table before you place the table on a form. For example, the Table Assistant dialog box provides options for adding body rows and columns, adding header and footer rows, adding sections, and applying shading to rows.

You can use the Table Assistant to create tables that have a fixed number of rows and columns, or tables that have a fixed number of columns but the number of rows changes depending on how much information is in the data source.

As you select options, you can see how the table will look in the Preview area of the Table Assistant. After you create a table using the Table Assistant, you can modify it using the commands on the Table menu and the various options in the Layout, Border, Object, and Accessibility palettes.

TIP: *By default, the Insert Table dialog box appears when you select Insert > Standard > Table or when you drag the Table object onto the Layout Editor. However, if you previously selected the Don't Show This Again option in the Insert Table dialog box, you can restore the default behavior by selecting Tools > Options > Wizards and Tips and selecting the Show When Creating Tables option.*

1) Select Table > Insert Table, and select Create Table Using Assistant.
2) In the Table Assistant, specify the body layout, and then click Next:
   – To add a specific number of rows and columns, select Body Has Fixed Dimensions, and then select values in the Number of Columns and Number of Rows lists.
   – For a table in which the number of body rows in the table matches the number of rows in the data, select Body Rows Vary Depending on Data, and then select a value from the Number of Columns list.
3) Specify whether you want to include a header row, and then click Next:
   – To have no header row, select No Header Row.
   – To add a header row, select Has Header Row, and then select Repeat Header Row on Each Page, if needed. The Repeat Header Row on Each Page option is available only when you select the Body Rows Vary Depending on Data option in the Body Layout page.
4) Specify whether you want to include a footer row, and then click Next:
   – To have no footer row, select No Footer Row.
   – To add a footer row, select Has Footer Row, and then select Repeat Footer Row on Each Page, if needed. The Repeat Footer Row on Each Page option is available only when you select the Body Rows Vary Depending on Data option in the Body Layout page.
5) Specify the sections to include, and then click Next:
   – To include body rows with no sections, select Has Body Rows and No Sections.
   – To include body rows grouped into sections, select Has Sections of Body Rows and do the following tasks:
   – Use the Add, Delete, Up, and Down buttons above the Sections list to add, remove, and order sections as needed.
   – To rename a section shown in the Sections list, double-click the section and rename it in the Table Section dialog box.
   – To apply options to a section listed in the Sections list, select the section and, under Section Options, select options as needed. Note that when you select Section Is Optional, the minimum occurrence for the section is set to 0. This means that if no data exists for the section, the section will not be shown.
6) Specify the row shading and then click Finish:
   – Select Alternating Row Colors.
   – In the First list, select the number of initial rows to shade, and then select a color.
   – In the Next list, select the number of subsequent rows to shade, and then select a color.

RELATED LINKS:

To repeat a header or footer row on subsequent pages

7.4. To create a table from existing objects

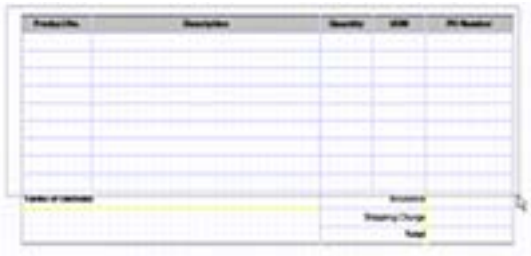
If you have an existing form that includes field objects formatted as a table, you can easily convert them to a table. For example, the Bill of Lading template that comes with Designer includes fields formatted like a table.

1) Open a form that contains fields formatted like a table.

   For example, start a new form based on the Bill of Lading template. Select File > New and select Based On A Template. Click Next and then select the Bill of Lading template. Click Next and finish filling in the New Form Assistant.

2) Drag the objects that you want to convert to a table.

   For example, select the header row and body rows but stop above the Terms of Delivery and Insurance section.



3) Select Table > Convert to Table.

   NOTE: *You can convert any group of fields to a table. For example, in this same template, you can convert the fields under Shipped To and Shipper to a table.*

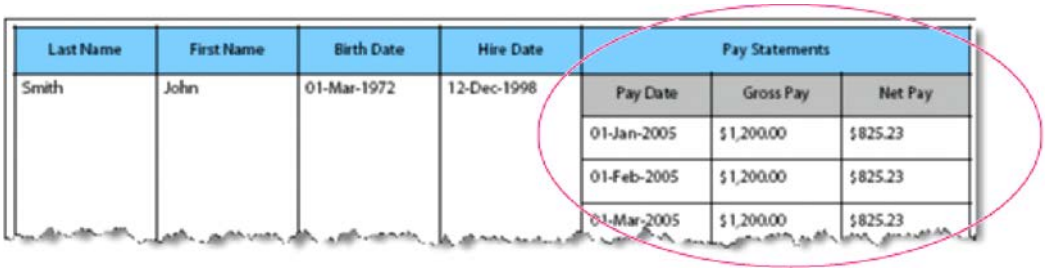
7.5. To create a table within a table

You can create a table within a table that is independent from the main table. For example, the nested table can have a different number of columns and rows than the main table. There is no limit to the number of levels that you can nest tables.

If you want a table to automatically flow to the next page, you must ensure the following conditions:

- The table must be in a subform that is set to Flowed in the Subform tab of the Object palette.
- The Allow Page Breaks Within Content option is selected in the Table tab.
- The table is not in a group.

Table within a table



Last Name	First Name	Birth Date	Hire Date	Pay Statements		
Smith	John	01-Mar-1972	12-Dec-1998	Pay Date	Gross Pay	Net Pay
				01-Jan-2005	$1,200.00	$825.23
				01-Feb-2005	$1,200.00	$825.23
				01-Mar-2005	$1,200.00	$825.23

1) Create a table. See *To create a simple table*.
2) In the Object Library palette, click the Standard category and drag the Table object into a cell in the main table.

   NOTE: *If you selected Don't Show This Again in the Insert Table dialog box, when you select the Table object from the Object Library palette, Designer automatically inserts a table with the same number of columns and rows that you inserted the last time you used the Insert Table dialog box.*
3) In the Insert Table dialog box, enter the number of columns and rows.

   You can enter a maximum of 20 columns and 50 rows. You can add more columns and rows after the table is created by using the Insert commands on the Table menu.
4) (Optional) To add a header row, select Include Header Row In Table.
5) (Optional) To add a footer row, select the Include Footer Row In Table.
6) Click OK.

   NOTE: *If you have an existing table, you can select the table and drag it to the cell where you want it to appear.*

7.6. To import a table from Microsoft Word

You can import a table from Microsoft Word into Designer.

1) In Designer, select File > Open.
2) In the Files of Type list, select Microsoft Word Document.
3) Select the Microsoft Word document that contains the table and click Open.

   Designer displays the New Form Assistant and Import A Microsoft Word Document is already selected.

4) Click Next and finish answering the questions when prompted in the New Form Assistant.
5) In the File Import Options dialog box, provide the required information and click OK.
   Designer imports the contents of the document, including the table.

RELATED LINKS:
To change a cell to another object type

7.7. To create a table using subforms

You can create a table using subforms. Subforms can contain a variety of objects including buttons, text fields, and other subforms. The flow direction of the subform indicates how the contained objects will be arranged. Typically, you will create tables by using the Table object in the Object Library palette. If you need to create a complicated table where the columns do not line up, you can use subforms.

1) Start the table:
   – In the Object Library palette, click the Standard category and drag a Subform object onto the form.
   – Resize the subform to match the required width of the table.
   – Expand the height of the subform so that you can add one or more child subforms.
   – Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `TableParent`.
   – In the Accessibility palette, select Table from the Subform Role list.
2) Create a table header:
   – Drag another Subform object into the TableParent subform.
   – Set the width of the header subform to match the width of its parent subform and set the height of the subform to match the required height of the header row.
   – Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `TableHeader`.
   – In the Accessibility palette, select Header from the Subform Role list.
3) Add text for the column headers:
   – Drag a Text object into the TableHeader subform.
   – Double-click the text in the Text object and type a name for the column.
   – Repeat as many times as required to add additional headers to the table.
4) Create a body row that will act as the repeating row in the table:
   – Drag another Subform object into the TableParent subform.
   – Set the width of the body row subform to match the width of its parent subform and set the height of the body row subform to match the required height of the body row.
   – Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `TableBodyRow`.
   – Add the same number of fields to go under the headers that you added in step 3 to display the data values in the table. Set the size of the fields.

- In the Layout palette, select None from the Caption list.
- In the Accessibility palette, select Body Row from the Subform Role list.

5) Select the TableParent subform, click the Subform tab of the Object palette, and select Flowed from the Type list.
6) In the Object palette, click the Binding tab and select Repeat Subform For Each Data Item.
7) (Optional) Consider enhancing the table as follows:
   - Add borders and shading for table elements by using the Border palette.
   - Specify overflow leaders and trailers for the rows of the table.

To create a table containing variable-width cells

1) Create two or more tables. Each table should have one body row. The first table should have a header row, and the other tables should have only body rows.
2) Arrange the tables one below the other.
3) Wrap all these tables in a parent subform.
4) Adjust the width of each cell as required.
5) Set the parent subform to Flowed.

RELATED LINKS:

About subforms

About overflow leaders and trailers

Making complex tables accessible

Border properties in the Border palette

Accessibility properties in the Accessibility palette

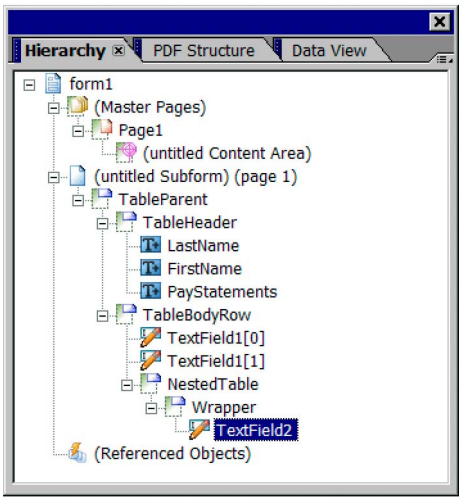
7.8. To create a nested table using subforms

After you create a table using subforms, you can insert a table into a table cell. For example, you can show two independent tables side by side (in separate cells) or nest tabular material.

Before you insert a table in to a cell, you must wrap the cell in a subform.

1) Start the nested table:
   - Select the cell (in the table that was created using subforms) where the nested subform will go.
   - Select Insert > Wrap In Subform.
   - Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `Wrapper`.
2) Create the nested table.
   - Right-click the cell again and select Wrap In Subform, or drag the Subform object from the Object Library palette to the cell.

- Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `NestedTable`.
- In the Accessibility palette, select Table from the Subform Role list.



3) Right-click the cell (TextField1 in the example) and select Delete.
4) Create a header row for the nested table:
   - Drag another Subform object into the NestedTable subform.
   - Set the width and height of the header row.
   - Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `NestedRow1`.
   - In the Accessibility palette, select Table Header from the Subform Role list.

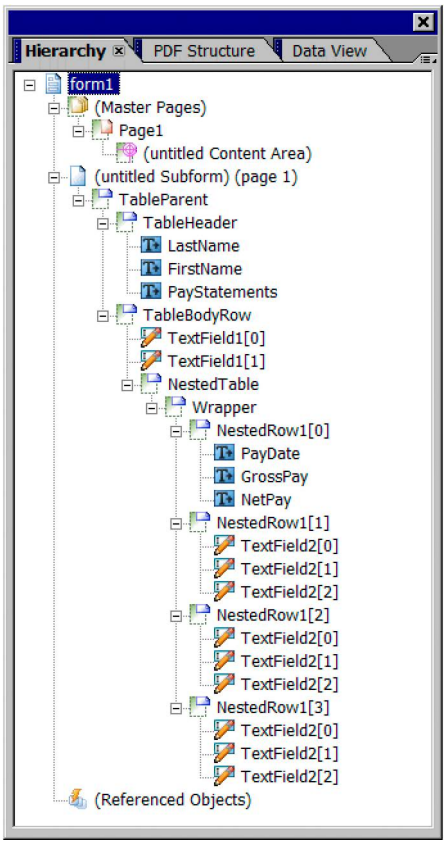
   NOTE: *A subform that is inserted into a parent subform that is set to Flowed may not appear in the correct position in the Hierarchy palette.*
5) Add text for the column headers:
   - In the Object Library palette, click the Standard category and drag two Text objects into NestedRow1.
   - Select one of the text objects and type a name for the text object. For example, type `Description`.
   - Select the second text object and type a name for the text object. For example, type `Cost`.
   - Select the two Text objects in NestedRow1 and select Layout > Group.
6) Create a body row for the nested table:
   - Drag another Subform object into the NestedTable subform.
   - Set the width and height of the body row.
   - Type a name for the subform in the Name box in the Binding tab of the Object palette. For example, type `NestedRow2`.
   - In the Accessibility palette, select Body Row from the Subform Role list.
   - In the Object Library palette, click the Standard category and drag three field objects into NestedRow2 (such as Text Fields).

7) Set the parts of the table to Flowed:

– Select the NestedTable subform and, click the Subform tab of the Object palette, and select Flowed from the Type list.

– Select the subform that contains the NestedTable subform called (untitled Subform) (page 1), click the Subform tab of the Object palette, and select Flowed from the Type list.

The Hierarchy palette could look like this illustration.



The form could look like this in the Preview PDF tab.

Last Name	First Name	Pay Statements		
Smith	John	Pay Date	Gross Pay	Net Pay
		01-Jan-2005	01-Feb-2005	01-Mar-2005
		01-Mar-2005	01-Feb-2005	01-Jan-2005
		01-Jan-2005	01-Mar-2005	01-Feb-2005

RELATED LINKS:

About subforms

About overflow leaders and trailers

Border properties in the Border palette

Accessibility properties in the Accessibility palette

7.9. To create a table that grows by using the Subform Instance Controls

You can create a table that grows when users click a button by using the Subform Instance Controls. These controls let users add and remove rows and move rows up and down in a table.

A.

Adds and removes rows

B.

Moves rows up and down

You must create the table before you insert the controls. You also must already have the predefined set of custom objects in the SAP environment.

To create a table that grows by using the Subform Instance Controls

1) Select Table > Insert Table.
2) In the Insert Table dialog box, select Create Table Using Assistant and click OK.
3) In the Body Layout panel of the Table Assistant, select Body Rows Vary Depending On Data.
   This creates a table that is dynamic so that it grows based on the data sent to it.
4) Enter the number of columns and click Next.
5) (Optional) Select Has Header Row and click Next.
6) (Optional) Select Has Footer Row and click Next.
7) (Optional) Select Has Body Rows And No Sections and click Next.
8) (Optional) Select Alternate Row Colors and click Finish.
9) Select Edit > Form Properties and click the Preview tab.
10) In the Preview Adobe XML Form As list, select Dynamic XML Form and then click OK.
    Now you must add the controls that will add and delete rows dynamically.

To add the Subform Instance Control buttons

1) Select the first cell of the body row and, in the Object palette, click the Cell tab and select Subform from the Type list.
   This cell needs to be a subform so that it can contain the buttons.

TIP: *You can also drag the Subform object from the Object Library palette into a cell in the table.*

2) In the Object Library palette, click the category that contains the custom objects for the SAP environment.

   You must have already set up a category that contains these custom objects.

3) Drag the Subform Instance Controls: Insert Remove Move object into a cell in the body row. For example, place them in the first cell of the body row.

4) Select the first row.

5) In the Binding tab of the Object palette, ensure that Repeat Row For Each Data Item is selected.

6) View the form in the Preview PDF tab.

   For example, this is what you would see if you formatted the table.

Here is what it would look like after a user adds rows and data.

RELATED LINKS:

About custom objects for the SAP environment

7.10. To create a table that grows using the Button object

You can create a table that adds a row when a user clicks an Add Row button. You can also include a Delete Row button. You must create the table, add the buttons, and then set the properties that will make the table grow.

A.

Buttons that add a row or delete a row

B.

When the user clicks the Add Row button twice, Designer adds two rows.

To create a table that grows using the Button object

1) Select Table > Insert Table.
2) In the Insert Table dialog box, select Create Table Using Assistant and click OK.
3) In the Body Layout panel of the Table Assistant, select Body Rows Vary Depending On Data.
   This creates a table that adds or removes rows, depending on the data sent to it.
4) Enter the number of columns and click Next.
5) (Optional) Select Has Header Row and click Next.
6) (Optional) Select Has Footer Row and click Next.
7) (Optional) Select Has Body Rows And No Sections and click Next.
8) (Optional) Select Alternate Row Colors and click Finish.
9) Save the form as an Adobe Dynamic XML Form (*pdf).

Now, you must add the buttons that will be used to add and delete rows.

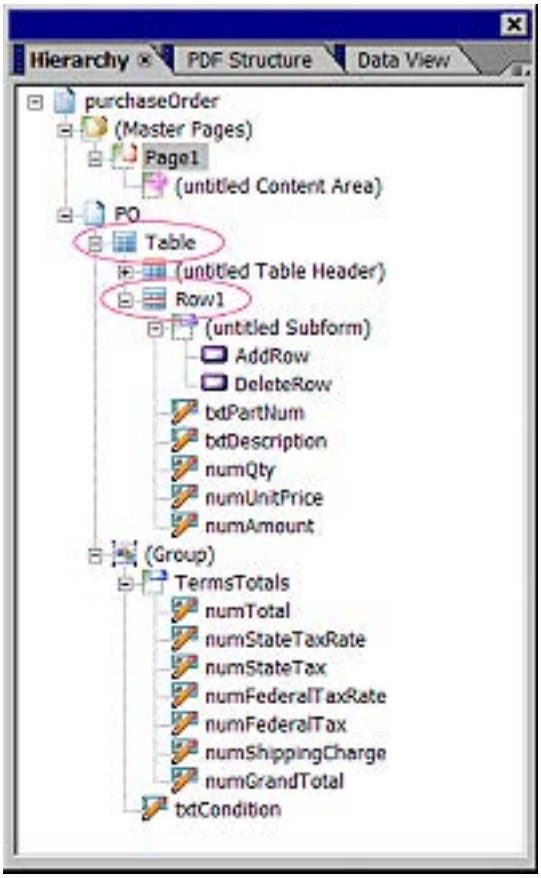
To add the buttons

1) Select the first cell of the body row and, in the Object palette, click the Cell tab and select Subform from the Type list.
   This cell needs to be a subform so that it can contain two buttons.
   TIP: *You can also drag the Subform object from the Object Library palette into a cell in the table.*
2) In the Object Library palette, click the Standard category and drag the Button object into a cell in the body row. For example, place it in the first cell of the body row.

3) In the Object palette, click the Field tab and type `Add Row` in the Caption box.
4) Repeat steps 2 and 3 to create a Delete Row button.
5) Select the Add Row button and, in the Script Editor, select Click from the Show list.
6) In the Script Editor, select JavaScript from the Language list.
7) Type the following script:

```
Table.Row1.instanceManager.addInstance(1);
```

To modify the script for your own form, you must understand how the script relates to the hierarchy. For example, in the hierarchy for the current form, both buttons are inside the same subform; therefore, you do not need to specify any objects beyond the Table level.



8) Select the Delete Row button and, in the Script Editor, select Click from the Show list.
9) In the Script Editor, select JavaScript from the Language list.
10) Type the following script:

```
Table.Row1.instanceManager.removeInstance(1);
```

TIP: *To reuse these buttons in another form, you can add them to the Custom category (or your own category) of the Object Library palette. Note that you may have to edit the script for the button if you use it in a different form.*

11) View the form in the Preview PDF tab.

For example, this is what you would see if you modified the Part table from the Purchase Order template.

7.11. Selecting, copying, moving, and navigating

To select a table, row, column, cell, or section

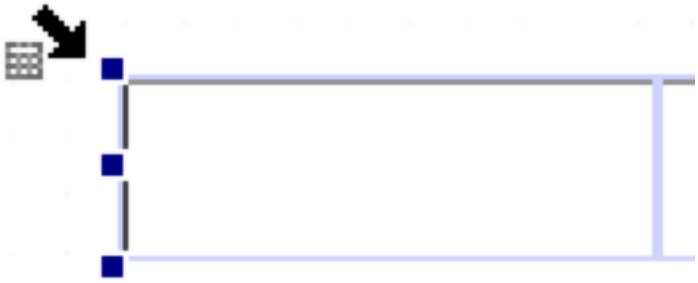
You can select rows, columns, cells, sections, or the entire table in different ways:

- Click in the table and use the Select commands in the Table menu.
- Drag the mouse pointer to select different parts of the table.
- Select objects in the Hierarchy palette.

You can also use one of the following methods.

To select the entire table

1) Click in the area on the upper left of the table, or click and drag to select the entire table.



To select the cells in a row

1) Click in the area to the left of the row until you see the following arrow.

To select a row

1) Click in the area to the left of the row until you see the following arrow.

To select the cells in a column

1) Click above the column.

To select a cell

1) Click the lower-right edge of the cell.

NOTE: *After a cell is selected, you can select more cells in the table by using the arrow keys to move within the table. Press Shift and then press the arrow key to select adjacent cells.*

To select multiple rows, columns, cells, or sections

1) Drag the mouse pointer across the rows, columns, cells, or sections.

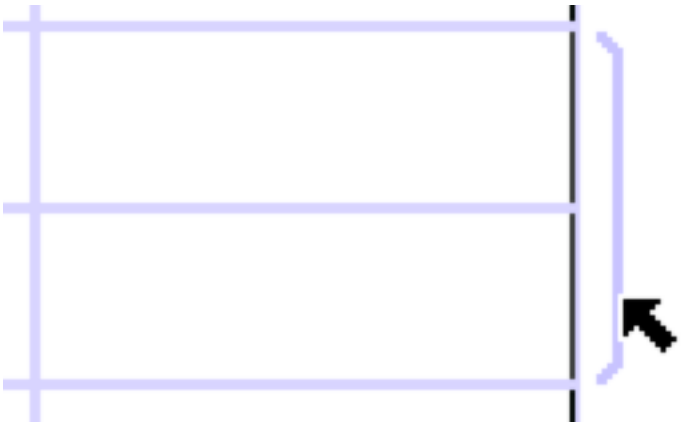
   To select multiple items that are not adjacent, click the first row, column, cell, or section you want, press Ctrl, and then click the next rows, columns, cells, or sections you want.

   To select adjacent cells, click the first cell you want, press Shift, and then use the arrow keys to select more cells in the table.

   NOTE: *The rows, columns, cells, or sections must be in the same table. A nested table is considered a separate table.*

To select a section

1) Click the right bracket that indicates a section.



To copy a table, row, column, cell, or section

1) Select the table, row, column, cell, or section you want to copy.
2) Do one of the following actions:
   – Press Ctrl and drag the copy to a new location.
   – Select Edit > Copy, click the where to place the object, and then select Edit > Paste.
   – Select Edit > Duplicate. This command does not work for single cells or cells in rows.

   TIP: *To make multiple copies that are positioned and aligned, use the Copy Multiple command in the Edit menu. This command does not work for single cells or cells in rows.*

To move a table

1) Select the table.
2) Click in the upper part or left side of the table to find the move cursor and drag it to the new location.

To go to a specific row

If the table is long, you can go to a specific row in a table, including a header, body, or footer row.

1) Select part of the table and select Table > Go to Row.
2) Select Header Row, Body Row, or Footer Row.

   You can also go to a specific header or footer row within sections.
3) Type the row number to go to and click OK.

7.12. Inserting and deleting

To insert a row, column, or section

1) Select part of the table.
2) Select Table > Insert and select an option.

   NOTE: *When you insert a section, Designer adds a header row for the section by default.*

To delete a table, row, column, or section

To delete a table

1) Select the table. See *To select a table, row, column, cell, or section*.
2) Select Table > Delete and click Table.

   TIP: *If the table is not a nested table, you can select Table > Delete to delete the table.*

To delete a row, column, or section

1) Select the rows, columns, or sections. See *To select a table, row, column, cell, or section*.
2) Select Table > Delete and click Row, Column, or Section.

7.13. Formatting a table

To add a table title

1) In the Object Library palette, click the Standard category and drag the Text object above the table.
2) Double-click the default text and type the title.
3) Set the font properties.

TIP: *You can also merge the cells in the first row of the table and type the title in the combined cell.*

Adding space

To add space around a table

You can add blank space around the edges of a table by using margins.

1) Select the table. See *To select a table, row, column, cell, or section*.
2) In the Layout palette, set the margins for Left, Right, Top, and Bottom.

To add space around cells

You can add space around cells by using margins.

1) Select the cell. See *To select a table, row, column, cell, or section*.
2) In the Layout palette, set the margins for Left, Right, Top, and Bottom.

To add space around rows

You can add space around rows by using margins. For example, in the following table, margins are set to 0.25 inch around the row.

Part No.	Description	Quantity	Unit Price
00010-100	Monitor	1	$350.00
00010-200	Desk lamps	3	$55.00
00025-275	Phone	5	$85.00
00300-896	Address book	2	$15.00

1) Select the row. See *To select a table, row, column, cell, or section*.
2) In the Layout palette, set the margins for Left, Right, Top, and Bottom.

Applying borders and shading to a table

Borders and shading can add interest and emphasis to a table. You can add borders to a table or an individual row, column, or cell. You can use shading to fill in the background of a table.

You can add a striped or alternating shading pattern to the rows. You can select the number of rows or columns to include in a stripe pattern and apply your color choices to the pattern.

To apply and remove borders and shading

1) Select the table. See *To select a table, row, column, cell, or section*.
2) To apply a border and shading, do one of the following actions:
   – To place borders only on particular sides, in the Border palette, select Edit Individually from the Edges list. Now you can set the border for each side.
   – To place the same border on all sides, select Edit Together from the Edges list.
3) Select an option from the list of line types and select a color from the color picker.
4) To apply shading, select a style from the Style list and select a color from the color picker.
5) To remove a border or shading do the following action:
   – In the Border palette, select Edit Together from the Edges list.
   – Select None from the list of line types.
6) To apply a striped or alternating shading to rows
   – In the Object palette, click the Row Shading tab and select Apply Alternating Row Shading.
   – Select the shading color for the first row or first set of rows.
   – Select the shading color for the next row or next set of rows.

   NOTE: *If you use the Border palette to set shading for a cell, it overrides the row shading set for the table.*
7) To remove the striped or alternating shading from rows
   – In the Object palette, click the Row Shading tab and deselect Apply Alternating Row Shading.

To remove the border from around the entire table

1) Select the table and, in the Border palette, select None from the list of line types.

To distribute rows and columns evenly

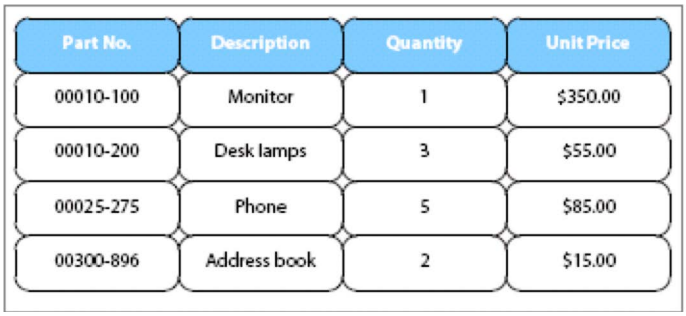
You can make multiple rows or columns the same size. All rows adjust to the height of the tallest selected row. All columns adjust to equal distances based on the width of the table. The table does not grow.

1) Select the rows or columns you want to make the same size. See *To select a table, row, column, cell, or section*.
2) Select Table > Distribute Rows Evenly or Distribute Columns Evenly.

NOTE: *If you used the Merge Cells command, the original columns still exist but are hidden (so that you can split the cells again if you want). To make the columns the same size after you have merged cells, delete the hidden columns first by using the Table > Delete > Column command.*

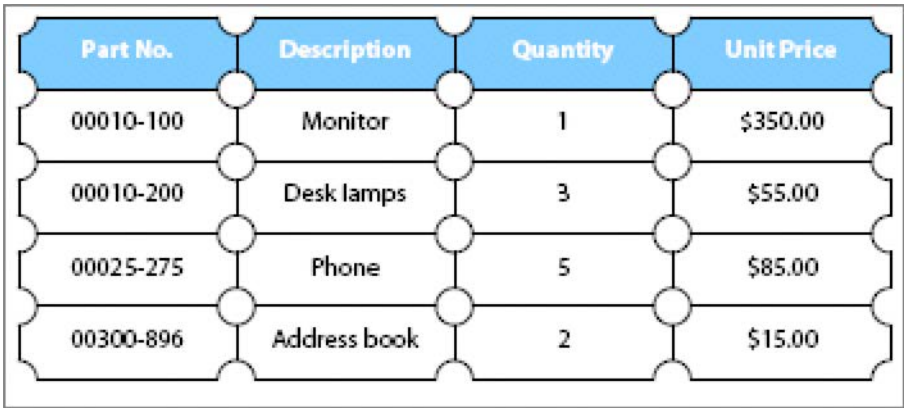
To format the corners of a table or cells

You can format a table or specific cells in a table to have rounded or notched corners.



Part No.	Description	Quantity	Unit Price
00010-100	Monitor	1	$350.00
00010-200	Desk lamps	3	$55.00
00025-275	Phone	5	$85.00
00300-896	Address book	2	$15.00

Part No.	Description	Quantity	Unit Price
00010-100	Monitor	1	$350.00
00010-200	Desk lamps	3	$55.00
00025-275	Phone	5	$85.00
00300-896	Address book	2	$15.00



1) Select the table or the cells. See *To select a table, row, column, cell, or section*.
2) In the Border palette, select one of the options for Corners.
3) In the Radius box, type a number. For example, the previous example uses .1 inch.

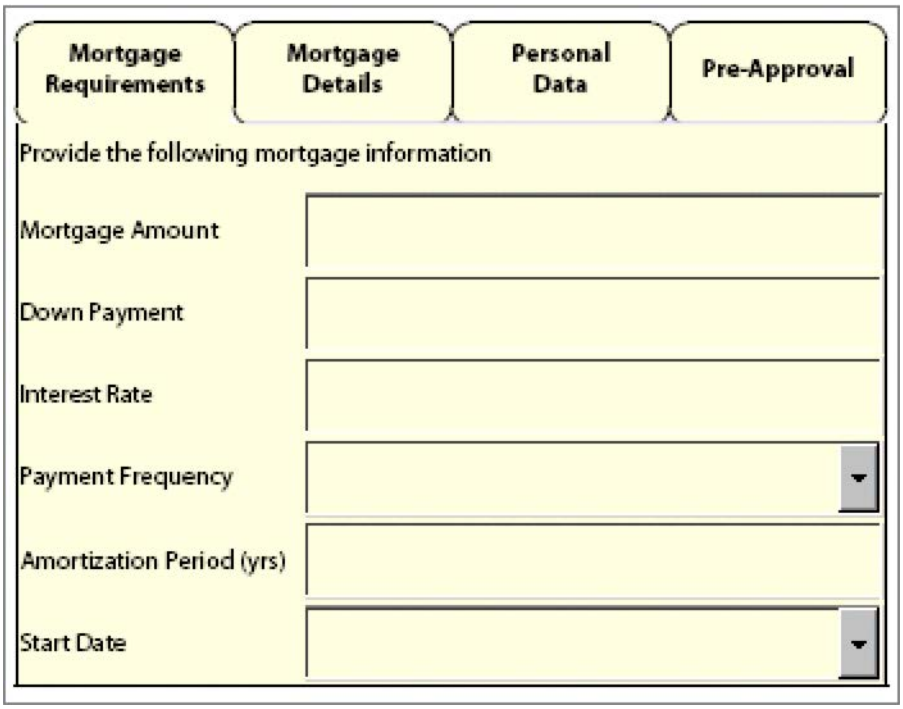
To show captions in rows

By default, captions for fields are not shown when a field is in a cell. For example, the following objects have captions, but they are hidden when the field is in a cell.

- Check Box

- Date/Time Field
- Decimal Field
- Signature Field
- Drop-down List
- Image Field
- List Box
- Numeric Field
- Password Field
- Text Field

You may want to show the caption in some tables. For example, in the following form, a table is used to present the information. Each row below the header row is one cell.



1) Select the cell. See *To select a table, row, column, cell, or section*.
2) In the Layout palette, select a position for the caption from the Position list.

7.14. To create a table style for the custom library

You can create your own table style to provide a consistent look to borders, shading, alignment, and fonts in tables. For example, you can format a table that looks like this one and save it in the custom library so that you can reuse it in another form.

Country	Q1	Q2	Q3	Total
Canada				
US				
Total				

1) Create a table.
2) Set the borders, shading, cell types, cell alignments, and fonts.
3) Ensure that the Custom category of the Object Library palette is open.
4) Select the table and drag it into the Custom category of the Object Library palette.
5) In the Add Library Object dialog box, type the name of the table style and click OK.

   You can type a description and select which tab group in the library that you want the object to appear in.

7.15. To create a calendar using a table

You can create a calendar by using a table.

1) Select View > Master Pages.
2) In the Object palette, click the Master Page tab and select Landscape.
3) Click the Design View tab.
4) In the Object Library palette, select Table > Insert Table.
5) In the Insert Table dialog box, type 7 for columns and 6 for rows.
6) Select Include Header Row In Table and click OK.
7) Move the table to the top of the page.
8) Resize the columns so that the table fits the width of the page.
9) Select Table > Distribute Columns Evenly.
10) Select the last row of the table. See *To select a table, row, column, cell, or section*.

11) In the Layout palette, type `1.1in` in the Height box.
12) Select the table. See *To select a table, row, column, cell, or section*.
13) Select Table > Distribute Rows Evenly.
14) Select the first row of the table. See *To select a table, row, column, cell, or section*.
15) Select Table > Merge Cells.
16) Type the month and set the font properties.
17) In the second row, type the days of the week and set the font properties.
18) In the remaining cells, type the numbers for the days of the week and set the font properties.

7.16. Aligning, resizing, and arranging tables

To align a table with other objects

1) Select the table. See *To select a table, row, column, cell, or section*.
2) Ctrl+click to select the other object that you want to align.
3) Select one of the following options.
   – To align the left edges of the objects, select Layout > Align > Left.
   – To align the right edges of the objects, select Layout > Align > Right.
   – To align the top edges of the objects, select Layout > Align > Top.
   – To align the bottom edges of the objects, select Layout > Align > Bottom.
   – To align the vertical centers of the objects, select Layout > Align > Vertical Center.
   – To align the horizontal centers of the objects, select Layout > Align > Horizontal Center.

To align the contents of a cell

You can change the horizontal and vertical alignment of the contents of a cell.

1) Select the cell. See *To select a table, row, column, cell, or section*.
2) In the Paragraph palette, select one of the following alignment options.
   – To left-align the contents of the cell, select Align Left.
   – To center-align the contents of the cell, select Align Center.
   – To right-align the contents of the cell, select Align Right.
   – To justify the contents of the cell, select Justify.

   NOTE: *Full justification is applied to all of the lines except the last line in a multiple-line cell (single line paragraphs cannot be justified).*
   – To align to the top of the cell, select Align Top.
   – To align to the middle of the cell, select Align Middle.

– To align to the bottom of the cell, select Align Bottom.

To make another object the same size as a table

You can make the size of an object the same size as a table in your form. For example, you can make a rectangle the same width as a table so that you can insert a dividing element or a border around a section of the form. Other objects can become the same size as the table, but a table cannot become the same size as other objects.

1) Select the object you want to make the same size as the table.
2) Ctrl+click or Shift+click to select the table. You must select the table last.
3) Select Layout > Make Same Size and select Width, Height, or Both.
   The object that you selected last is used to set the size of both objects.

To resize an entire table

You can resize an entire table. You can only resize on the bottom or right edge of the table. To resize the table outside of the content area, select the last column or last row and change its width or height in the Layout palette.

1) Select the table. See *To select a table, row, column, cell, or section*.
2) Place the pointer on the edge of the table until a double-headed arrow appears.
3) Drag the table boundary until the table is the size you want.

To resize rows and columns

You can resize rows and columns. You can only resize on the bottom, left, or right edge of the rows and columns. To resize the last column outside of content area, change the width in the Layout palette.

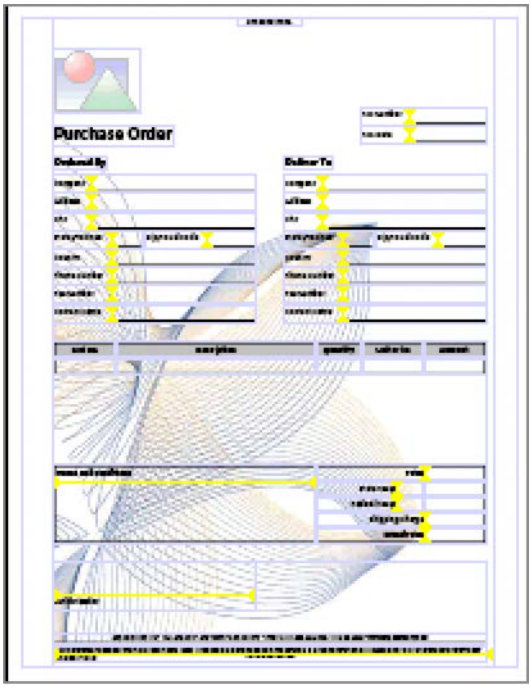
1) Place the pointer on the row or column borders until a double-headed arrow appears.
2) Drag the boundary until the row or column is the size you want.

To center a table

1) Select the table. See *To select a table, row, column, cell, or section*.
2) Select Layout > Center in Page and select either Horizontally or Vertically.

To arrange a table with other objects

You can control how objects overlap by putting them in front of or in back of other objects. For example, you can put a watermark image behind a table.



1) Select the table. See *To select a table, row, column, cell, or section*.
2) Choose one of these options:
   - To bring the table forward, select Layout > Bring Forward.
   - To bring the table to the front, select Layout > Bring To Front.
   - To send the table backward, select Layout > Send Backward.
   - To send the table to the back, select Layout > Send To Back.

7.17. Working with data in tables

To create a table from a data source

Some data from a data source could be shown in a table. When you connect to a data source, the Data View palette identifies any items that could be tables. Designer also shows which items will become rows in the table if it finds a repeating data group.

After you connect to a data source, you can drag items from the Data View palette onto the form design to quickly create fields that are bound to the data source. If you drag a table item from the Data View palette onto the form design, Designer creates a table and, when you select a data file, you see a table like this one at run time. The repeating data group in the data file has four records.

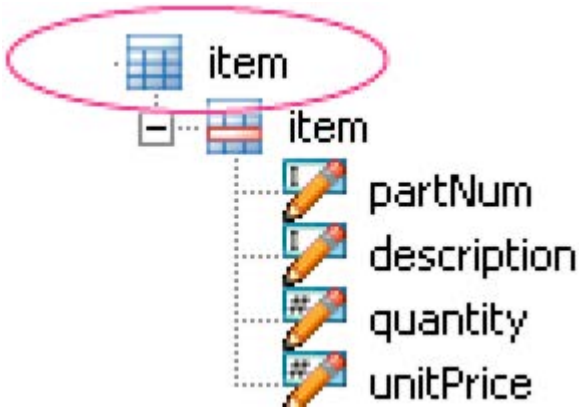
Part Num	Description	Quantity	Unit Price
00010-100	Monitor	1	350
00010-200	Desk lamps	3	55
00025-275	Phone	5	85
00300-896	Address book	2	15

1) Ensure you are connected to a data source. See *Connecting to a data source*.

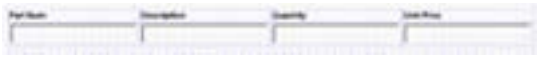
   For example, if you or your administrator installed the samples that come with Designer, locate and open the following schema: Purchase Order.xsd in the Data Connections dialog box.

2) In the Data View palette, locate a node that could be a table. For example, locate the following *item* node.

   Item node represents the table



3) Select the node and drag it onto your form design. For example, this is what you should see after dragging the item node onto your form.



   The Data View palette shows that the node is bound to an object.

4) To select a data file, select File > Form Properties and click the Preview tab.

   For example, if you or your administrator installed the samples that come with , locate and open the Purchase Order.xml data file.

5) View the form in the Preview PDF tab.

   Designer automatically creates a table that dynamically grows, depending on the amount of data in the data source.

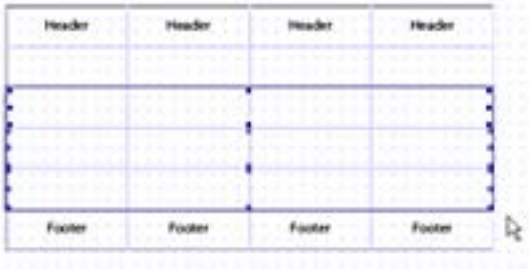
To set the rows in a table to adjust to accommodate data

If you have already created a table that has a fixed number of columns and rows, you can convert it to one that dynamically grows based on the amount of data from a data source. Converting the table involves setting the subform that contains the table to Flowed and setting the body row or rows to repeat.

1) Create a table. See *To create a simple table*.

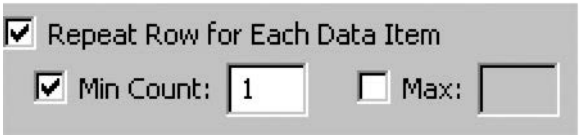
2) Select the rows below the first row.

In a table whose number of columns are fixed, but its number of rows will change depending on how much information is in the data source, only one row is needed. Therefore, you must delete all the rows except the first row in the table. You can keep the footer row.



3) Select Table > Delete > Row.
4) Save the form as Adobe Dynamic XML Form (*.pdf).
5) In the Hierarchy palette, select the subform that contains the table.
6) In the Object palette, click the Subform tab and select Flowed from the Content list.

7) In the Hierarchy palette, select the body row (for example, Row1).
8) In the Object palette, click the Binding tab and select the Repeat Row For Each Data Item option.



9) Connect to a data source. See *Connecting to a data source*.
You must connect to a data source, such as an XML schema, to show data in the table.
10) Select a data file. See *To preview a form using sample data*.
You must select a data file to view and test the form with the data.
11) View the form in the Preview PDF tab.

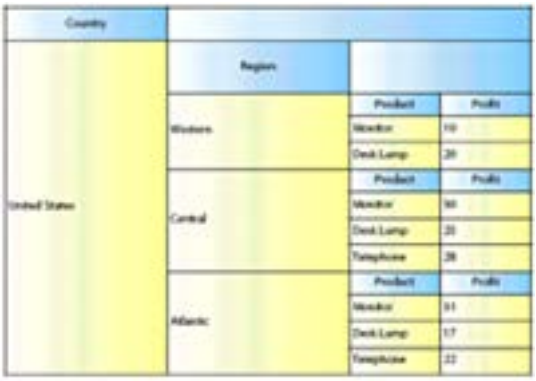
7.18. To create a table that groups data

You can display data in a grouped fashion inside a table. Grouping organizes and arranges data into relationships, such as Country or Region. You can nest groups to easily identify relationships and see trends. It also helps present summaries, such as totals and counts.

For example, if you supply products, such as monitors, desk lamps, and telephones, you may want to see which country and region is selling the most product. You can create this table, which contains a nested table inside a nested table inside the main table:

- Country is the table header for the main table.

- Region is the table header for the first nested table.
- Product and Profit makes up the table header for the nested table inside the first nested table.



For example, the sample XML file you connect to could have the following syntax:

```
<form1>
<Sales>
 <SalesData>
    <country>United States</country>
    <CountryData>
         <RegionRow>
         <region>Western</region>
    <RegionData>
    <Item>
         <product>Monitor</product>
         <profit>10</profit>
    </Item>
    <Item>
         <product>Desk Lamp</product>
         <profit>20</profit>
    </Item>
    </RegionData>
         </RegionRow>
         <RegionRow>
         <region>Central</region>
    <RegionData>
    <Item>
         <product>Monitor</product>
         <profit>30</profit>
    </Item>
    <Item>
         <product>Desk Lamp</product>
         <profit>25</profit>
    </Item>
    <Item>
         <product>Telephone</product>
         <profit>28</profit>
    </Item>
    </RegionData>
```

```
            </RegionRow>
            <RegionRow>
        <region>Atlantic</region>
        <RegionData>
        <Item>
            <product>Monitor</product>
            <profit>31</profit>
        </Item>
        <Item>
            <product>Desk Lamp</product>
            <profit>17</profit>
        </Item>
        <Item>
            <product>Telephone</product>
            <profit>22</profit>
    </Item>
            </RegionData>
            </RegionRow>
        </CountryData>
     </SalesData>
    </Sales>
    </form1>
```

Before you perform this task, you must ensure that the following settings are in effect:

- Ensure that you are connected to a data source. See *Connecting to a data source*.
- To test the form with sample data, ensure that you are pointing to a data file. See *To preview a form using sample data*.

To create the main table

1) Select File > Save As, type a name for the file, and select Adobe Dynamic XML Form (*.pdf). Click OK.
2) Select Table > Insert Table.
3) In the Insert Table dialog box, enter the number of columns and rows. For example, enter 2 for the number of columns, 1 for the number of rows.

   You can enter a maximum of 20 columns and 50 rows. You can add more columns and rows after the table is created by using the Insert commands in the Table menu.
4) (Optional) To include a header row, select Include Header Row In Table.
5) (Optional) To include a footer row, select Include Footer Row In Table.
6) Click OK.
7) Rename the header row. For example, name the first header `Country` and delete the header text for the second column.
8) Resize the table.

To create the first nested table

1) Drag the Table object from the Object Library palette to a cell in the table.
2) Rename the header row text. For example, name the first header `Region` and delete the header text for the second column.

To create the last nested table inside the first nested table

1) Drag the Table object from the Object Library palette to a cell in the table to create another nested table. For example, drag it to the second cell of the body row in the first nested table.
2) In the Insert Table dialog box, enter the number of columns and rows.
3) (Optional) To include a header row, select Include Header Row In Table.
4) (Optional) To include a footer row, select Include Footer Row In Table.
5) Click OK.
6) Rename the header row. For example, change the first header to `Product` and the second header to `Profit`.

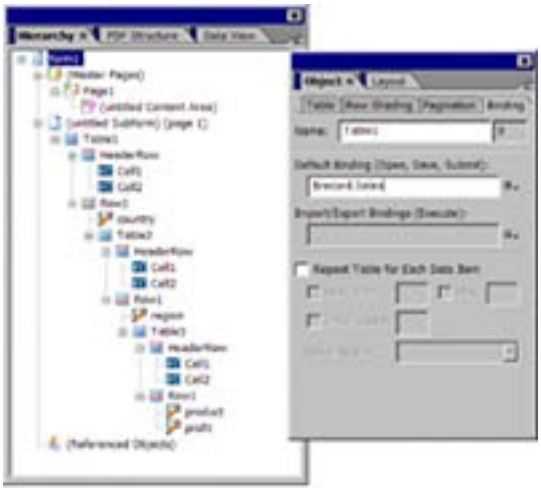
   The form should now look like one.

To show data in the cells

1) In the Data View palette, drag a node to the cell under the first heading. For example, drag the country node to the cell under the Country heading.
2) Repeat for the remaining cells.

   For example, drag the region node to the cell under the Region heading. Drag the product node to the cell under the Product heading. Drag the profit node to the cell under the Profit heading.
3) In the Hierarchy palette, select each Row1 and, in the Binding tab of the Object palette, select Repeat Row For Each Data Item.
4) Match the table, row, and field names to those in the sample XML file.

   For example, select Table1 in the Hierarchy palette. Then, in the Object palette, click the Binding tab and type `$record.Sales` in the Data Binding (Open, Save, Submit) box.



For example, set the default binding as indicated in this table.

Select in the Hierarchy palette	Set Data Binding to the corresponding string
Row1	SalesData
country	country
Table2	CountryData
Row1	RegionRow[*]
region	region
Table3	RegionData
Row1	Item[*]
product	product
profit	profit

5) Select the subform that the main table is in and, in the Subform tab of the Object palette, select Flowed from the Content list.
6) Format the table. For example, add borders and shading.
7) View the form in the Preview PDF tab.

7.19. To make a table optional

Making a table optional is useful when you do not want to display information contained in a table or when no data is available to display in the table.

1) Connect to a data source. See *Connecting to a data source*.

   You must connect to a data source, such as an XML schema, to be able to show data in the table.
2) Select a data file. See *To preview a form using sample data*.

You must select a data file to view and test the form with the data.

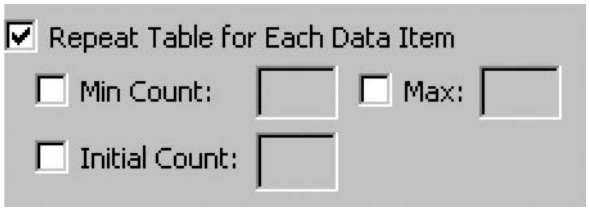
3) Ensure that the table is in a subform that is set to Flowed.

4) Select the table in which the number of rows change depending on how much information is in the data source. See *To select a table, row, column, cell, or section*.

   If you select a table that has a fixed number of rows and columns and make it optional, the header row is repeated for each data item. To change this table to one in which the number of rows will change according to the data, deselect Repeat Row For Each Data Item for the header row but select Repeat Row For Each Data Item for the body row.

5) In the Object palette, click the Binding tab and select Repeat Table For Each Data Item.

6) Deselect Min Count.



7) View the form in the Preview PDF tab.

   If you have not connected to a data source, the table should not appear. If you have bound the cells in a table to a data source and there is data for the bound cells, the table should appear.

7.20. To perform calculations in a table

You can use the `sum` function in FormCalc to total the values in a column. You must add the calculation to the footer row of the table.

Country	Q1	Q2	Q3	Total
Canada	10K	15K	20K	45K
US	15K	20	25K	60K
Total	25K	35K	45K	105

1) Select the cell in the footer row where you want the calculation. For example, select the cell that corresponds to the total for Q1.

2) In the Script Editor, select Calculate from the Show list.

3) In the Language list, select FormCalc.

4) In the Run At list, select Client.

5) In the Script Source field, insert your FormCalc calculation. For example, to calculate the total for the Q1 data, type the following expression:

```
sum (Table.Row[*].Q1[*])
```

6) Repeat for the totals for Q2 and Q3.

7) To calculate the totals for the country, type the following expression:

```
sum(Q1 + Q2 + Q3)
```

8) Repeat for the remaining row totals.

9) For the grand total, type the following expression:

```
sum(TotalQ1 + TotalQ2 + TotalQ3)
```

TIP: *To show the data as 10K, in the Object palette, click the Cell tab, click Patterns, and type z9'K' in the Pattern box. In the Value tab, select Calculated - Read Only from the Type list.*

7.21. To use the TableCalcs custom object

Use the TableCalcs custom object to include calculations in fields on one or more pages in a form. Here are the calculations that you can set on a form:

Count

Calculates the number of non-null fields that match the specified field name.

Sum

Calculates the sum of non-null fields that match the specified field name.

Average

Calculates the average of non-null fields that match the specified field name.

Unless you provide a range of pages for the TableCalcs object, the calculation is applied across the entire form.

Here is the list of function calls to set the calculations.

```
NUMBER Count(STRING sFieldName, [NUMBER pageNum])
NUMBER Count(STRING sFieldName, [OBJECT xfaObject])
NUMBER Count(STRING sFieldName, [NUMBER startPage], [NUMBER endPage])
NUMBER Count(STRING sFieldName, [OBJECT xfaObject], [OBJECT xfaObject])

NUMBER Sum(STRING sFieldName, [NUMBER pageNum])
NUMBER Sum(STRING sFieldName, [OBJECT xfaObject])
NUMBER Sum(STRING sFieldName, [NUMBER startPage], [NUMBER endPage])
NUMBER Sum(STRING sFieldName, [OBJECT xfaObject], [OBJECT xfaObject])

NUMBER Average(STRING sFieldName, [NUMBER pageNum])
NUMBER Average(STRING sFieldName, [OBJECT xfaObject])
NUMBER Average(STRING sFieldName, [NUMBER startPage], [NUMBER endPage])
NUMBER Average(STRING sFieldName, [OBJECT xfaObject], [OBJECT xfaObject])
```

The first argument, `sFieldName`, is the field name, which indicates which fields are calculated. The second and third object optional arguments (`startPage`, `endPage`) are page numbers. The

page numbers can be a number or a reference to another field object on the form. When you provide a field object reference, the page number is the page where the object reference is located.

1) Add the TableCalcs object to your existing form design.
2) In the Hierarchy view, move the TableCalcs object under the root subform.
3) Write a script that will invoke the function of the script object.

   This example returns the Sum of all fields called NumericField1 on the entire form.

   ```
   this.rawValue = tableCalcs.Sum("NumericField1");
   ```

   This example returns the number of non-null fields called NumericField1 on page 3 only.

   ```
   this.rawValue = tableCalcs.Count("NumericField1", 2);
   ```

   NOTE: *Numeric page references are 0-based.*
4) Save, preview, and test the form in the Preview PDF tab.

   You may want to enable the JavaScript Debugger to catch any errors while testing the form.

For more information about scripting, including the JavaScript Debugger in Acrobat Professional, see *Scripting Basics* .

RELATED LINKS:

About subforms

To add objects to a form design

Objects that support scripting and calculations

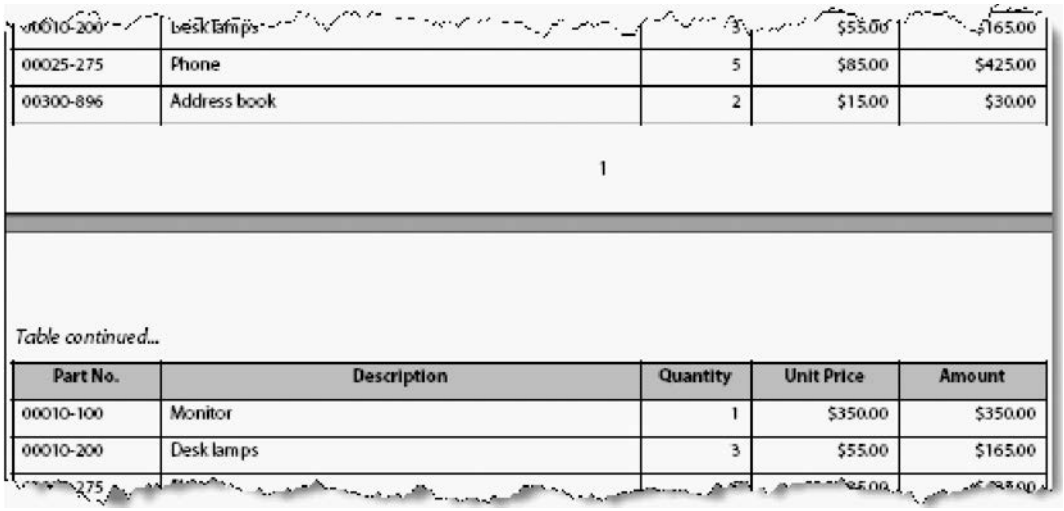
7.22. Working with pagination in tables

To set up a table to span multiple pages

To enable tables to automatically flow to the next page, you must ensure the following conditions:

- The table must be in a subform that is set to Flowed in the Subform tab of the Object palette.
- The Allow Page Breaks Within Content option is selected in the Table tab.
- The table is not in a group.

Now you can make adjustments to the table to ensure that the information is displayed the way you want when the table spans multiple pages. You can include a Table continued tag at the top of the next page if the table spans more than one page.



00010-200	Desk lamps	3	$55.00	$165.00
00025-275	Phone	5	$85.00	$425.00
00300-896	Address book	2	$15.00	$30.00

1

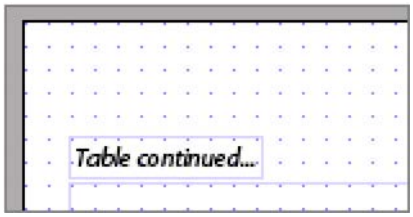
Table continued...

Part No.	Description	Quantity	Unit Price	Amount
00010-100	Monitor	1	$350.00	$350.00
00010-200	Desk lamps	3	$55.00	$165.00

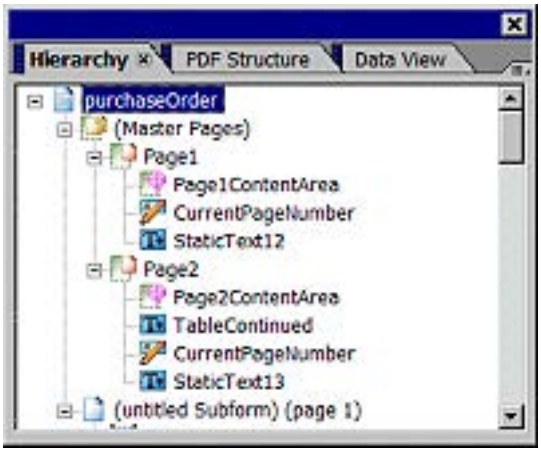
If you want the header row to repeat on the next page, see *To repeat a header or footer row on subsequent pages*.

To set up a master page that includes the Table continued tag

1) Create a table that grows based on the amount of data from a data source. See *To set the rows in a table to adjust to accommodate data*.
2) Select the header row.
3) In the Object palette, click the Pagination tab and select Include Header Row In Subsequent Pages.
4) Select View > Master Pages to display the Master Pages tab if it is not already displayed.
5) Click the Master Pages tab and select Insert > New Master Page.
6) Resize the content area so that it is smaller on the new master page.
7) In the Object Library palette, click the Standard category and drag the Text object above the content area.
8) Double-click the default text and type `Table continued ...`
9) Select the text and, in the Font palette, set the text to Bold and Italic.



10) In the Hierarchy palette, rename the (untitled Content Area) on Page 2 to Page2ContentArea.



To set the pagination for the table

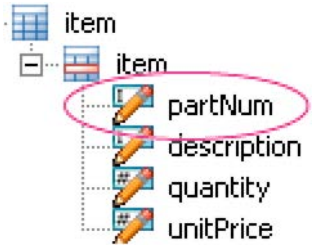
1) In the Design View tab of the Layout Editor, select the table. See *To select a table, row, column, cell, or section*.
2) In the Object palette, click the Table tab, and ensure that Allow Page Breaks Within Content is selected.

   NOTE: *Although the Allow Page Breaks Within Content option is selected by default for tables, it is deselected for table rows. You must select this option for the table rows to allow page breaks within the table.*
3) In the Object palette, click the Pagination tab and, next to the Overflow list, click the arrow and select Go To Content Area > Page2ContentArea.

To connect to a data source

1) Ensure that you are connected to a data source. See *Connecting to a data source*.

   For example, if you or your administrator installed the samples that come with Designer, locate and open the following schema: Purchase Order.xsd in the Data Connections dialog box.
2) In the Data View palette, locate the partNum node.



3) Drag the node into the first cell under the Part No header.
4) Repeat steps 2 and 3 for description, quantity, and unitPrice.

To select a data file

1) Select File > Form Properties and click the Preview tab. Browse to the location of the data file you want to use.

   NOTE: *You will need to select a data file that contains enough data to fill a table that spans more than one page.*

2) View the form in the Preview PDF tab.

To repeat a header or footer row on subsequent pages

1) Click and drag to select the table header or footer.
2) In the Object palette, click the Pagination tab, and do one of the following actions:
   – Select Include Header Row In Subsequent Pages.
   – Select Include Footer Row In Subsequent Pages.

   NOTE: *The table must be inserted into a subform that is already set to Flowed.*

RELATED LINKS:

Edit Conditional Breaks dialog box

Table properties in the Pagination tab

Section properties in the Pagination tab

7.23. Working with header and footer rows in tables

To change a row to a header, body, or footer row

After you insert a table, you can change a row to another type. For example, you can change a body row to a header row to set up a table inside a table. Also, you can change a body row to a footer row to show summaries in the last row that you want to appear on each page in the form.

1) Select a row. See *To select a table, row, column, cell, or section*.
2) In the Object palette, click the Row tab and select an item from the Type list.

To control table, header row, body row, footer row, and section breaks using conditional statements

Designer provides the capability to create customized conditional breaks for table objects as well as for header rows, body rows, footer rows, and sections. Instead of paginating these objects in response to data overflow, conditional breaks allow you to manually control how these objects break on a form based on a series of checks called conditional statements.

Through conditional statements, you can verify data for a field within a table, header row, body row, footer row, or section against previous instances of that field. The table, header row, body row, footer row, or section can then be broken in response to a change in the data supplied to the field.

For example, on a telephone bill, you could break a table object in response to changes in the field that stores the date of each billing entry. The telephone bill could then be visually broken down by date, making it easier for a user to read.

In addition to specifying a breaking condition, you can also specify leader and trailer subforms, and indicate where to place the next instance of the repeating subform on the form.

Before you perform this task, you must ensure that the table is in a subform that is set to Flowed.

1) Select a table, header row, body row, footer row, or section. See *To select a table, row, column, cell, or section*.
2) In the Object palette, click the Pagination tab.
3) Click the Edit button and then click the Add button to insert a new conditional break list item.
4) Select a scripting language from the Language list. The conditional break condition statement is created by using the scripting language you select.
5) In the Run At list, select where you want the conditional break to execute.
6) Click Insert Sample Expression and select the form design object within the table, header row, body row, footer row, or section to use as the comparison field for the conditional break. Alternatively, you can enter your own conditional statement in the field. To correctly evaluate as a conditional break, however, any user-defined conditional statements must evaluate to either true or false.
7) Select when you would like the table, header row, body row, footer row, or section to break by selecting either Before or After. Selecting Before inserts a break immediately before the current instance of the table, header row, body row, footer row, or section is inserted into the form, and selecting After inserts the break immediately after.
8) In the To field, select where you want to place the remaining occurrences of the table, header row, body row, footer row, or section.
9) In the Trailer and Leader lists, select trailer and leader subforms to use for the current conditional break, if any.
10) Repeat steps 2 to 9 for each conditional break you want to include for the selected object, and click OK when you have finished adding entries to the list.

    After you create all of your conditional break entries, you should review the order in which they appear in the Edit Conditional Breaks dialog box. Designer processes the conditional breaks specified in this dialog box in sequential order from top to bottom. Each conditional break for which the conditional statement evaluates to true is executed.

    Use the Up and Down buttons to move individual conditional break list entries into the order you want.

RELATED LINKS:

Header and footer row properties in the Pagination tab

Body row properties in the Binding tab

7.24. Working with cells and cell contents

To clear a cell

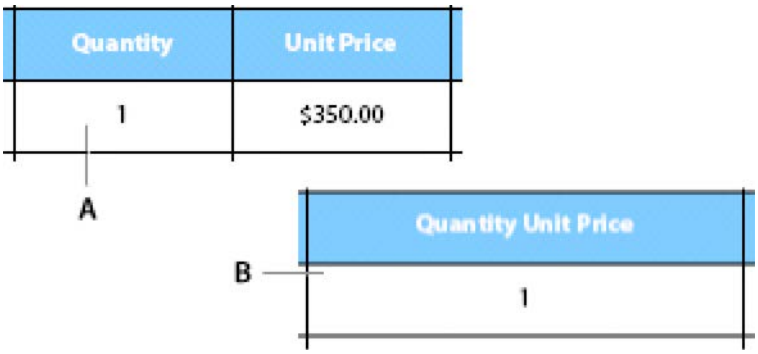
You cannot delete a cell. However, you can clear the contents of the cell. This action changes the cell type to a Text object that is empty.

1) Select the cell. See *To select a table, row, column, cell, or section*.
2) Select Edit > Clear Contents.

To merge and split cells

You can combine two or more adjacent cells in the same row into a single cell. For example, you can merge several cells horizontally to create a table heading that spans several columns. You can only merge cells that are in the same row.

You cannot merge the data from two cells into one cell. You can only merge text. For example, if the header row is merged, the text is merged because these two cells are Text objects. However, if the body row is merged, and the cells are bound to items in the data source, only the left most cell is retained.



A.

Before merging cells

B.

After merging cells, the header row text is merged, but the data for the body row only retains data from the left-most cell

TIP: *It is a good idea to merge cells at the end of the process of designing a table because adding new columns or removing columns does not work the same if there is a merged cell in the table.*

To merge cells into one cell

1) Select the cells you want to merge. See *To select a table, row, column, cell, or section*.
2) Select Table > Merge Cells.

If the cells contain Text objects, Designer combines the text. If the cells contain other objects, the objects are deleted.

TIP: *To make the columns the same size by using the Distribute Columns Evenly command after you have merged cells, delete the hidden columns first by using the Table > Delete > Column command.*

To split a merged cell

1) Select a merged cell. See *To select a table, row, column, cell, or section*.
2) Select Table > Split Cell Horizontally.

   The split cells will be converted to an empty Text object or, if the original cells were subforms, they will be converted to empty subforms. For example, if the merged cell was a numeric field, all the split cells will be numeric fields.

   NOTE: You can use these two buttons in the Cell tab of the Object palette: Merge Cells , Split Cells Horizontally .

To orient text vertically in cells

You can change the orientation in table cells so that the information is displayed vertically instead of horizontally.

Only cells can be rotated (not rows, columns, sections, or entire tables).

1) Select the cell. See *To select a table, row, column, cell, or section*.
2) In the Layout palette, select one of the rotation buttons.
3) Click the orientation you want. For example, the vertical text in the previous example is rotated 90°.
4) In the Paragraph palette, select the align option that works best for the cell. For example, the vertical text in the previous example is aligned right and aligned to the top.

   NOTE: *If you rotate a field that users fill in, users will have to enter their data at the angle of rotation.*

To change a cell to another object type

By default, all cells are set to text objects. Text objects present read-only text that users cannot edit. You can use text objects to do these tasks:

- Label an area in the form, such as headers in the table
- Provide instructions for filling the form
- Enhance the form

You can change the cell to any other type of object, such as a numeric field or text field. In this way, you could use a table to lay out an entire form.

A cell can also be a button that submits data, executes a web service operation or database query, or emails data to someone.

A cell can even be a subform. A subform acts as a container for other objects, including fields, boilerplate objects, and other subforms. Subforms also help to position objects relative to each other and provide structure. If a cell is a subform, the cell can hold more than one object. For example, it can have two buttons.

When a cell becomes an object other than a Text object, it behaves differently in a cell than if it were in the form design on its own. For example, a Text Field object in a cell has the caption set to None.

Here are ways to change a cell to another object type; however, the results are different, depending on what is already in the cell and what you change it to:

- Use the Type list in the Cell tab of the Object palette.
- Drag an object from the Object Library palette into a cell.

Depending on what you select from the Type list or Object Library palette and what is in the cell, the new object replaces the existing object. If you change a cell to a subform, objects that existed in the cell already are wrapped in the subform. For example, if a text field already existed in the cell and you choose Subform from the Type list, the Text Field is wrapped in the subform. If a text field already existed in the cell and you choose Numeric Field from the Type list, the Numeric Field replaces the Text Field.

- Drag an existing object in the form design that is outside of the table into a cell. This method preserves the look of the object. For example, if you drag a formatted phone number field into a cell, the caption is retained along with the size of the field.

  If you click and drag a Subform object that is outside of the table into a cell, it replaces the contents of the cell.

  NOTE: If you change a cell to a text field, you can select the Allow Multiple Lines option in the Cell tab of the Object palette to show more than one line of text.

To change a cell to another object type by using the Type list in the Cell tab of the Object palette

1) Select the cell. See *To select a table, row, column, cell, or section*.
2) In the Object palette, click the Cell tab and select another object type from the Type list.

To change a cell to another object type by dragging an object from the Object Library palette

1) Drag the object from the Object Library palette into the cell where you want it to appear.

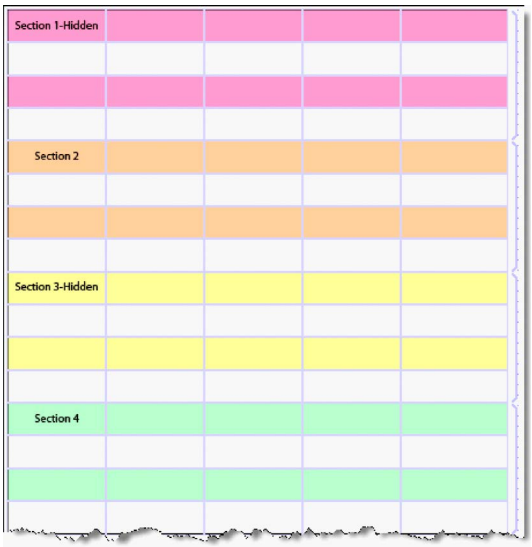
To change a cell to another object type by dragging an object from the form design

1) In the form design, drag the existing object that is outside of the table into the cell where you want it to appear.

7.25. Working with table sections

A table section is a grouping of rows that remains a unit. Using sections lets you organize your table. A section can have its own header and footer so that you can show detailed information grouped by common values. Each section appears in a predictable order in the form. By default, Designer inserts a header row for each section.

After you create a table, you can group the rows into sections. You can also apply row shading to each section so that each section has different colors. For example, here is a table that is grouped into four sections.



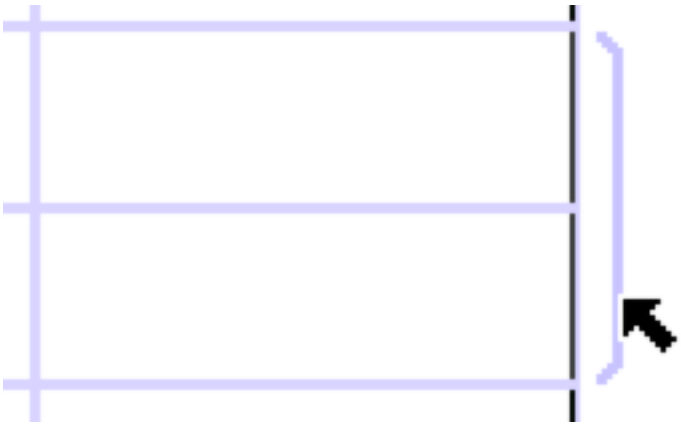
You can also make a table section optional when you do not want to display information contained in a section. For example, here is a table where section 1 and 3 are hidden.

To insert a table section

1) Select a cell or a row. See *To select a table, row, column, cell, or section*.
2) Select Table > Insert and then select Section Above or Section Below.
   Designer inserts one header row and one body row for the section.

To group rows into a section

1) Select the rows that you want to group. See *To select a table, row, column, cell, or section*.
2) Select Table > Group as Section.
   Designer shows a bracket on the right side of the table to indicate a section.

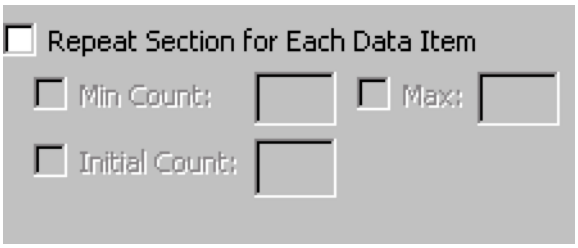


To ungroup a section into rows

1) Select the section and select Table > Ungroup Section.

To make a table section optional

1) Select the section. See *To select a table, row, column, cell, or section*.
2) In the Object palette, click the Binding tab and deselect Repeat Section For Each Data Item.



TIP: To show a table section that has been hidden, select the section, select Repeat Section For Each Data Item, select Min Count, and type `1` *in the box.*

7.26. Creating choice sections in tables

A choice section is a section within a table that can be configured to customize the display of specific rows from within the section.

This customization is applied to individual rows within the section by using conditional statements. For example, you can configure a choice section to display instances of a specific row using different text colors, depending on the value of a specific field within that row.

You can also add header, body, and footer rows to a choice section, and remove header, body, and footer rows from a choice subform set by choosing one of these options:

- Using the Edit Data Nominated Subforms dialog box
- Manually editing the contents by dragging subform objects into or out of the choice section by using the Hierarchy palette

To create a choice section from an existing section

1) Select a table section and in the Object palette.
2) Click the Section tab and select One Subform from Alternatives from the Type list.

To create a choice section from a row

1) Select a row within the table and select Table > Group as Choice Section.

To add a header, body, or footer row to a choice section

1) Select a choice section.
2) In the Object palette, click the Section tab and ensure that Select One Subform from Alternatives is selected from the Type list.
3) Click the Edit Alternatives button.
4) Click the Add button to insert a new Alternative Subforms list item. You can also drag rows into the choice subform set by using the Hierarchy palette.

   Adding a new row adds a duplicate of the currently selected row. If no row is selected, a new un-named row is added to the choice section.
5) Use the Up and Down buttons to move the new row entry to the location you want. Ordering entries in the Alternative Subforms list is important because entries are processed sequentially at run time, and the first entry with an expression that evaluates to true appears on the form.

   IMPORTANT: *This procedure outlines the process for adding new rows to an existing choice section. When you add rows in this way, Designer first creates a new row object and then adds the row to the table and the choice section. If you want to add existing table rows to a choice section, drag row objects into the choice section from the Hierarchy palette.*

To remove a header, body, or footer row from a choice section

1) Select a choice section.
2) In the Object palette, click the Section tab and click Edit Alternatives.
3) Select an entry from the Alternative Subforms list and click Delete.

   IMPORTANT: *Removing a row by using the Edit Data Nominated Subforms dialog box completely removes the row object from your form design. If you want to preserve the row but remove it from the choice section, you must manually drag the row out of the choice section by using the Hierarchy palette.*

RELATED LINKS:

Edit Data Nominated Subforms dialog box

7.27. Creating data bindings and conditional statements for choice table sections

After you create a choice section, you can create data bindings with or without conditional statements for the various rows within the section. Using conditional statements provides greater control over when rows within the choice section are displayed on your form than specifying only a data binding.

For each row listed in the Alternative Subforms list in the Edit Data Nominate Subforms dialog box, you can specify a data node from the data connection. If you specify a conditional statement, you can enter an expression that will evaluate to either true or false at run time. Only the first entry in the Alternative Subforms list with a conditional statement that evaluates to true will appear on your form.

To create data bindings and conditional statements for choice table sections

1) Select a subform set object.
2) In the Object palette, click the Section tab and ensure that Select One Subform from Alternatives is selected from the Type list.
3) Click Edit Alternatives.
4) Choose how you want to specify a row data binding from within the choice section by performing one of the following actions:
   – Click Choose Subform Whose Name Matches Data Element or Attribute to bind data nodes to rows by name. In this case, the names of the associated data nodes must match the names of the rows on your form design.
   – Click Choose Subform Using Expression to bind rows from your form design to data nodes from the data connection by manually specifying a binding.
5) In the Data Connection list, select the data source you want to bind data from.
6) Click Add to insert a new row into your table, or select an existing list item. Adding a new row adds a duplicate of the currently selected row. If no row is selected, a new unnamed row is added to the choice section.
7) Insert a new subform into the choice subform set, or select an existing list item. Adding a new subform adds a duplicate of the currently selected subform. If no subform is selected, a new unnamed subform is added to the choice subform set.
8) In the Name field, enter a name for a new row object, if necessary. If you are binding the row by data element or attribute name, ensure that the name in the field matches exactly with the name of the associated data node.

   If you selected the Choose Subform Whose Name Matches Data Element or Attribute in step 4, you can go directly to step 12. Otherwise, continue with step 9.

9) Click the arrow to the right of the Binding field and select a data node from the pop-up menu. Designer automatically populates the Binding field with a scripting reference to the data node you select. Alternatively, you can manually type a reference into the field.
10) Select a scripting language from the Language list.
11) In the Expression field, enter your scripting to perform the actions or processing you want for the specified row and data node.
12) Repeat steps 6 to 10 for any additional rows within the choice section, and then click OK.

RELATED LINKS:

Edit Data Nominated Subforms dialog box

8. Using subforms

8.1. About subforms

A subform is a section in the form design that provides anchoring, layout, and geometry management for objects. The objects in a subform can be arranged in rows, columns, or some other kind of balanced arrangement.

More than one subform can be used in a form design. Subforms can be placed inside other subforms. This relationship is displayed in the Hierarchy palette.

Subforms are used to organize a form into different sections. They can also be used to create a form that contains sections that automatically expand and shrink to accommodate the data. If you set a subform to grow, the layout of the form changes in response to the amount of data that is merged when the form is rendered. When the data is merged, subforms ensure that objects and their data are positioned consistently relative to each other. Subforms can be used to match the data hierarchy in XML data.

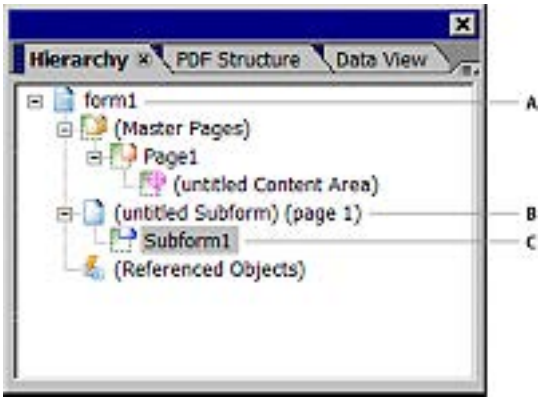
If you are designing a form that has a fixed layout, it is unlikely that you will need to work with more than one subform because the default subform positions the objects automatically.

You can manipulate the properties of a subform in the Subform and Binding tabs of the Object palette. You can define these properties:

- Give a meaningful name to the subform (recommended)
- Enable the subform to span page breaks or force it to be rendered on the next page when the data is merged
- Specify whether to place the subform after the previous subform, in the specified content area, or on a page that is formatted according to the specified master page
- Specify whether to place the subform in the same content area as the previous or next subform
- Specify the flow order of merged data after the subform is placed.
- Define the subform as visible, invisible, or hidden
- Specify a locale for the subform
- Specify whether the subform will repeat its objects each time a unique data item is provided for one of its objects
- If required, create an overflow leader or trailer for a subform that is capable of repeating the rendering of its objects
- Specify a binding method for controlling how the subform’s objects are mapped to data

All forms contain a root (parent) subform. In the Hierarchy palette, the root subform (form1) is displayed as the top-level node with the default page subform (untitled Subform) appearing as a child node below the root subform.

Designer automatically adds to every page a default subform that covers the whole page, and corresponds in size and position to the default content area on the master page. Any subforms that you subsequently add to the pages are nested in and appear below the default page subform in the Hierarchy palette.



A.

Root (parent) subform

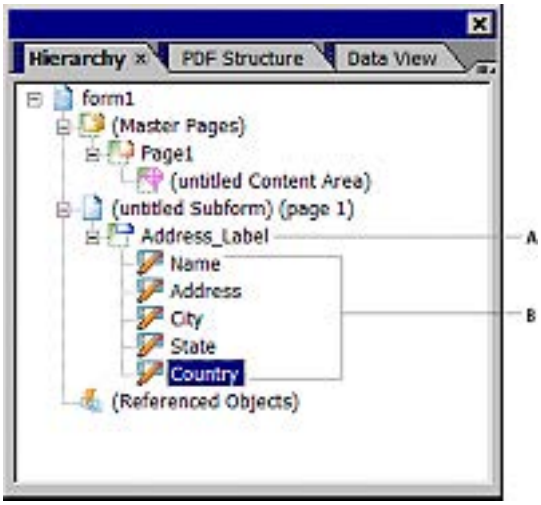
B.

Default subform

C.

New subform

In the Hierarchy palette, each subform is represented by a node, and the objects wrapped in a subform are displayed under the subform node. The children of the subform do not inherit changes made on the subform level; the properties of each object must be defined individually.



A.

Renamed subform

B.

Objects in the subform

If you look at the Purchase Order sample, the root subform, form1, is shown as the top-level node with the default page subform, purchaseOrder, appearing below as a child of the root subform. The other subforms used to wrap objects on the page (header, detailHeader, detail, and total) are nested under the page subform. In the Hierarchy palette, each subform is represented by a node, and the objects wrapped in a subform are displayed under each node.

RELATED LINKS:

Using subforms

To add a subform

To wrap and unwrap objects in a subform

Subform properties in the Subform tab

Subform properties in the Pagination tab

Subform properties in the Binding tab

8.2. Subforms that position content

Objects in the subform are positioned according to their individual X and Y coordinates. When the form is rendered, the subforms are placed in an order determined by their positions in the Hierarchy palette.

All subforms except the root subform (form1) are set to position content by default. When a subform is set to position content, the subform can still expand to fit any amount of merged data, but none of the objects within the subform can move from their anchor points. As a result, if a subform is configured to position content, you must frequently test your form design to make sure that any objects within the subform that you expect to expand in response to data merging do not interfere with other objects in the form design. Remember that any objects you configure to expand, such as text field objects, can possibly overlap other objects when the form is rendered. To avoid this design concern, you can set the subform to flow and expand to fit the content. (See *Subforms that flow content*.)

Designer automatically sets the default page subform to position content to make it easier to create forms that have a fixed layout and are interactive forms. For more information, see *About subforms*.

However, when designing a form whose layout will adjust to accommodate data, you will need to reset the default page subform to flow content after you complete the form design. It is a good idea to do this last so that the subform remains visible and the objects you place within the subform remain in the intended position on the page.

You use the Position Content option in the Type list in the Subform tab to position content in a subform. When you select the Position Content option, the X and Y coordinates of each object within the subform are maintained. The objects are placed at their X and Y coordinates relative to the position of the subform.

When designing nested subforms to emulate tables, the header subform has to be "positioned" content. Otherwise, the field elements are not positioned properly on subsequent pages.

8.3. Subforms that flow content

Objects in the subform are positioned during the data-merging process so that no objects are rendered on top of each other. Subforms that flow content are placed in ascending order. You can use subforms that flow content when you need the form to automatically expand to fit the contents.

The root subform (form1) is set to flow content by default. The root subform always flows content according to the flow direction option that you apply to the associated default content area. Because the root subform is set to flow content, all subforms nested under the root subform automatically flow, as needed, from one form page (content area) to the next when data is merged. When the form is rendered, the subforms under the root subform are placed in descending order according to their position in the object hierarchy.

You can set any other subform to flow content as well. Each subform that is set to flow content can hold varying amounts of data, whereas the objects within the subform move together during the data-merging process so that none of the objects interfere with each other.

You use the Flow Content option in the Type list in the Subform tab to flow content in a subform. When you select the Flow Content option, the objects are positioned based on the Flow Direction option you select. Notice that the Flow Direction list and the Allow Page Break option are enabled after you select Flow Content in the Type list.

NOTE: When you resize a subform that flows content, the subform is automatically converted to a subform that has a fixed layout, which prevents page breaks.

RELATED LINKS:

Using subforms

8.4. Creating and configuring subforms

RELATED LINKS:

To specify a subform to span multiple pages

Subform properties in the Binding tab

To define custom data-binding properties for a subform

Making objects visible, invisible, or hidden

Subform properties in the Subform tab

Layout properties in the Layout palette

Border properties in the Border palette

To add a subform

You can add a subform from the menu or the Library palette.

To add a subform from the menu

1) With the page of the form displayed, select Insert > Standard > Subform.

To add a subform from the Library palette

1) Do one of the following actions:
   – In the Library palette, click the Standard tab, and then select the Subform object and draw the object on the page of the form.
   – In the Standard tab of the Library palette, drag a Subform object onto the page of the form.
   – In the Library palette, click the Standard tab, and then double-click the Subform object.

To name a subform

To name or rename a subform, you must first select the subform.

To name or rename a subform using the Hierarchy palette

1) In the Hierarchy palette, right-click the subform and select Rename Object.
2) Type the new name and press Enter.

To name or rename a subform using the Binding tab of the Object palette

1) Select the subform.
2) In the Binding tab, type a new name for the subform in the Name box and press Enter. Designer maintains the occurrence number automatically.

To make a subform visible, invisible, or hidden

1) Select the subform.
2) In the Object palette, click the Subform tab.
3) In the Presence list, select the option that suits your needs:
   – To make the object visible on-screen, visible in the printed form, and occupy space in the form layout, select Visible.
   – To make the object visible on-screen, not visible in the printed form, and occupy space in the form layout, select Visible (Screen Only).
   – To make the object not visible on-screen, visible in the printed form, and occupy space in the form layout, select Visible (Print Only).
   – To make the object not visible on-screen, not visible in the printed form, and occupy space in the form layout, select Invisible.
   – To make the object not visible on-screen, not visible in the printed form, and not occupy any space in the form layout, select Hidden (Exclude From Layout). The Hidden (Exclude from Layout) option works as described only when you apply this option to a subform that is placed within a parent subform that is set to Flowed.

To specify a locale (language and country or region) for a subform

1) Select the subform.
2) In the Object palette, click the Field tab.
3) In the Locale list, select one of these options or one of the provided alternatives for the subform's localization setting:
   – To use the default locale specified in the Defaults tab of the Form Properties dialog box, select Default Locale.
   – To use the system locale of the user's computer, select Viewer's System Locale.

   NOTE: *At design time and run time, formatted values in the field are displayed in the locale-sensitive format.*

To wrap and unwrap objects in a subform

Objects are wrapped in a subform when they are contained by the subform.

To add objects to an existing subform

1) Select the objects to include in the subform, and then select Insert > Wrap in Subform.

To unwrap the objects in a subform

1) Select the subform that contains the objects to unwrap, and then select Insert > Unwrap Subform.

8.5. To specify how a subform manages content

By default, all subforms except the root subform are defined to position content. The root subform always flows content according to the Flow Direction setting of its associated content area. Because the root subform flows content, all subforms nested under the root subform can flow from one form page or content area into the next automatically if required when data is merged.

In contrast, when a subform positions content, the area bounded by the subform may expand to accommodate any amount of merged data, but none of the objects in the subform can move from their anchor points. Therefore, if a subform contains objects that merge with variable sizes of data, you must verify that those objects do not expand to the extent that they overrun the area occupied by another object. Any objects that expand in response to data merging, such as a text field, have the potential to be rendered on top of other objects.

To avoid this design problem, you can make the subform flow content. A subform that flows content places objects correctly during the data-merging process so that none of its objects are rendered on top of each other.

The default subform for the page positions content to support the creation of interactive forms and forms that have a fixed layout. If you are authoring a form that contains subforms that adjust to accommodate data, you need to work with subforms that position content as well as those that flow content. Different techniques are available for working with subforms in this type of form design. The approach

you take depends on your experience with creating form designs whose layout adjusts to accommodate data. For more information, see *Creating interactive forms that have a flowable layout*.

To specify how a subform positions content

1) In the Object palette, click the Subform tab and, in the Content list, select one of these options:
   – Positioned
   – Flowed

To specify a flow direction for the objects in a subform that flows content

1) In the Object palette, click the Subform tab and, in the Flow Direction list, select one of these options (objects are always placed starting at the top of the subform):
   – Top to Bottom
   – Western Text
   – Right to Left

   NOTE: *The Flow Direction option is only available when Flowed is selected in the Content list.*

To change a subform that positions content into a subform that flows content

1) Select the subform in which you want content to flow.
2) In the Object palette, click the Subform tab and, in the Content list, select Flowed.
3) In the Flow Direction list, select one of these options (objects are always placed starting at the top of the subform):
   – Top to Bottom
   – Western Text
   – Right to Left

RELATED LINKS:

Subform properties in the Subform tab

To position subforms

To specify how to merge data between subforms

8.6. To position subforms

To specify where to position a subform

1) In the Object palette, click the Pagination tab, and then click the arrow to the right of the Place box and select one of these presentation options:
   – Following Previous

- In Content Area > *[name_of _content_area]*
- Top of Next Content Area
- Top of Content Area > *[name_of _content_area]*
- On Page > *[name_of _page]*
- Top of Next Page
- Top of Page > *[name_of _page]*
- On Odd Page
- Top of Next Odd Page
- On Even Page
- Top of Next Even Page

To keep the subform within the same content area or page as the previous or next subform

1) In the Object palette, click the Pagination tab, and then select Keep W/ Previous or Keep W/ Next.

RELATED LINKS:

Subform properties in the Subform tab

To specify how a subform manages content

To specify how to merge data between subforms

8.7. To specify how to merge data between subforms

1) In the Object palette, click the Pagination tab, and then click the arrow to the right of the After box and select one of these options:
   - Continue Filling Parent
   - Go to Next Content Area
   - Go To Content Area > *[name _of_content_area]*
   - Go To Next Page
   - Go To Page > *[name_of_page]*
   - Go to Next Odd Page
   - Go to Next Even Page

RELATED LINKS:

Subform properties in the Subform tab

To specify how a subform manages content

To position subforms

8.8. To define custom data-binding properties for a subform

Using the Binding options, you can build a form that captures data for enterprise infrastructures or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

Subforms themselves do not capture or display data, but the objects in a subform can. The binding settings of a subform have a direct influence on how nested objects are mapped to data.

A subform can be bound to a data group, and the subform's objects can be bound to data values within that data group. By default, the bindings of the objects in a subform are relative to the subform's binding.

1) Select the subform.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the subform and its objects to their corresponding data nodes. For information about how to bind objects to a data source, see *Binding fields to a data source*.

RELATED LINKS:

Using subforms

Subform properties in the Binding tab

To name a subform

To create a repeating subform

8.9. Using subform sets

A subform set is a grouping of two or more repeating or multipage subforms.

Use a subform set to control the order in which the subforms are rendered within the set, based on the order that the subforms appear within the set.

For example, in the sample Purchase Order form, you can see how the detailHeader and detail subforms are grouped into a subform set. The subform set keeps the two subforms together so that both subforms for each data item are rendered, or neither subform is rendered if no data exists for the detail subform.

By grouping subforms into a subform set, you ensure that subforms will not rendered within the set unless data exists to render the subform.

You can control the order in which subforms are rendered within a set by selecting one of these options from the Type list of the Subform Set tab in the Object palette:

- Use All Subforms in Order

  All of the subforms in the subform set will appear in the final document and in the order they appear in the Hierarchy palette.
- Select One Subform From Alternatives

  One subform from the set appears in the final document. For more information, see Using choice subform sets.

When choosing how to render subforms, consider the following facts:

- If you set a subform to a minimum count of zero, the subform is not rendered if no data exists for that subform.
- If you set a subform to a minimum count value less than the number of actual occurrences in the data, the remaining data values will not be displayed. Conversely, if you set a subform to a maximum count value greater than the number of actual occurrences in the data, the specified number of subforms are rendered, which means that some objects in the subform will render without data and appear blank.
- You can, if you want, nest subform sets within other subform sets to any number of levels. However, you cannot place individual objects such as lines, circles, or text field objects inside a subform set.

To insert a subform set

A subform set is a grouping of two or more subforms that you want to keep together. You can use subform sets to combine subforms and control the order in which they are rendered.

Before you create a subform set, it is a good practice to consider the order in which you want the form to be rendered so that you can select the correct option from the Type list in the Subform tab.

1) Right-click in the Hierarchy palette and select Insert Subform Set. An untitled subform set is added to the Hierarchy palette.
2) In the Hierarchy palette, drag the subforms you want into the subform set.
3) With the subform set selected in the Hierarchy palette, in the Object palette, click the Subform Set tab and, in the Type list select the option you want.
4) Click the Binding tab. Optionally, you can type a name for the subform set in the Name box.
5) If you want the subform set to repeat according to the data provided, select Repeat Subform For Each Data Item, and then specify a minimum and maximum count, if necessary.
6) Preview the form design to test the subform set.

RELATED LINKS:

Subform set properties in the Subform Set tab

Subform set properties in the Pagination tab

Subform set properties in the Binding tab

8.10. Creating repeating and multipage subforms

You can place a subform anywhere on a page; however, when a large amount of data is merged with the form, the subform may expand beyond the bottom edge of the rendered page. To deal with this issue, you can either select the Allow Page Breaks Within Content option to cause the server to break the subform in the middle or deselect this option to force the entire subform to the top of the next rendered page. Forcing a subform to the next page ensures that any nested subforms are displayed together on the same page. If you do not select the Allow Page Breaks Within Content option, you will probably want to create an overflow leader and overflow trailer subform. (See *About overflow leaders and trailers*.)

When data is merged, the server positions the content of a subform within the confines of content areas only. In cases where a large amount of data needs to be merged with any of the objects in a subform, a subform may span multiple form pages. When a form page is filled, the server automatically renders another identical page and continues to place the subform on the new page unless you explicitly direct the flow to a different content area or master page. When more than one content area or master page exists in the same form, you can specify whether a subform will be placed in a particular content area or positioned according to the specified master page.

To create a repeating subform

1) In the Hierarchy palette, select the parent subform of the subform you want to repeat. (See *About subforms*.)
2) In the Object palette, click the Subform tab, and in the Content list, select Flowed.
3) Select the subform to repeat.
4) In the Object palette, click the Subform tab and, in the Content list, select either Positioned or Flowed.
5) Click the Binding tab and select Repeat Subform For Each Data Item.
6) To specify the minimum number of repetitions, select Min Count and type a number in the associated box. If this option is set to 0 and no data is provided for the objects in the subform at data-merge time, the subform is not placed when the form is rendered.
7) To specify the maximum number of subform repetitions, select Max and type a number in the associated box. If you do not specify a value in the Max box, the number of subform repetitions will be unlimited.
8) To specify a set number of subform repetitions, regardless of the quantity of data, select Initial Count and type a number in the associated box. If you select this option and either no data is available or fewer data entries exist than the specified Initial Count value, empty instances of the subform are still placed on the form.

   NOTE: *The value in the Initial Count box must be between the Min Count and Max values. If the Min Count value is not specified or is 0, the Initial Count value defaults to 0.*

To specify a subform to span multiple pages

Subforms can be placed anywhere in a form design. However, when a large amount of data is merged, the subform may expand beyond the bottom edge of the rendered page.

To deal with this issue, you can either enable a page break in the middle of the subform or force the entire subform to be rendered starting at the top of the next rendered page. Forcing a subform to the next page ensures that any nested subforms are displayed together on the same page.

When data is merged, the content of a subform is placed within the confines of content areas only. In cases where a relatively large amount of data needs to be merged with any of the objects in a subform, a subform may span multiple form pages.

When a form page is filled, an additional identical page is rendered automatically. The subform continues to be placed on the new page unless you explicitly direct the flow to a different content area or master

page. When more than one content area or master page exists in the same form, you can specify whether a subform will be placed in a particular content area or positioned according to the specified master page.

To enable a subform to span multiple pages

1) In the Object palette, click the Subform tab and select Allow Page Breaks Within Content.
   When this option is deselected, the objects in the subform are kept together on the same page.

About overflow leaders and trailers

Because the length of a form that has a flexible layout varies depending on the amount of data merged with the form, forms that have a flexible layout are often longer than one page. Using overflow leader and overflow trailer subforms is an effective way to start and finish subforms that repeat over multiple pages. You can use any subform that is configured to position content as an overflow leader or trailer. For example, in the Purchase Order sample form, the detail subform is configured to position content and repeat for every data item. When the form is merged with data, if the first page has insufficient space to display all occurrences of the detail subform, a new page is added and the data continues to flow into the next content area.

When data flows this way across multiple pages, you may want to carry forward onto each new page certain text such as “Continued from previous page” or specific formatting elements such as a column header row that includes the descriptive labels for each column of data. Using column header rows on each subsequent page makes the resulting form much easier for users to follow. To do this, you can create an overflow leader subform that will act as the column header row for each additional page. In the Purchase Order sample form, for example, the detailHeader subform is selected as the overflow leader. As a result, a copy of the detailHeader subform is rendered at the top of every new page before the first occurrence of the detail subform.

Similarly, you may want to include information following the last occurrence of the repeating subform, at the bottom of all pages except the last page. For example, you may want to include text such as “Continued on next page” at the bottom of these intervening pages. To do this, you can create an overflow trailer subform for the repeating subform in the same way that you created an overflow leader subform.

When a subform overflows to the new page, the server performs these operations:

- Places the overflow trailer on the current page
- Places the overflow leader on the next page
- Flows the expanding subform and the remainder of its repeating objects onto the new page

An *overflow leader* is a special type of positioned subform that appears at the top of the next page whenever a page overflow occurs.

Overflow leaders are similar to the heading row in a standard table. The heading row appears at the top of the table and contains a descriptive label for each of the columns in the table. You can format the table so that when it expands beyond one page, the heading row is repeated at the top of the new page. This makes the information in the table easier to understand as the reader moves from page to page.

An overflow leader subform behaves in a similar way. When you specify that a specific subform will be the overflow leader for a subform that repeats, the overflow leader subform will appear once before the repeating subform at the top of the current page and each subsequent page thereafter.

The repeating subform is added as many times as necessary when merged with data. When there is no more room on the first page, a new page is added and the data continues to flow into the next page until all the data is consumed. The overflow leader subform will appear once at the top of each page.

You can see an example of how an overflow leader subform is used in the sample form design whose layout adjusts to accommodate data that is included with Designer. The sample, Purchase Order.xdp, is in the Samples folder where Designer is installed on your system. In that example, the subform named detailHeader acts as the overflow leader for the repeating subform named detail.

Bookend leaders are subforms that appear before a repeating subform. If you define a subform sibling just above a repeating subform and then specify it as an overflow leader, you have defined it as a bookend leader and as an overflow leader.

An overflow trailer appears at the bottom of the next page whenever a page overflow occurs. Use an overflow trailer to include information that appears only once, after all the data is positioned.

Bookend trailers are subforms that appear just below a repeating subform. If you define a subform just below a repeating subform and then specify it as an overflow trailer, you have defined it as a bookend trailer and as an overflow trailer.

To create and assign overflow leaders and trailers

An overflow leader or overflow trailer is a positioned subform that can be assigned to any repeating subform. Typically, you create the subform that will become overflow leader or trailer first. Then you assign them to a repeating subform and give them overflow properties.

If the overflow leader is also treated as a bookend leader, it is placed in the Hierarchy palette just above the subform. Also, if the overflow trailer is also treated as a bookend trailer, it is placed in the Hierarchy palette just below the subform.

To create an overflow leader or overflow trailer subform

1) Create a form design with all the necessary subforms.
2) On your form design, locate the subform that you want to use as an overflow leader or an overflow trailer, select the subform, and do these tasks:
   - Look at the Subform tab in the Object palette. Ensure that the Content list displays Positioned. In the Object palette, click the Subform tab and make sure that Positioned is selected in the Content list.
   - Look at the Binding Tab in the Object palette. In the Object palette, click Binding tab and select Repeat Subform For Each Data Item to have the leaders and trailers appear for more than one overflow. Bookend leader and trailers are not counted against this number.

To apply an overflow leader or overflow trailer to a subform

1) Select the subform or subforms that you want to repeat.
2) In the Object palette, click the Binding tab and select Repeat Subform For Each Data Item to set the subform to repeat.
3) In the Object palette, click the Pagination tab and, in the Overflow box, specify either a content area or page where the form should place the overflow subforms."]
4) If you want to specify an overflow leader, in the Overflow Leader list, select the subform that you want to use as the overflow leader subform for the current repeating subform. Alternatively, select New from the Overflow Leader list to create and assign a new overflow leader subform.
5) If you want to specify an overflow trailer, in the Overflow Trailer list, select the subform that you want to use as the overflow trailer subform for the current repeating subform. Alternatively, select New from the Overflow Trailer list to create and assign a new overflow trailer subform.

   NOTE: *You do not need to have both an overflow leader and an overflow trailer. Whether you have one, both, or none is determined entirely by the requirements of your form.*

RELATED LINKS:

Working with forms that have a flowable layout

Use overflow leader and overflow trailer subforms

Using subforms

Subform properties in the Subform tab

Subform properties in the Pagination tab

Subform properties in the Binding tab

8.11. To control subform and subform set breaks by using conditional statements

Designer provides the capability to create customized conditional breaks for repeating subform objects. As opposed to paginating in response to data overflow, conditional breaks allow you to manually control how a subform breaks in a form based on a series of checks called *conditional statements*.

Through the conditional statements, you can verify data for a field within a repeating subform against previous instances of that field. The repeating subform can then be broken in response to a change in the data supplied to the field.

For example, on a telephone bill, you could break a repeating subform in response to changes in the field that stores the date of each billing entry. The telephone bill could then be visually broken down by date, making it easier for a user to read.

In addition to specifying a breaking condition, you can also specify a leading or trailing subform and indicate where to place the next instance of the repeating subform in the form.

1) Select a subform or subform set.
2) In the Object palette, click the Pagination tab.
3) Click Edit and then click Add to insert a new conditional break list item.

4) Select a scripting language from the Language list. The conditional break condition statement is created using the scripting language you select.
5) In the Run At list, select where you want the conditional break to execute.
6) Click Insert Sample Expression and select the form design object within the subform to use as the comparison field for the conditional break. Alternatively, you can enter your own conditional statement in the field. To correctly evaluate as a conditional break, however, any user-defined conditional statements must evaluate to either `true` or `false`. Conditional statements that evaluate to `true` are executed.
7) Use one of these options to specify when you want the subform object to break:
   – Select Before to insert a break immediately before the current instance of the subform is inserted into the form.
   – Select After to insert a break immediately after the current instance of the subform is inserted into the form.
8) In the To field, select where you want to place the remaining occurrences of the broken subform.
9) In the Trailer and Leader lists, select trailer and leader subforms to use for the current conditional break, if any.
10) Repeat steps 2 to 8 for each conditional break that you want to include for the selected subform.
11) Click OK when you have finished adding entries to the list.

    After you create all of your conditional break entries, you should review the order in which they appear in the Edit Conditional Breaks dialog box. Designer processes the conditional breaks specified in the Edit Conditional Breaks dialog box in sequential order from top to bottom. Each conditional break for which the conditional statement evaluates to `true` is executed.

Use the Up and Down buttons to move individual conditional break list entries into the preferred order.

RELATED LINKS:

Edit Conditional Breaks dialog box

Subform properties in the Pagination tab

8.12. Using choice subform sets

A *choice subform set* is a variation of the subform set object that allows you to customize the display of specific subforms from within the set through the use of conditional statements. As with subforms sets, choice subform sets are bound to data from a data connection. However, using choice subform sets provides a greater level of control over which subforms within the set are displayed at run time. For example, you can configure a choice subform set to display instances of a repeating subform by using different text colors, depending on the value of a specific field within that repeating subform.

To create choice subform sets

You can create a choice subform set from either an existing subform set or from an existing subform object.

To create a choice subform set from an existing subform set

1) Select a subform set object.
2) In the Object palette, click the Subform Set tab and select the Select One Subform from Alternatives option from the Type list.

To create a choice subform set from a subform object

1) Select a subform object and select Insert > Wrap in Choice Subform Set.

To add and remove subforms in choice subform sets

You can add subform objects to a choice subform set or remove subform objects from a choice subform set in using either of these methods:

- Using the Edit Data Nominated Subforms dialog box
- Manually editing the contents by dragging subform objects into or out of the choice subform set by using the Hierarchy palette.

  IMPORTANT: *Removing a subform from the Alternative Subforms list by using the Edit Data Nominated Subforms dialog box completely removes the subform object from your form design. If you want to preserve the subform object but remove it from the choice subform set, you must manually drag the subform object out of the choice subform set by using the Hierarchy palette.*

To add a subform to a choice subform set

1) Select a subform set object.
2) In the Object palette, click the Subform Set tab and click Edit Alternatives.
3) Click the Add button to insert a new Alternative Subforms list item. You can also drag and drop subform objects into the choice subform set by using the Hierarchy palette.

   Adding a new subform adds a duplicate of the currently selected subform. If no subform is selected, a new unnamed subform is added to the choice subform set.
4) Use the Up and Down buttons to move the new subform entry to the preferred location. Ordering entries in the Alternative Subforms list is important because entries are processed sequentially at run time, and the first entry with an expression that evaluates to `true` appears in the form.

To remove a subform from a choice subform set

1) Select a subform set object.
2) In the Object palette, click the Subform Set tab and click Edit Alternatives.
3) Select an entry from the Alternative Subforms list and click the Delete button .

RELATED LINKS:

Using subform sets

8.13. Working with forms that have a flowable layout

A form that has a flowable layout contains subforms and other elements that adjust to accommodate the amount of data available to fill the form. The form can be interactive, which means that users can fill the form, or non-interactive, where a server-based process merges data into the form. (See *Form design layouts*.)

Master pages, content areas, and subforms are the elements that control how Designer places objects in the form and adjusts to display varying amounts of data.

Differences at design time and run time

In a flowable layout, keep in mind that what you see at design time is not what users see when the form is rendered. For example, a form design may contain one item row to enter data in. However, when the form is rendered on the client, the form may contain several item rows and users may be able to insert additional item rows. The number of rows that appear at design time depends on whether you wrapped the objects in the item row in a subform that uses the Min Count or Max options. Setting up the subform Min Count or Max options controls the number of rows that are initially available for users to fill and the number of additional rows that users can later add.

Because a form that has a flowable layout adjusts automatically to accommodate data, you do not have to set the size of objects or the number of item lines that the form requires. For example, by selecting options such as Allow Multiple Lines, Allow Page Breaks Within Content, and Expand To Fit, you can design flowable interactive forms that adjust to an undetermined amount of data.

For an example of the differences between the design and run-time views of non-interactive forms, see *How non-interactive forms that have a flowable layout work*.

RELATED LINKS:

Building actions in forms

Controlling the placement of objects in forms that have a flowable layout

Creating form designs that have a flowable layout

Creating non-interactive forms that have a flowable layout

Design tips for forms that have flowable layout

Guidelines for PDF forms

Controlling the placement of objects in forms that have a flowable layout

You can use either of two ways to control how Designer places objects in a form:

Content areas

On the master pages, content areas control the areas and flow direction in which Designer places objects on the pages. Every master page has a default content area.

Subforms

Subforms control how Designer places objects in the form. By wrapping objects into subforms, you can use the options in the Subform and Binding tabs in the Object palette to regulate how many times objects are rendered and their placement on the page.

Content areas in a flowable layout

All master pages contain a default content area, which is displayed on the associated pages. The content area outlines the area in which you can place objects on the pages, much like the margins in a text document. You cannot place objects on the pages outside the area represented by the content area.

TIP: *Every form must have at least one content area.*

The content area does not delineate or limit the area in which you can position objects on the master pages. In fact, you can place boilerplate objects anywhere on the master pages, inside or outside the content area. If you place objects outside the content area to contain header and footer text, the date and time, or page numbering, test the form to ensure that the objects appear as intended. You want to make sure that objects on the pages do not overlap and hide the objects on the master pages when the form is rendered.

With forms that have sections that adjust to accommodate data, remember that the layout of the form is ultimately data-driven. If the form is being rendered through Forms or Output, the pages are added until all the available data is merged. As new pages are added, data flows from page to page within the area defined by the content area on the master pages.

If you want the same layout for each page in your form, the default content area is all you need. However, if the layout is more involved, you can add additional content areas to the default master page. Keep in mind that if you want different flow direction, page orientations, or two-sided pages, you can insert additional master pages and configure the other master pages and content areas to suit your design.

TIP: *Remember that you cannot work with content areas on the pages.*

Subforms in a flowable layout

Planning subforms is an important part of creating a form with a flowable layout. In a form that contains flowable sections, subforms contain the objects within each section of the form. They maintain the form layout as it adjusts in length according to the amount of data merged into it. Subforms also ensure that the objects move as a unit and are placed consistently relative to each other. When you wrap objects in subforms, you ensure that regardless of the amount of data merged with the form, the format of each section remains consistent, and the layout of the entire form is predictable from one rendering to the next.

You can nest subforms to any depth. By nesting them, you can make the form design match the XML data file that is merged with it. For example, if the XML data file includes two levels, you can create the subform hierarchy by matching the subform names with the tag names in the XML data file so that Designer can map the data without transforming the data file.

A form that has a flowable layout, such as the Purchase Order sample, typically includes various subforms nested together, with one or more objects in each subform. For example, you can see how the various objects in the Purchase Order sample are wrapped in the subforms named *header*, *detailHeader*, *detail*, and *total*. Notice how you can align text and text field objects into rows and columns and wrap them in

a subform. Using subforms to wrap groups of objects such as the various text field objects in the detail subform not only lets you maintain the layout of the objects, but also lets you control how often the subform is repeated.

You can configure each subform so that when the form is rendered, only the subforms that contain the objects that are necessary for representing the data content are placed. For example, you may want to produce different purchase orders for each customer. In one purchase order form, the detail subform is placed 12 times to indicate the items purchased. In another purchase order form, the same subform is placed 50 times, spanning multiple pages.

The sample Purchase Order form illustrates how you can use subforms to achieve this configuration. Open the form in Designer and look at the Hierarchy palette to examine the structure of the subform. Look at the Object palette to see how the subforms are configured. Pay particular attention to the purchaseOrder, detailHeader, and detail subforms and how they are configured. Keep in mind that the parent subform, which is purchaseOrder, controls the flow of content in the form and expands to fit the available data. The detailHeader subform is configured with a minimum count of 1, which means that when the form is rendered, this subform appears only once. However, the detail subform is configured to repeat with each data item and therefore, when the form is rendered, appears as many times as there are occurrences of the data.

You can also use a subform to draw a border around a group of subforms. For example, you can create one or more dynamic subforms, nest those subforms within a wrapper subform, and then specify a border for the wrapper subform. When the form is rendered, a border is drawn around all the internal subforms.

Creating form designs that have a flowable layout

There are two basic approaches to creating a form design containing subforms that adjust to accommodate data:

- Starting the form design with a fixed layout
- Creating the form design in a flowable layout

Both methods produce the same results. However, you may find that one suits your design style more than the other. As you become more familiar with creating form designs that have a flowable layout, you can adopt a style that works best for you.

Start with a fixed layout

If you are familiar with creating forms that have a fixed layout, you may find that starting your design with a fixed layout is most comfortable and familiar. You can plan the layout of the form, including the necessary objects and information that you want users to see. Using a fixed layout, build the form design by using as many master pages and design pages as are required. When the object layout and formatting are completed, set the appropriate properties to those areas of the form design that flow content.

In addition, this approach provides the easiest and most straightforward way of converting an existing form that has a fixed layout to a form design that has a flexible layout.

The following steps provide a general idea of the design process for creating a form design starting with a fixed layout, assuming that the planning stage is completed.

1) Create the required master pages.
2) On the page, create the form content. If you are converting a form that has a fixed layout, simply begin by wrapping the objects on the form into subforms. If you are starting with a blank form, do the following tasks:
   – Add objects to the form and enclose them in the appropriate subforms.
   – Apply the necessary formatting to the objects.
   – Finalize the layout of the objects within their subforms, and subforms on the page.
   – Add scripting if necessary.
3) Implement the dynamic concepts:
   – Unwrap the default subforms on each of the pages. The subforms become children of the root subform, form1, which has flowed content. From this point on, you are designing in a flowable layout. Notice that the subforms are positioned one below the other because the flow direction is top to bottom.
   – Apply the necessary formatting to the subforms, such as subform binding type and borders.
   – Define the flow. Set the occurrence values, page breaks, leader and trailer subforms, define which subforms to keep together, associate subforms to master pages, and set margins.
4) Test the form by using sample data.

   NOTE: *The sample forms included with Designer follow this procedure. For usability reasons, the subforms that contain the actual content are wrapped inside a subform that has its binding type set to none. By configuring the subforms in this way, you can quickly understand the overall structure of a form when you see it in Hierarchy View for the first time.*

Tips for starting with a fixed layout

When you are creating form designs by starting with a fixed layout, keep in mind the following points:

- Set the form' content to flowed only when you are satisfied with the layout. When the form content is set to flowed, you can revert to a positioned content, but you may experience problems with the form layout. For example, if you choose to wrap the subforms inside a single positioned container, the subforms are positioned exactly where they were on their pages and may overlap. To avoid the overlapping, wrap the subforms by using page grouping.
- The default minimum and maximum count for subforms is 1. Adjust these values for forms with a flowable layout.
- Wrapping objects inside a subform discards any extra space defined beyond the objects. Add left and right margins to realign the subform horizontally. You can resize the subform, but doing so repositions the objects because they are positioned relatively to the parent.

RELATED LINKS:

Controlling the placement of objects in forms that have a flowable layout

Using content areas

Guidelines for PDF forms

Using subforms

Subform properties in the Subform tab

Creating interactive forms that have a flowable layout

Using Designer, you can create interactive PDF forms for data capture that have a flowable layout. This type of form combines both flowable elements and interactive form functionality so that users can enter data directly into expandable fields, add or remove sections, and return the form data electronically. For example, you can create an interactive form where users can add or remove item rows and can choose to add their comments to the form. (See *Form design layouts*.)

To fill interactive PDF forms that have a flowable layout, end users need Acrobat 7.0.5 or Adobe Reader 7.0.5 or later.

To fill interactive HTML forms that have a flowable layout, end users require a client application such as a web browser. (See *CreatingHTMLforms*.)

NOTE: Users cannot use Acrobat commenting features when filling interactive forms that have a flowable layout.

Keep in mind that you can also use the Action Builder dialog box on the Tools menu to build common interactive capabilities in forms that have a flowable layout, without writing scripts. (See *Build an action*.)

When creating interactive forms that have a flowable layout, many form authors prefer to start with an interactive form design that has a fixed layout. By starting in this manner, you can see the form in its final state. The form design provides the final layout including all the elements that users need.

After completing the form design in fixed layout, you can then begin to divide it into its essential parts, identify the subforms, and specify how they will flow the data being entered into the form.

Adding dynamic capabilities to an interactive form typically involves the following techniques:

- *Position the header and footer on the master page*
- *Wrap form objects in subforms and setting occurrence values*
- *Modify margins to accommodate subforms*
- *Use overflow leader and overflow trailer subforms*
- *Create subforms with flowable elements*
- *Create a button to add and remove a section*
- *Create add and delete buttons*

In addition, you must understand how to add and remove subforms so that the form can respond to the amount of data a user enters. (See *Add and remove subform instances using scripting* and *Add and remove tables and table rows by using scripting*.)

RELATED LINKS:

Guidelines for PDF forms

Working with forms that have a flowable layout

About forms

Creating HTML forms

Position the header and footer on the master page

In an interactive form that has a fixed layout, header and footer text is positioned on the page. When you add subforms that are set to flow content in the form, you typically want header and footer text on the master page.

Because the length of an interactive form containing subforms that are set to flow content can expand depending on the amount of data the form contains, additional pages may be added to the form as the end user enters more data. By placing the header and footer on the master page, you ensure that they appear on each page instead of only on the first page.

RELATED LINKS:

Plan the form design

Wrap form objects in subforms and setting occurrence values

When designing an interactive form that has a fixed layout, must provide a sufficient number of item lines for users to fill. However, if you wrap the item lines in subforms that are set to flow content, users can add item lines as they need them.

Because of the dynamic capabilities of subforms, you can include one or two item lines in the form along with Add and Delete buttons. By wrapping the objects in a subform, you can determine when, where, and how many times particular subforms are repeated by using the options on the Subform and Binding tabs of the Object palette.

Wrapping groups of form objects in subforms is an important part of creating a form with dynamic functionality.

Subforms contain and control the objects arranged within sections of the form to maintain the form's layout as it adjusts in length according to the amount of incoming data. Subforms preserve the x and y coordinates of each object to ensure that the objects move as a unit and are placed together.

When you wrap groups of objects in subforms, you ensure that, regardless of the amount of data contained in the form, the format of each section and the layout of the entire form remains consistent and predictable.

Remember that end users must see at least one item row to enter data when they open the form. That means that you must set the Min Count option for the subform to at least 1. Depending on the form, initially displaying two or more rows in the PDF form can give the form a more balanced appearance, as well as result in a more intuitive and functional end-user experience.

Subforms also provide the dynamic capabilities that let users add or remove rows and sections in the form. For example, you can include various buttons that users can use to add and remove item rows or to submit comments.

Plan how you want to wrap objects in subforms to create sections in your form design. Then test the form design to ensure that the subforms move and repeat as you intended and do not break across pages in ways not intended.

RELATED LINKS:

Using subforms

Guidelines for forms

Creating and configuring subforms

Create a button to add and remove a section

Modify margins to accommodate subforms

When applying dynamic functionality to an interactive form that has a fixed layout, a certain amount of reformatting is typically needed. Using the options in the Layout palette, you can set the size and position, margins, and captions of objects. For example, you may need to set margins around subform objects to regulate the alignment of and the spacing (white space) between the sections in the form.

RELATED LINKS:

Layout properties in the Layout palette

Use overflow leader and overflow trailer subforms

Because the length of an interactive form containing subforms that are set to flow content can vary depending on the amount of data displayed in the form, the form often has more than one page. As a result, you may want certain subforms (sections) in the form to continue on subsequent pages.

Using overflow leader and overflow trailer subforms, you can designate different subforms to precede and follow repeating or expanding subforms that are likely to carry over to additional pages. Overflow leaders and trailers are special types of subforms that you can use in form designs that have a flowable layout. After you specify a subform as an overflow leader subform in the Binding tab of the Object palette, the subform appears at the top of each new page.

For example, you can use the heading row at the top of a table as an overflow leader subform. Placing the heading row at the top of a table ensures that it is repeated at the top of each new page, making the information in the table rows easier to read and follow from page to page.

Similarly, you can assign an overflow trailer subform to follow the last occurrence on a page of a repeating or expanding subform that could possibly span multiple pages.

RELATED LINKS:

Using subforms

Creating and configuring subforms

Create subforms with flowable elements

Depending on the purpose of your form design, you may want to add an adjustable section to the form where users can type information such as special requests or comments. With Designer, you can apply different options to text fields and subforms that enable the size of sections in the form to adjust to the amount of data that the user enters.

For example, your interactive form may contain a Comments section that users can display or hide by clicking a button. You would create the section by using a text field placed inside a subform with these options selected to enable dynamic data entry:

Text field

Allow Multiple Lines and Plain Text Only (Field tab), User Entered - Optional (Value tab), Height - Expand to fit (Layout palette)

Enables users to enter as much text as they want in the Comments section and ensures that the text field automatically expands in height to display the data available.

Subform

Allow Page Breaks Within Content (Subform tab), Max (count) with a value of 1 (Binding tab), Auto-fit (Layout palette)

Allows the comments subform to carry over to succeeding pages when necessary and to appear only once in the form and ensures that the subform automatically enlarges to display the data available.

RELATED LINKS:

Using subforms

Creating and configuring subforms

Create a button to add and remove a section

Create a button to add and remove a section

In interactive form designs, it is common practice to have one or more sections in the form that are not displayed until the user selects the option to include it. With Designer, you can add a button along with a script that dynamically adds or removes a particular section (subform) from the form when the user clicks a button.

For example, in the sample interactive Purchase Order, the user can click the Add Comments button to display the Comments section (comments subform). The button has one of two alternating captions, Add Comments or Clear Comments, depending on the current state of the subform. Each time the user clicks the button, the script checks whether the comments subform is displayed and then updates the button caption accordingly.

The button triggers a script that uses `instanceManager`, the XML Form Object Model object that manages the instance creation, removal, and movement of form objects. When the end user deletes the Comment subform, the `instanceManager` object removes the subform from both the Form data document object model (DOM) and the Data DOM.

Note that `instanceManager` uses four methods: `addInstance`, `removeInstance`, `moveInstance`, and `setInstances`. The naming convention of an `instanceManager` is the subform name prefixed with an underscore (`_subformname`). The syntax for `instanceManager` is `_subformname.methodname()`.

In the sample interactive Purchase Order form, the form author typed the following JavaScript script in the Script Editor by using the `setInstances` method to add and remove the comments subform and change the button's caption. Notice that the `comments.count == 0 property` returns the number of subform instances instantiated.

```
// Invoke the Instance Manager to add and remove the comments subform.

if (_comments.count == 0) {// The count property specifies the current
number
// of instances instantiated.
_comments.setInstances(1);// Add the comments subform.
this.resolveNode("caption.value.#text").value = "Clear Comments";//
Change the button's caption.
}

else {
_comments.setInstances(0);// Remove the comments subform.
this.resolveNode("caption.value.#text").value = "Add Comments";// Change
the button's caption.
}
```

TIP: *You can also use the ActionBuilder dialog box on the Tools menu to build common interactive capabilities in forms that have a flowable layout, without writing scripts.*

RELATED LINKS:

Building actions in forms

Create add and delete buttons

Adding buttons to an interactive form provides end users with a way to initiate an action, such as adding and deleting instances of subforms that define sections such as item lines in an order form.

You can also add a tool tip to the delete button to display the words “Delete Item” when the user positions the pointer over the button. Using separate delete buttons is a good way to allow users to remove specific rows from the form.

For example, the sample interactive Purchase Order has an Add Item button and a delete button labeled “X” for each detail subform. When the user clicks the Add Item button, a script adds an item line. Alternatively, when the user clicks one of the delete buttons, a script deletes the associated item line.

TIP: You can also use the Action Builder dialog box on the Tools menu to build common interactive capabilities in forms that have a flowable layout, without writing scripts. See *Building actions in forms*.

Add Item button

The following JavaScript script in the `click` event of the Add Item button lets users add an item line to the sample interactive Purchase Order form. The script also recalculates the form so that the Total field includes the new line in the calculation.

Because users only add item lines by using the Add Item button, the script does not need to verify the minimum count (occurrence) value.

```
// Invoke the Instance Manager to add one instance of the detail subform.
_detail.addInstance(1);

//Invoke the recalculate method to include the field values from the added
subform in calculations.
xfa.form.recalculate(1);
```

Delete button

The following JavaScript script in the `click` event of the Delete button lets users use the Delete button to delete an instance of the detail subform from the sample interactive Purchase Order form. The script also recalculates the form so that the Total field no longer includes the deleted line in the calculation.

```
// Invoke the Instance Manager to remove the current instance of the detail
subform.
_detail.removeInstance(this.parent.index);

// Invoke the recalculate method to update the form calculations.
xfa.form.recalculate(1);
```

Because the initial minimum occurrence value for the detail subform is 2, the script needs to reduce the minimum occurrence value to allow the person filling the form to delete the two item lines that appear automatically when the form is rendered. This script is added to the `initialize` event of the detail subform.

```
// Reset the minimum occurrence value of the detail subform.
this.occur.min = "0";
```

Because the form allows users to delete all instances of the detail subform, the script for the `calculate` event of the Total field (`numTotal`) must verify that at least one instance of the `numAmount` field in the detail subform exists. Otherwise, an error appears because the calculation cannot find any occurrences of the `numAmount` field. This script is added to the `Calculate` event of the `numTotal` field.

```
// Verify at least one instance of the numAmount field exists.
if (exists(detail[0].numAmount) == 1) then
Sum(detail[*].numAmount)
endif
```

Add and remove subform instances using scripting

You can add or remove subform instances by adding calculations or scripts to objects on your form. It is important to name the subforms when adding and removing subforms instances though scripting. Scripting against existing subforms is error prone when the subforms are accessed through a relative position instead of unique names.

Before you begin, perform the following tasks:

- If it is not already visible, display the Script Editor by selecting Windows > Script Editor. Expand the Script Editor so that it is displayed in multiline view.
- Ensure that the subform that you want to add instances to is contained within a flowed subform.

  TIP: You can also use the Action Builder dialog box on the Tools menu to build common interactive capabilities in forms that have a flowable layout, without writing scripts. See *Building actions in forms*.

RELATED LINKS:

Building actions in forms

Add and remove tables and table rows by using scripting

Subform properties in the Binding tab

Using the methods of the instancemanager to controlsubforms
addInstance
removeInstance

To add a subform instance to a form

1) Select the object on your form design that you want to add the script to and select an event from the Show list in the Script Editor. For example, choose the `click` event of a button or other interactive form object.
2) Type one of the following scripts in the Script Editor, where *Subform1* is the name of the subform to add instances to, and the value `true` indicates that the new subform instance should be merged with the form data:

FormCalc

```
Subform1.instanceManager.addInstance(true) // Default instance manager
syntax
xfa.form.recalculate(true) // Invoke the recalculate method to include the
field values from the added subform in the form calculations.
```

or

```
_Subform1.addInstance(true) // Short form of the instance manager syntax
xfa.form.recalculate(true) // Invoke the recalculate method to include the
field values from the added subform in the form calculations.
```

JavaScript

```
Subform1.instanceManager.addInstance(true); // Default instance manager
syntax
xfa.form.recalculate(true) // Invoke the recalculate method to include the
field values from the added subform in the form calculations.
```

or

```
_Subform1.addInstance(true); // Short form of the instance manager syntax
xfa.form.recalculate(true) // Invoke the recalculate method to include the
field values from the added subform in the form calculations.
```

To remove a subform instance from a form

1) Select the object on your form design to add the script to, and select an event from the Show list in the Script Editor.
2) Type one of the following scripts in the Script Editor, where `Subform1` is the name of the subform to remove an instance from and *integer* is the zero-based index number of the instance to remove:

FormCalc

```
Subform1.instanceManager.removeInstance(integer) // Default instance
manager syntax
```

or

```
_Subform1.removeInstance(integer) // Short form of the instance manager
syntax
```

JavaScript

```
Subform1.instanceManager.removeInstance(integer); // Default instance
manager syntax
```

or

```
_Subform1.removeInstance(integer); // Short form of the instance manager
syntax
```

Add and remove tables and table rows by using scripting

You can add or remove tables, body rows, header rows, and footer rows by adding instance manager scripting expressions to objects on your form by using the Script Editor.

Before you begin, make sure that you perform the following tasks:

- If it is not already visible, display the Script Editor by selecting Windows > Script Editor. Expand the Script Editor so that it is displayed in multiline view.
- Ensure that the table to add instances to is contained within a flowed subform so that you can add new instances of either the table or the rows within the table.

To add a table or table row instance to a form

1) Select the object on your form design to add the script to, and select an event from the Show list in the Script Editor.
2) Type one of the following scripts in the Script Editor, where `Table1` and `Row1` are the names of the table and table row to add instances to:

FormCalc

```
Table1.instanceManager.addInstance(true) // Default instance manager
syntax
Table1.Row1.instanceManager.addInstance(true) // Default instance manager
syntax
xfa.form.recalculate(true) // Invoke the recalculate method to include the
field values from the added table or row in the form calculations.
```

or

```
Table1.addInstance(true) // Short form of the instance manager syntax
Table1._Row1.addInstance(true) // Short form of the instance manager
syntax
xfa.form.recalculate(true) // Invoke the recalculate method to include the
field values from the added table or row in the form calculations.
```

JavaScript

```
Table1.instanceManager.addInstance(true); // Default instance manager
syntax
Table1.Row1.instanceManager.addInstance(true); // Default instance
```

```
manager syntax
xfa.form.recalculate(true); // Invoke the recalculate method to include
the field values from the added table or row in the form calculations.
```

or

```
Table1.addInstance(true); // Short form of the instance manager syntax
Table1._Row1.addInstance(true); // Short form of the instance manager
syntax
xfa.form.recalculate(true); // Invoke the recalculate method to include
the field values from the added table or row in the form calculations.
```

To remove a table or table row instance from your form

1) Select the object on your form design to add the script to, and select an event from the Show list in the Script Editor.
2) Type one of the following scripts in the Script Editor, where `Table1` and `Row1` are the names of the table and table row to remove instances from and `integer` is the zero-based index number of the instance to remove:

FormCalc

```
Table1.instanceManager.removeInstance(true) // Default instance manager
syntax
Table1.Row1.instanceManager.removeInstance(true) // Default instance
manager syntax
```

or

```
Table1.removeInstance(true) // Short form of the instance manager syntax
Table1._Row1.removeInstance(true) // Short form of the instance manager
syntax
```

JavaScript

```
Table1.instanceManager.removeInstance(true); // Default instance manager
syntax
Table1.Row1.instanceManager.removeInstance(true); // Default instance
manager syntax
```

or

```
Table1.removeInstance(true); // Short form of the instance manager syntax
Table1._Row1.removeInstance(true); // Short form of the instance manager
syntax
```

RELATED LINKS:

Table properties in the Binding tab

Body row properties in the Binding tab

Header row properties in the Binding tab

Add and remove subform instances using scripting

Creating non-interactive forms that have a flowable layout

When you need a form to present varying amounts of data, typically from a data source, non-interactive forms that are designed to flow content are useful. Just as with interactive forms that are designed to capture varying amounts of user-supplied data, the layout of this type of form adjusts automatically to the amount of data being merged with the form design by a server-based process. You do not need to predetermine the number of sections or pages for the form as you must do with a form that has a fixed layout.

Typically, non-interactive forms are printed and/or stored electronically. For example, you can create a non-interactive form design that has a flowable layout for use with Output. Output can then merge the form design with XML form data and output the form to a network printer, a disk file, and an email recipient as a file attachment. You can output these forms as PDF (including PDF/A documents), PostScript, Printer Control Language (PCL), and Zebra Programming Language (ZPL) formats.

RELATED LINKS:

Using subforms

Design tips for forms that have flowable layout

How non-interactive forms that have a flowable layout work

In a form design that has a fixed layout, you typically use only one subform (the default subform on the page), which Designer sets to position content by default. When Designer positions the content in a subform, none of the objects within the subform move from their anchored positions, regardless of the characteristics and quantity of data.

However, if you want sections of the form to expand to accommodate data, you generally use multiple subforms: the default subform on the page, which you set to flow content, along with additional subforms that you can set to repeat for each data item, expand to fit, or both. When you reset a subform to flow content, the objects within the subform, including other subforms, move to accommodate the data merged into the repeating and expanding subforms.

With Designer, you can bind the objects in the form to the data elements in a data file, and you can configure the text field and subform objects in the form to appear, repeat, or expand, depending on the characteristics and quantity of data merged with the object. Because the objects in the form are bound to the source data, the layout of the form is data-driven.

When authoring a form design that contains sections that expand and shrink to accommodate data, it is important to understand which subforms appear once in the form, such as an address block, and the subforms that repeat according to the amount of data, such as a detail line. For those subforms that repeat, include only one instance of the subform and its components in the form design. Consequently, what you see at design time is not what users see when the form is rendered.

For example, the following illustrations of the non-interactive Purchase Order sample show the form design before it is merged with data and the resulting form that is presented to a user after the form design is merged with data. Notice how the detail line (detail subform) repeats four times to accommodate the list of parts (available data).

Form that has a flowable layout without merged data

Form that has a flowable layout with merged data

Design tips for forms that have flowable layout

The complexity of a form significantly influences the amount of time it takes Adobe Reader to render or reender the document. You can measure the complexity in several ways, but a main factor is the number of visible form objects. Even a complex form design with many objects, subforms, and pages can perform well if a limited part of its complexity is visible at once. It can perform well because the objects that have a presence set to hidden are ignored, and no time is spent to position and render them.

Keep in mind the following points when creating forms with a flowable layout:

General

- Reduce the overall complexity the form design:
  - Use field captions instead of static text objects.
  - Remove captions form a field when not required.
  - Specify borders instead of drawing lines.
  - Specify a subform margin instead of using objects to add spacing between subforms.
  - Remove duplicate objects, scripts, or constructs that you can replace with fragments, script objects, and global fields.
- Use either explicit or implicit breaks:

  - With implicit pagination, the master pages are instantiated in the order they are listed in the hierarchy, based on their minimum and maximum occurrences.
  - Explicit breaks are allowed and often required for complex forms, but are not necessary to create complex forms.

- Start with the layout, ant then add the dynamic behaviors and scripting.
- Use fixed layout when possible; for example, when a container such as a subform has one child.
- Use fixed size objects when possible.
- Resize the content area to leave enough space on the page for other objects that you do not want overlaid (for example, a page number, title, logo, and so on). Watermarks are meant to be overlapped content.
- For a column layout, consider using multiple content areas or tables. To flow content from one column to the other, use content areas. To align the content of each column side by side, use a table. Create the table without a header and footer row.

Subforms

- Avoid placing flowed subforms inside a positioned subform. Doing so causes problems with page breaks, overlapping objects, and repeating subforms.
- If a subform contains objects that merge with data of varying sizes, verify that the objects do not expand and overrun the area that another object occupies. Expandable objects, such as text fields, may render on top of other objects. Set the subform to flow and expand to fit the content.
- When you create a subform, resize it so that its width is the same as the width of the content area. Resize the subform before you place objects in it. This way, you avoid having to reposition the subform's children after you resize the subform.
- Always set up overflow leader and trailer subforms to Positioned content and deselect the Allow Page Breaks option. Otherwise, the rendered form may contain errors such as duplicate headers or overlapping fields.
- Specify an overflow leader and trailer subform for subforms that break between pages.
- To quickly resize a subform to fit around its children, select the Autofit option in the Layout palette and then disable it.
- It is good practice to rename nameless subforms with unique names. Naming subforms makes scripting easier and helps you locate objects in the Hierarchical view. A nameless subform does not participate in the data merge. To achieve the same result with a named subform, set its binding type to None.

  NOTE: *If you create a large interactive PDF form with no structure, end users may experience slow performance when tabbing between fields. This problem is averted if you save the form without tagging. However, if you need your form to be accessible. However, to make your form accessible, save it as tagged PDF. To work around this situation, wrap sections of the form in unnamed subforms. This task adds the required structure to the form.*
- You can have several content areas on a master page. The content areas are filled with content in the order they are listed in the Hierarchy view regardless of their position on the page.
- If you are creating a compliant form, use Acrobat to compare the documents. Select Compare Documents from the Acrobat Advanced menu.

- Setting the margins of a subform may generate unexpected results. Most subforms have positioned content and therefore a fixed height. Adding margins offsets the subform content and can cause the objects below it to overlap.
- To wrap the content of a subform exactly, without using the Expand To Fit option, resize the subform by using the properties in the Layout palette. The sample forms that are included with Designer use both methods.

Accessibility

- Accessibility tags are generated from left to right and then from top to bottom. Accessibility problems can occur in subforms that position content when the objects are not positioned precisely.

Scripting

- When scripting, avoid placing code in the initialize event of objects on the master page. In earlier versions of Adobe Reader, the initialize events for objects on master pages are executed more often than necessary. This extra script execution affects the performance of the form. If possible, use a different event, such as the calculate event.

Data binding

- If you have a data connection, bind objects to the data connection or set the binding type to None. Also, use relative binding references. Do not have reoccurring data that is not bound to reoccurring subforms. It is best to bind fields to a data element and bind subforms to a data group. Data groups and subforms can repeat.
- By default, the subform binding type is set to Use Name. When working with a schema, it is preferable that you set the default binding type to No Data Binding. You can set the default data binding in Tools > Options > Data Binding.
- Set the data binding to No Data Binding for objects you do not want exported in the data. Set the binding type of subforms that contain no fields to No Data Binding. For example, an overflow leader or trailer subform usually has no fields. Setting the binding type to No Data Binding prevents it from participating in the merge. Excluding the overflow leader or trailer from the merge can improve performance.

RELATED LINKS:

Creating form designs that have a flowable layout

Common mistakes

- Not specifying preview data.
- Forgetting that the data entered in the Preview PDF tab is not the form data. When you save the form as a PDF form and open it in Acrobat or Adobe Reader, the data is not displayed.
- Previewing the form as a static Static PDF Form instead of a Dynamic XML Form.
- Forgetting to allow content to break between pages.
- Placing a flowed subform inside a positioned subform.

- Do not know how to set a subform to repeat. The repeat option is disabled when the parent container uses positioned layout.
- Resizing or moving the contents of containers with flowed contents.
- Unnecessary nesting of subforms, especially subforms that have one child container.
- Removing subform instances from Document Object Model (DOM) instead of hiding the subform.
- Not using the Report palette or the JavaScript Debugger in Acrobat (Ctrl+J) to view scripting errors.

9. Using fragments

9.1. About fragments

A *fragment* is a reusable part of a form. For example, a fragment can include an address block or legal text.

A fragment is managed in the Fragment Library and is stored as a separate XDP file that can be inserted into multiple form designs. In the stand-alone version of Designer, fragments are stored in the file system and in the Fragment Library. In Designer with Workbench, fragments are stored in the Fragment Library and in the AEM forms repository.

Using fragments simplifies and speeds up the creation and maintenance of large numbers of forms. When creating a form, you insert a reference to the required fragment and the fragment appears in the form. The fragment reference contains a subform that points to the physical XDP file.

All fragments share common characteristics:

- You create all fragments the same way.
- You can create a fragment in the current file or in a separate file, and you can create multiple fragments in the same file.

  When you create a fragment in a separate file, a file is generated to store the fragment in the file system or in the AEM forms repository.
- Other form authors can use the fragments in their form designs.
- You edit the fragment source files in Designer.

Fragments and subforms

To create a fragment, you can either select an existing subform or select one or more objects. Subforms also include tables, table rows, header rows, and footer rows. If you select objects that are not in a subform, the objects are wrapped in a subform when the fragment is created.

You can also use a choice subform set that contains multiple fragment references. A *choice subform set* is a variation of the subform set object that allows you to customize the display of specific subforms from within the set by using conditional statements. You use conditional statements to determine which subform from within the set appears in the delivered form.

For example, each subform in a set can include information for a particular state, and the subform that is displayed can be determined based on the state where the form is filed.

Script fragments

A *script fragment* contains reusable JavaScript functions or values that are stored separately from a particular object, such as a date parser or a web service invocation. These fragments include a single script object that appears as a child of variables in the Hierarchy palette. Fragments cannot be created from scripts that are properties of other objects, such as event scripts like validate, calculate, or initialize.

For more information, see *UsingScriptFragments*.

RELATED LINKS:

Fragment Library palette menu

Binding fragments to a data source

Using choice subform sets

9.2. When to use fragments

Create fragments when you identify form content or a script that you intend to use in multiple forms. Fragments allow you to quickly create or change common elements that you want to share with multiple form authors. A company logo is a good example of content that you use in multiple forms. Changes to the logo are simplified because you make a single change in one place, which is the fragment file.

A script fragment is also useful where a function defines a standard way of formatting or doing some type of calculation.

Content reuse

You can use fragments to reuse content in multiple form designs. When you must use some of the same content in multiple forms, using a fragment is faster and simpler than copying or re-creating the content. Using fragments also ensures that the frequently used parts of a form design have consistent content and appearance in all the referencing forms.

Global updates

You can use fragments to make global changes to multiple forms only once, in one file. You can change the content, script objects, data bindings, layout, or styles in a fragment, and all XDP forms that reference the fragment reflect the changes. To update a fragment in a PDF form, resave the form in Designer.

For example, a common element across many forms can be an address block that includes a drop-down list object for the country. If you update the values for the drop-down list object, you must open many forms to make the changes. If you include the address block in a fragment, you open only one fragment file to make the changes.

Shared form creation

You can use fragments to share the creation of forms among several resources. Form developers with expertise in scripting or other advanced features of Designer can develop and share fragments that take advantage of scripting and dynamic properties. Form designers can use those fragments to lay out form designs and to ensure that all parts of a form have a consistent appearance and functionality across multiple forms designed by multiple people.

Security

If you have Designer and Workbench, you can use the AEM forms repository to limit access to a fragment and to store and share fragments.

RELATED LINKS:

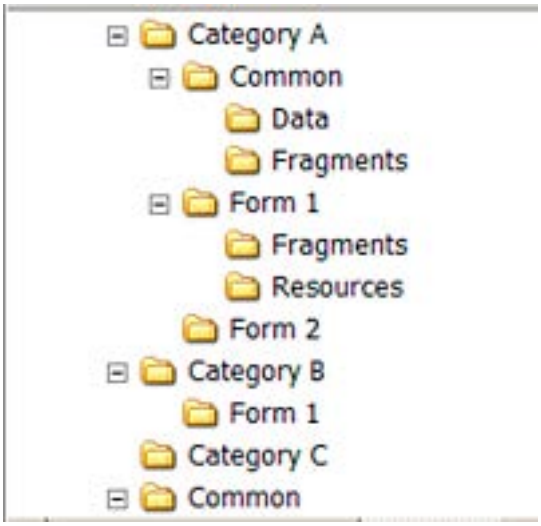
Inserting fragment references

Editing and embedding fragments

9.3. Naming and organizing and fragments

An easy way to organize fragments is to create folders that represent categories of forms and store the fragments in a common folder under each category. The common folders under each category of forms hold fragments common to those forms. It is also good practice to create a common folder outside the categories of forms to hold fragments that are common across all forms (for example, fragments such as a company logo or address in the common folder).

Example of folder structure that includes fragments



Make sure that you use a unique name for each fragment and that you add information about the purpose of the fragment in the Description box.

When stored in the AEM forms repository, the only way to view the fragment information is to open the fragment and view the information. Properly naming the fragment can make it easier for form authors to find the fragment they want to use.

You can maximize content reuse across teams by creating a fragments catalog and distributing it to form authors. For example, create a form design that displays all the fragments that are available in the fragment library in logical order. The fragments catalog provides a central location to view all fragments and obtain information about them.

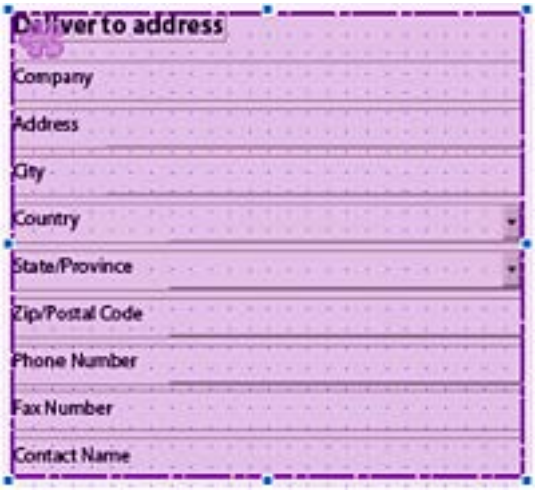
9.4. Fragment references

Designer provides visual cues that identify fragment references in the Layout Editor and the Hierarchy palette. These visual cues provide information about how fragments behave.

Fragment references in the Layout Editor

Object boundaries appear around fragments and fragment references. You can set the color of the object boundary borders in the Drawing Aids palette.

A fragment icon appears in the upper-left corner of a fragment reference and uses the color of the object boundary borders. The objects within a fragment reference have gray borders to indicate that you cannot edit them from the fragment reference.

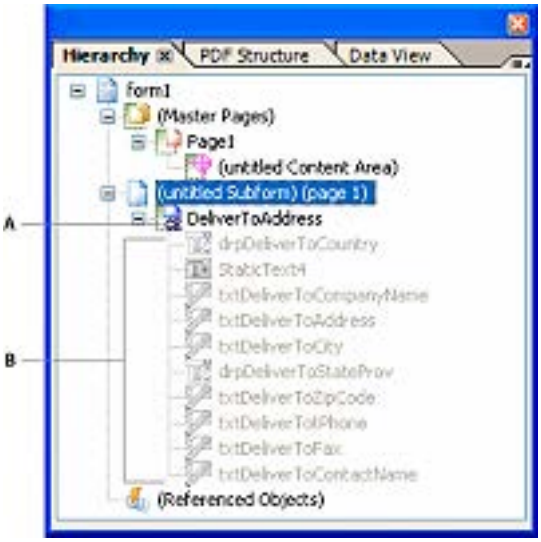


Clicking anywhere within a fragment reference selects it, and dragging anywhere on a fragment reference moves it.

You cannot select any of the objects in a fragment reference. For example, if the fragment reference contains a table, you cannot select a row or column. To select objects in a fragment reference, open the fragment source file for editing.

Fragment references in the Hierarchy palette

The objects in a fragment reference are dimmed in the Hierarchy palette because you cannot select or edit them from the fragment reference.



A.

Fragment reference icon on a subform

B.

The objects in a fragment are dimmed

You can move a fragment reference and its contents as a single entity within the Hierarchy palette in the same way that you move other form objects. However, you cannot drag objects into a fragment reference.

RELATED LINKS:

Drawing Aids palette

To use the drawing aids

Layout Editor

Hierarchy palette

Editing and embedding fragments

9.5. How fragment references are resolved

Fragment references are resolved when you open a form design in Designer, preview it, or save it as a PDF file. With Workbench, you can also save a form design as a PDF file in Forms or Output.

When you update a fragment, all form designs that reference the fragment are refreshed with the changes when you open the form design in Designer. All form designs that are open in Designer are also updated.

Fragment references in PDF forms are resolved when you open or save the form design in Designer. If you update a referenced fragment after you save the PDF form, the old fragment is still visible when you view the PDF form in Acrobat. To update the PDF form with the changed fragment, resave the PDF form in Designer.

RELATED LINKS:

Creating and inserting fragments

Inserting fragment references

9.6. Binding fragments to a data source

When you create a form design that includes fragment references, you cannot bind individual objects in the fragment. Instead, either bind the fragment reference to the data or create bindings in the fragment source file. When you create bindings in the fragment source file, set the binding of the fragment in the host form to No Data Binding so that it does not interfere in the data binding set in the fragment source file.

RELATED LINKS:

About fragments

Binding fields to a data source

9.7. Creating and inserting fragments

You can create a fragment in a separate XDP file or in the current form design. When you create the fragment in a separate file, you can replace the selected objects with a reference to the fragment file. If you do not replace the selection with a fragment reference, the fragment is created, but the selection remains unchanged and is not connected to the new fragment. Leaving the selection unchanged saves steps when you need to create multiple, similar fragments.

Creating the fragment in the current file is useful when you want to define multiple fragments in one file. Creating multiple fragments in one file can make it easier to update multiple fragments, particularly when the fragments are similar.

NOTE: You cannot create a fragment in a form created by importing a PDF file as artwork. For more information, see *Importing PDF documents as artwork*.

Fragments are identified by the fragment name. When you create the fragment in a separate file, you can also specify the location and the file name. The fragment name does not have to match the file name.

The fragment name appears in the Fragment Library when you click OK in the Create Fragment dialog box. When you create a fragment in the current form design and save it as an XDP file, the fragment name appears in the Fragment Library when you save the file.

When you create a fragment, you can either select an existing subform or select one or more objects. If you select objects that are not in a subform, the objects are wrapped in a subform when the fragment is created.

NOTE: After you create a fragment by selecting existing objects, update script references in the fragment to reflect that the objects are wrapped in a new subform.

You can use the menu to create a fragment, or you can drag the selected objects into the Fragment Library. When you create a fragment by dragging the selection into the Fragment Library, the option for creating the fragment in the current file is unavailable.

Save fragment source files as XDP files. If you save a fragment source file as a PDF file, the fragments cannot be referenced in other forms.

When you create a fragment in a AEM forms application and check it in, it is saved in the repository and appears in the Applications view in Workbench.

Create script fragments from script objects instead of subforms, and create them from the Hierarchy palette.

For more information about creating and inserting script fragments, see *UsingScriptFragments*.

Create a fragment

1) Select the objects to include in the fragment. You can select a fragment or multiple objects.
2) Select Edit > Fragments > Create Fragment.
3) (Optional) In the Description box, type a description of the fragment.
4) Select a method for creating the fragment:
   – To define the fragment in a separate XDP file that is stored in the Fragment Library, select Create Fragment In Fragment Library.
     • In the stand-alone version of Designer, select the Fragment Library where you want to save the fragment file.
     • In Designer with Workbench, from the Fragment Libary list, select a AEM forms application folder where you want to save the fragment file. You cannot select the root application folder.
     • (Optional) To use a different file name, in the File Name box, type the file name for the fragment.
     • To reference the new fragment in the current form design, select Replace Selection With Reference To New Form Fragment, and then click OK.

Inserting fragment references

You can use fragments to reuse content in multiple forms. When creating a form design, insert a reference to an existing fragment and the fragment appears in the form design.

After you insert a fragment reference, you can reposition it in the form design.

If you are using Designer with Workbench, you can insert a reference to a fragment that is outside the root application folder or is not in a Workbench application. However, you must copy the fragment and the files it references into the current application. The files are copied into the same folder as the current form, and the references to the files are updated.

NOTE: You cannot insert a fragment in a form that was created by importing a PDF file as artwork. (See *Importing PDF documents as artwork*.)

TIP: *To preview the fragments in the Fragment Library palette, select Show Preview Pane from the palette menu.*

To insert a fragment from the Fragment Library palette

1) Do one of the following actions:
   – Drag the fragment onto the form design.

     NOTE: *You cannot drag a fragment from the Fragment Library palette onto a subform that is a child of a SubformSet. The cursor changes and Designer displays a warning icon (a circle with a diagonal line) to indicate that this operation is illegal.*
   – Double-click the fragment.

To insert a fragment from the Insert menu

1) Select Insert > Fragment.
2) Navigate to the file that contains the fragment.
3) Select the file.
4) If the file contains more than one fragment, select the fragment.
5) Click OK. The fragment appears in the center of the visible page.

RELATED LINKS:

Fragment Library palette menu

To select objects

Create Fragment dialog box

Insert Fragment dialog box

9.8. Editing and embedding fragments

You can edit a fragment by selecting the fragment in the Fragment Library palette or by selecting the fragment reference in any form.

When you select the edit option, the fragment source file opens in Designer and the fragment is selected.

NOTE: You can also open and edit a fragment source file the same way you edit any other XDP file.

You can rename the fragment or modify the form objects in the fragment. When you save the fragment source file, your changes are immediately reflected in all open forms that reference the fragment. You do not need to reload the forms to see the changes.

NOTE: Renaming the fragment subform in the Hierarchy palette in the referenced XDP file breaks all references to that fragment. You can change the fragment metadata name in the Fragment Information dialog box or in the Object palette.

When you edit a fragment, all forms that reference the fragment reflect the changes when you save the form in Designer. If you update a referenced fragment after you save a PDF form, the old fragment is still visible when you view the PDF form in Acrobat. To update the fragment that is visible in the PDF form, resave the PDF form in Designer

To edit a fragment from the Fragment Library palette

1) Open the fragment library and select the fragment.
2) In the fragment library panel menu, select Edit Fragment. The fragment source file opens.
3) Edit the file as required.
4) Save the fragment source file.

To edit a fragment from a fragment reference

1) Do one of the following actions:
   – Double-click the fragment reference.
   – Select Edit > Fragments > Edit Fragment.

   The fragment source file opens.
2) Edit the file as required.
3) Save the fragment source file.

To view fragment information

You can view the fragment information that was added when the fragment was created. The fragment information includes the fragment name, a description, and the location of the fragment source file.

The same information is displayed for fragments and fragment references. When you select a fragment, you can view or edit the information, but when you select a fragment reference, you can only view the information.

1) View the fragment information:
   – To view the fragment information for a fragment reference, select the fragment reference.
   – To view the fragment information for a fragment, open the fragment for editing.
2) Select Edit > Fragments > Fragment Info.

To convert a fragment reference to an embedded object

When you convert a fragment reference, it becomes an embedded copy of the fragment, and the fragment metadata is removed. The embedded object does not retain any relationship to the fragment source file. The fragment source file is unchanged and you can still reference it in your forms.

Converting a fragment reference is useful when you no longer need to reference the fragment but do not want to remove the objects from a form design. For example, you can convert a fragment reference if you want to create a new fragment based on an existing fragment.

If the referenced fragment contains one or more fragment references, you can either embed only the selected fragment reference or embed the selected fragment and the nested fragment references.

1) Select the fragment reference to embed.
2) Select Edit > Fragments > Convert To Embedded Object.

RELATED LINKS:

About fragments

Fragment references

9.9. Viewing and embedding fragments

To view fragment information

You can view the fragment information that was added when the fragment was created. The fragment information includes the fragment name, a description, and the location of the fragment source file.

1) View the fragment information:
   - To view the fragment information for a fragment reference, select the fragment reference.
   - To view the fragment information for a fragment, open the fragment.
2) Select Edit > Fragments > Fragment Info.

To convert a fragment reference to an embedded object

When you convert a fragment reference, it becomes an embedded copy of the fragment, and the fragment metadata is removed. The embedded object does not retain any relationship to the fragment source file. The fragment source file is unchanged and you can still reference it in your forms.

Converting a fragment reference is useful when you no longer need to reference the fragment but do not want to remove the objects from a form design. For example, you can convert a fragment reference if you want to create a new fragment based on an existing fragment.

If the referenced fragment contains one or more fragment references, you can either embed only the selected fragment reference or embed the selected fragment and the nested fragment references.

1) Select the fragment reference to embed.
2) Select Edit > Fragments > Convert To Embedded Object.

RELATED LINKS:

About fragments

Fragment references

9.10. Fixing overrides and broken fragment references

You create overrides on a fragment reference when you edit its properties in the referencing form design. These changes affect only the fragment reference in the current form and do not affect the fragment source file or any other references.

You can create fragment reference overrides on properties in the Object, Layout, Border, and Accessibility palettes of the fragment subform. In the Script Editor, you can create overrides for scripts that are defined in the fragment reference subform. The properties you can edit include the tab order, binding, scripts, X and Y positions, and subform properties such as presence and locale.

You cannot create overrides on the properties of any objects in the fragment reference, including nested objects such as subform sets or tables. For example, if a fragment reference includes a table, you cannot edit the row shading because row shading is a property of the cells in a table.

To remove fragment reference overrides

You can remove the overrides from fragment reference properties by using the Clear Fragment Property Overrides option in the Object, Border, Layout, and Accessibility palette menus. This option removes all overrides, on all the palettes, from the selected fragment reference, except overrides on the X and Y properties when the fragment is in a subform that positions content.

When overrides exist, an information marker appears on the fragment reference and an entry appears in the Warnings tab in the Report palette.

1) Select the fragment reference that has overrides.
2) In the palette menu of the Object, Border, Layout, or Accessibility palette, select Clear Fragment Property Overrides.

To resolve broken fragment references

A fragment reference can be broken when Designer cannot find either the fragment or the fragment source file because it was renamed, deleted, or moved.

When you open a form that has a broken fragment reference, the fragment reference is replaced with this broken fragment icon.

The broken fragment icon also appears on broken fragment references in the Hierarchy palette.

1) In the Object palette, click Open Fragment Source File.
2) Select the file that contains the fragment and click Open.
3) In the Fragment Name box, select the name of the fragment.

RELATED LINKS:

About fragments

Fragment references

About palettes

Properties in the Object palette

9.11. Fragments tips

When working with fragments, keep in mind the following points:

- Because fragments are used for content reuse, keep them generic enough so that they do not quickly become unusable in some forms when changes are made to it.
- When changing a fragment, verify whether you must also change the following items:
  - Digital signatures that sign a collection that includes the fragment file.
  - Update the schema for the fragment or host form to accommodate the changes.
  - The form design layout of the host form to ensure that the changes did not cause errors. More work may be required to complete the change to the host forms. If you are not using the AEM forms server to generate the PDF files on demand, manually open each form design and resave it as a PDF file in order for the fragment changes to appear. If the PDF files are generated by using the AEM forms server, the fragment references in the host form design are resolved before the form is rendered so that no additional effort is required.
- When placing fragments on a master page, leave consistent space between the page border and the fragments to maintain consistent margins.
- When creating the fragments, such as the body of a letter, consider the spacing you want between each paragraph. Then make the spacing part of the static text object that contains the paragraph or part of the fragment subform itself (where the fragment subform has a greater height than the static object it contains). When the paragraph fragments are flowed into the body pages, they are consistently spaced. The easiest way to add spacing is to use the Paragraph palette and define the spacing on the static text object.
- When creating paragraph fragments that contain floating fields, always ensure that they are wide enough to allow for arbitrary data width. For example, make the salutation of a letter as wide as the page so that it can accommodate long names.

10. Working with Objects

10.1. About Objects

Objects are the building blocks of every form. Each object provides some piece of functionality to your form, such as a place to enter text or a button to use to email the form. As you create your form, you select objects and add them to the body or master page of the form design. You will find all the available objects in the Object Library palette.

The objects in the Object Library palette are grouped into category. The Standard and Barcode categories contain the core objects. In addition, a number of predefined custom objects are available in the Custom category.

Categories of objects

The majority of the objects fall into two categories:

Boilerplate or static objects

Static objects do not capture or display data. Instead, they present fixed text or graphic information. These objects include circles, lines, rectangles, images, and text. Static objects can be used in all types of forms.

Field objects

Field objects both capture and display data. These objects include barcodes, buttons, check boxes, date/time fields, drop-down lists, image fields, list boxes, password and signature fields, radio buttons, and text fields. Some field objects, such as text fields, can be used in all types of forms, whether the form is interactive, to be filled by a user, or to be filled by merging with a data source. Others, such as drop-down lists, are intended specifically for interactive forms.

The following objects are available in the Standard and Barcodes categories of the Object Library palette.

Object	Description	Category	Use	For more information
Barcode	A barcode identification symbol.	Static	Designed for any type of form.	*Using barcodes*
Button	A button for issuing commands or client requests.	Field	Designed for interactive forms.	*Using buttons*

Object	Description	Category	Use	For more information
Check Box	An object that has an enabled (on) or disabled (off) state.	Field	Designed for any type of form.	*Using checkboxes*
Circle	A circle, ellipse, or arc.	Static	Designed for any type of form.	*Using circles, lines, and rectangles*
Content Area	A container that serves as the top-level object in an object hierarchy and defines the area in a form in which objects may be placed.	Static	Every master page defines at least one content area.	*Using content areas*
Date/Time Field	A field that accepts and displays date/time data and supports pattern recognition.	Field	Designed for any type of form.	*Using date/time fields*
Decimal Field	A field that accepts and displays decimal data and supports pattern recognition.	Field	Designed for any type of form.	*Using decimal and numeric fields*
Extended Decimal Field	A field that accepts and displays decimal data up to 31 digits and supports pattern recognition.	Field	Designed for any type of form.	*Using extended decimal fields*
Signature Field	A control that users can use to attach an electronic signature to the form.	Field	Designed for any type of form.	*Using signature fields*
Drop-down List	A list of multiple options from which one option can be selected. Only one option appears at a time.	Field	Designed for interactive forms.	*Using drop-down lists and list boxes*
Email Submit Button	A button that form users can use to return form data to a specified email address.	Field	Designed for interactive forms.	*Email submit buttons*
Flash Field	An object that displays flash content such as an instructional video..	Field	Designed for interactive forms.	*Using flash fields*

Object	Description	Category	Use	For more information
HTTP Submit Button	A button that form users can click to return form data by HTTP post to a specified URL.	Field	Designed for interactive forms.	*HTTP submit buttons*
Image	An object that displays an image such as a logo or icon.	Static	Designed for any type of form.	*Using image fields*
Image Field	A placeholder for loading an image dynamically when the form is rendered.	Field	Designed for any type of form.	*Using image fields*
Lines	A solid, dashed, or dotted line.	Static	Designed for any type of form.	*Using circles,lines, and rectangles*
List Box	A list of multiple options from which one option can be selected. More than one option appears at a time.	Field	Designed for interactive forms.	*Using drop-down lists and list boxes*
Numeric Field	A field that accepts and displays numeric data and supports pattern recognition.	Field	Designed for any type of form.	*About numeric fields*
Paper Forms Barcode	A 2D barcode that encodes user-entered data in an interactive form. To use the paper forms barcode, your organization must have the Acrobat Reader DC extensions implemented.	Field	Designed for interactive forms.	*Using drop-down lists and list boxes*
Password Field	A field that accepts and masks the display of alphanumeric passwords.	Field	Designed for any type of form.	*Using password fields*
Print Button	A button that is configured for form users to print the form.	Field	Designed for interactive forms.	*Print buttons*

Object	Description	Category	Use	For more information
Radio Button	An object that represents a single choice in a group of mutually exclusive choices.	Field	Designed for any type of form.	*Using radio buttons*
Rectangle	A rectangle with regular, notched, or rounded corners.	Static	Designed for any type of form.	*Using circles, lines, and rectangles*
Reset Button	A button that form users can click to reset the contents of the form's fields to their default values.	Field	Designed for interactive forms.	*Reset buttons*
Subform	A container that controls the positioning of objects. Each page is a subform and may contain additional subforms.	Static	Designed for any type of form.	*About subforms*
Table	A container made up of rows and columns of cells that you can fill with form fields or merge with data.	Can be a static object or field object	Designed for any type of form.	*Using tables*
Text	Read-only text.	Static	Designed for any type of form.	*Using text*
Text Field	A field that accepts and displays textual data and supports pattern recognition.	Field	Designed for any type of form.	*Using text fields*

RELATED LINKS:

Using custom objects

To create a custom object

To change an object type

There are several ways you can change the object type:

- Delete the object on your form design and then drag the required object from the Object Library palette

- Use the Type list in the Field tab of the Object palette (for field objects such as text fields, numeric fields, and buttons)
- Use the list in the Object Editor (for field objects such as text fields, numeric fields, and buttons)
- Use the Type list in the Draw tab of the Object palette (for static objects such as circles, lines, and text)

To change an object type using the Type list in the Object palette tabs

1) Select the object you want to change.
2) In the Object palette, click either the Field tab or the Draw tab and then select another object type from the Type list.

To change an object type using the Type list in the Object Editor

1) Select the object you want to change. If the Object Editor does not appear around the object, select View > Object Editor.
2) Select another object type from the Type list.

To name and rename objects

When you drag an object from the Object Library palette onto your form design, its default name is the name of the object followed by a number that represents the number of copies of the same object in the form design. For example, if you have two list box objects on your form design, the default names are ListBox1 and ListBox2. This naming is called the *occurrence number*.

While creating the form design, you can see the name of the object in the following places:

- Hierarchy palette
- Binding tab of the Object palette
- Object Editor
- A pop-up tool tip on the form design when you position the pointer over the object

You can name or rename an object by using the Hierarchy palette, the Binding tab of the Object palette, or the Object Editor.

NOTE: If you plan to create calculations or scripts to enhance your form, avoid using the names of scripting properties, methods, and objects when naming your form objects. Using similar names can result in calculations and scripts not executing properly.

To name or rename an object by using the Hierarchy palette

1) In the Hierarchy palette, right-click the object and select Rename Object.
2) Type the new name and press Enter.

To name or rename an object by using the Binding tab of the Object palette

1) Select the object.
2) In the Object palette, click the Binding tab.

3) In the Name box, type a new name for the object and press Enter.

To name or rename an object by using the Object Editor

1) Select the object. If the Object Editor does not appear, select View > Object Editor.
2) In the Name box, type a new name for the object and press Enter.

To set the border style

- To set the border style for an object, in the Object palette, click the Field tab and select a border style for the box from the Appearance list.
- To define a custom look for an object, select Custom from the Appearance list.
- To define a border for the whole object, use the Border palette.

RELATED LINKS:

Border properties in the Border palette

To specify a locale (language and country or region) for an object

1) In the Object palette, click the Field tab.
2) In the Locale list, select one of these options or one of the provided alternatives for the object's localization setting:
   – To use the default locale specified in the Defaults tab of the Form Properties dialog box, select Default Locale.
   – To use the system locale of the user's computer, select Viewer's System Locale.

   NOTE: At design time and run time, formatted values in the field are displayed in the locale-sensitive format.

RELATED LINKS:

Locales

Objects that support scripting and calculations

Objects that accept data can be associated with FormCalc or JavaScript expressions. For example, you can use scripts to calculate a value or validate user-entered data. In most cases, the validity of data is controlled through the attributes of the object. In the case of objects that require complex input (edit) patterns, the validity is controlled through pattern settings in the Value tab of the Object palette.

Some objects can be used to initiate client requests. For example, you can set up a button to query a data source at run time. In this case, you must also write a script to return the requested data value to a specific field in the form.

You can write scripts to execute commands, functions, or calculations when any of an object's events occur at run time. An event will initiate at run time when the user performs the action that the event specifies. You can call any of an object's supported methods and examine or set properties by defining a script.

Some objects, such as circles and content areas, do not support events. The objects that support scripting and calculations or that initiate events include barcodes, buttons, check boxes, date/time fields, drop-down lists, flash fields, image fields, list boxes, numeric fields, password fields, radio buttons, signature fields, subforms, and text fields. The supported events are object-specific and vary from one object to the next.

RELATED LINKS:

Using objects in interactive forms

Using objects in interactive forms

Designer includes several standard and custom objects that provide data entry and access capabilities, and support calculations and scripts to initiate actions and handle data capture and manipulation. You can use any of these objects on a form, as well as objects such as text fields and drawing objects.

In addition, you can create custom objects whose properties and characteristics you plan to use often in your forms. You create custom objects by defining properties and adding required functionality such as scripts or calculations to standard Designer objects. You save custom objects in the Object Library palette, either in the Custom category or in a category you create yourself.

When choosing objects, you should understand what functions and capabilities the objects provide. There are several things to consider when selecting objects:

- Determine the type of information you need to include in your forms and familiarize yourself with the characteristics of the objects.
- Be aware that fields have varying length and appearance requirements and options, as well as data entry restrictions and requirements.
- Analyze the types of data that you want to display and capture, and determine which objects support those types of data.
- Certain objects allow you to add calculations and write scripts to initiate actions on the field. Determine what types of scripts you can use with particular objects.

RELATED LINKS:

About Objects

To create a custom object

10.2. Using objects

Using barcodes

Businesses use barcodes extensively, particularly for inventory control. Barcodes can be used to identify forms, but they are often printed on adhesive paper to create labels for inventory purposes.

Designer supports two types of barcodes:

Hardware barcodes

Only use when the form is being printed directly to the printer from the server. Because a printer is required to print them, Designer uses a placeholder to represent hardware barcodes in the form.

Software barcodes

Can be drawn by Designer and are visible in Acrobat and Adobe Reader. They can be printed on any general-purpose printer.

Some types of barcodes can hold arbitrary binary data. Others are limited to a particular set of characters or codes. It is the responsibility of the form author to ensure that the data is appropriate for the barcode, for example, by imposing a validation on the field.

RELATED LINKS:

Supported barcode formats

Valid barcode text characters

To dynamically populate a validation pattern message

To dynamically populate a validation script message

Barcode properties in the Field tab

Barcode properties in the Value tab

Barcode properties in the Binding tab

Barcode formats

Designer provides a variety of barcode formats that you can work with. The supported barcode formats are listed in the Barcodes category of the Object Library palette. If you use a barcode that requires a specific type of printer, Designer represents the barcode as a shaded rectangle in the form.

NOTE: Interactive barcodes that can accept user input are only supported for PDF forms that are filled in Acrobat 7.0.5 or Adobe Reader 7.0.5 or later.

Designer also supports the two-dimensional paper forms barcode. For more information, see *Using paper forms barcodes*.

Barcode position and appearance

The function of a barcode is to be read by a specialized piece of hardware called a barcode reader. Since the barcode is intended to be read by a machine, its appearance is usually strictly constrained. For example, for a particular type of barcode, the bars may have to be a particular height and distance apart. In addition it is common for a barcode to require a minimum amount of white space around it (the quiet zone) and a particular range of distances from a designated edge of the page.

Designer does nothing to express or enforce positioning or quiet zone requirements. It is up to the form creator to ensure that these requirements are met.

Barcode properties

After you add a barcode to the form design, you can manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Text position
- Length of the data
- Any additional properties supported by the barcode (for example, optional checksum capabilities, and text positioning and embedding)
- Presence of the barcode as visible, invisible, or hidden
- Binding method for storing and retrieving bound data

To define barcode field properties

To define the properties of a barcode, you must first select the barcode and set basic characteristics in the Field tab of the Object palette. The number and type of barcode properties vary from one barcode to another.

1) Select the barcode.
2) In the Object palette, click the Value tab and then type the barcode text in the Default box.
3) Click the Field tab and, in the Location box, select the placement of the text.

   The PDF 417 format does not support text positioning. It supports binary data of size upto 1 Mb. The EAN8, EAN13, and UPC-A formats support the Below Embedded option only.
4) In the Value column, do one or more of the following actions:
   - In the Data Length box, type the length of the data. If you are defining an MSI barcode, the data length must be a value between 1 and 14. The MSI, UPC-E, UPC EAN2, UPC EAN5, US Postal Zip-5, US Postal DPBC, and US Postal Standard barcode formats have fixed data lengths that cannot be changed.
   - If a Checksum box is displayed, enable or disable the checksum. To enable the default checksum, select Auto. To disable the checksum, select None. If you are defining an MSI barcode, select one of the available checksum methods.
   - If an End Char or Start Char box is displayed, type the end character, start character, or both. If you are defining a Codabar barcode, the valid end and start characters are A, B, C, D, a, b, c, d, *, N, T, E, n, t, and e. If you are defining a Code 49 barcode, the valid start characters are A, 1, 2, 3, 4, and 5.
   - If a Wide/Narrow Ratio box is displayed, type a wide-narrow-ratio value. For Codabar, Code 2 of 5 (Industrial, Interleaved, and Matrix), and Code 3 of 9, the wide-narrow-ratio value must be a value from 2.2 to 3.0. For Code 11, Logmars, MSI, and Plessey barcodes, the value must be a value from 2.0 to 3.0.

To set the default value for the barcode

1) In the Object palette, click the Value tab and then type a default value in the Default box.

To control how a barcode obtains data

To define the properties of a barcode, you must first select the barcode. Set properties that control how the barcode obtains data in the Value tab of the Object palette.

TIP: You can dynamically populate a validation pattern or script message with a value from a data source. This allows you to ensure users enter the correct value in the field.

1) In the Object palette, click the Value tab and then select one of these options from the Type list:
   - To allow users to choose to enter data or not, select User Entered - Optional.
   - To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
   - To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
   - To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users cannot edit the calculated value.
   - To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.
   - To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users cannot edit the value.
2) If the value is recommended or required, type a prompt in the Empty Message box.
3) If the value will be calculated, attach the calculation script to the object by using the Script Editor.
4) (Optional) If a calculated value can be overridden, type a message into the Override Message box.

To define custom data-binding properties for a barcode

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the barcode.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the barcode to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

Encoding non-printing characters in barcode data

You can encode hidden, non-printing characters between the data in Code 128 barcodes. For example, you can encode the characters that represent the prefix and the delimiter as part of the Code 128 barcode standard. The application that reads the barcode then decodes the meaning of these characters.

Some characters are reserved for use as non-printing characters in Code 128 barcodes. To encode these characters in a barcode, use the following mechanism in Designer.

Code 128 non-printing characters	Use these values in Designer
FNC1	[F1]
FNC2	[F2]

Code 128 non-printing characters	Use these values in Designer
FNC3	[F3]
SHIFT	[SH]
Change to Subset A	[CA]
Change to Subset B	[CB]
Change to Subset C	[CC]
Start in Subset A	[SA]
Start in Subset B	[SB]
Start in Subset C	[SC]

For example, to encode FNC1 in a Code 128 barcode, insert [F1] as follows: 00[F1]12345[F1]67890.

Using buttons

If you want users to initiate actions such as submitting data, executing a web service operation, or executing a database query, you can add a button to the form. With the help of Designer's built-in support for client-server communications and scripting, forms can support these kinds of actions through buttons:

- Execute a calculation
- Process and manipulate data through a script
- Submit data (including optional attachments) to a server
- Sign submitted form content
- Encrypt submitted form content
- Open a connection to a host
- Submit client requests to a server
- Invoke a web service operation
- Query a data source

The actions associated with the button are initiated when the user clicks the button.

After you add a button object to the form design, you can edit the caption text and manipulate the object's properties in the Field, Submit, and Execute tabs of the Object palette. You can define these properties:

- Change the caption for the button
- Set a border style for the button
- Specify the highlight style for the button
- Define the button as visible, invisible, or hidden
- Specify a locale for the button

- Specify run-time behavior

Designer also includes four additional buttons that are preconfigured to provide specific functions. They include an email submit button, an HTTP submit button, a print button, and a reset button.

RELATED LINKS:

Using digital signatures

Using XML encryption

Button properties in the Field tab

HTTP submit button properties in the Field tab

Email submit button properties in the Field tab

About processing options for a button

Working around web browserlimitations

Email submit buttons

The email submit button is a standard button that has certain properties already set and a specialized Object palette. This object makes it easier for you to create buttons that users can click to return their form data by email. Because Adobe Reader does not save changes to PDF files, including form data, it is a good idea to include an email submit button on forms that users may fill in Adobe Reader.

The email submit button operates like a standard button object with these settings:

- Control Type set to Submit
- Submit Format set to XML Data
- Submit To URL set to use the mailto: protocol, such as mailto:name@adobe.com
- Encoding set to UTF-8

If you use the New Form Assistant to create a form and select one of the Submit return methods, an email submit button will be included on the form automatically.

After you add a button to the form design, you can edit the caption text, and you can manipulate the object's properties in the Field tab of the Object palette. You can define these properties:

- Change the caption for the button
- Set a border style for the button
- Specify the highlight style for the button
- Set the To address and the Subject of the email message
- Sign submitted form content
- Encrypt submitted form content
- Define the button as visible, invisible, or hidden
- Specify a locale for the button

You can also change the button name in the Hierarchy palette and set a style for the caption in the Font and Paragraph palettes.

To configure the email submit button

You can specify the e-mail address that the form data will be sent to and the subject line of the email that will be sent.

1) Add an email submit button to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Email Address box, type the e-mail address you want to send the submission to.
4) (Optional) In the Email Subject box, type a subject line for the e-mail.
5) (Optional) To apply a data signature to the submitted data, select Sign Submission and then click Settings to configure optional signature settings.
6) (Optional) To apply XML encryption to the form content, select Encrypt Submission and then click Settings to configure optional encryption settings.

HTTP submit buttons

The HTTP submit button is a standard button that has certain properties already set and a specialized Object palette. This object makes it easier for you to create buttons that users can click to return their form data by HTTP post.

The HTTP submit button operates like a standard button object with these settings:

- Control Type set to Submit
- Submit Format set to URL-Encoded Data
- Submit To URL set to use the http: protocol

If your data needs to be returned through the secure https: protocol, you can also use the standard button object.

After you add a button to the form design, you can edit the caption text, and you can manipulate the object's properties in the Field tab of the Object palette. You can define these properties:

- Change the caption for the button
- Set a border style for the button
- Select the highlight style for the button
- Specify the URL that the data will be sent to
- Sign submitted form content
- Encrypt submitted form content
- Define the button as visible, invisible, or hidden
- Specify a locale for the button

You can also change the button name in the Hierarchy palette and set a style for the caption in the Font and Paragraph palettes.

To configure the HTTP submit button

1) Add an HTTP submit button to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab and type the URL protocol in the URL box.

To	Use this URL protocol	Example
Submit the package to an ftp site	ftp	ftp://ftp.gnu.org/gnu/GPL
Submit the package to a web server	http	http://myserver/cgi-bin/
Submit the package to a secure web server	https	https://myserver/cgi-bin/
Submit the package to an email address	mailto	mailto:username@domain.com

NOTE: When submitting data to a URL, it is recommended that you specify an absolute target. Relative targets are interpreted relative to the user environment, which can vary from one user to the next at run time.

3) (Optional) To apply a data signature to the submitted data, select Sign Submission and then click Settings to configure optional signature settings.
4) (Optional) To apply XML encryption to the form content, select Encrypt Submission and then click Settings to configure optional encryption settings.

Print buttons

The print button opens a Print dialog box so that the user can print the form. A Print button is a standard button object that has the Control Type set to Regular and a script included in the button's `click` event that prints the form when the button is clicked.

In the stand-alone version of Designer, if you use the New Form Assistant to create a form and select one of the Print return methods, a print button will be included on the form automatically.

Reset buttons

A reset button resets all fields on the form to their default values. The Reset button object is a standard button object that has the Control Type set to Regular and a script included in the button's `click` event that resets the field values when the button is clicked.

To insert a standard button that runs a calculation or script

1) Add the button to a form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) Select an event from the Show box. For example, if you want the button to perform a calculation, select Click.
4) In the Script Editor, select a language from the Language list.
5) Type the calculation or script in the Script Editor.

Processing options for a button

In the Execute tab in the Object palette, you can select a data connection for the button and specify whether the associated processing will be carried out on the client computer, the server, or both. The data connection must provide access to a web-service interface or OLEDB database server that will either return data to the form at run time or perform an action elsewhere.

After you specify the data source (see *Working with Data Sources*), you can specify the operation or query to run. When the button is clicked at run time, processing is completed as defined through the operation or query.

Returned data can be merged with the form so that only data associated with existing objects is overwritten without refreshing the structure of the form. Alternatively, you can choose to update the structure of the form based on the returned XML data. In this case, the structure of the form is updated, and the returned data is merged after processing is completed.

To define processing options for a button, you must first select the button. The processing options are set in the Execute tab of the Object palette. To view the Execute tab, the Execute option must be selected in the Control Type area in the Field tab.

RELATED LINKS:

Button properties in the Execute tab

Previewing and testing forms

Working with Data Sources

To create a bound field or subform

To execute a web service operation or database query using a button

Using an Execute button type, you can execute a process request to a web service. A web service processes operation that you send to it to define input and output information in your form.

You add a connection to the web service and then create or bind fields to the request and response messages. The bound fields can generate input data to send to the service or receive data from the service. Multiple fields can be used to create the request message, and the response message can populate multiple fields. To process the operation, you can use an Execute button or write a script. For example, you can use a web service to look up an interest rate, a currency exchange rate, or a stock quote.

1) Add a button to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Control Type area, select Execute. A web-service operation or database query will be executed according to the settings in the Execute tab.
4) Click the Execute tab and select New Data Connection from the Connection list.
5) Select WSDL File and click Next.
6) In the WSDL File box, type the URL to the WSDL document, or click Browse to locate and select the document on your computer's hard disk. If the WSDL document is stored on a secure server, either the Authentication Required dialog box or the Select a Digital ID dialog box is displayed.

7) Perform one of the following actions:
   – In the Authentication dialog box, type the correct user name and password, and click OK.
   – In the Select A Digital ID dialog box, select the correct digital ID to use as a client certificate, and click OK.
8) Select the operation to call, and click Next.

   The window on the right displays details for the selected operation. If the selected operation is not SOAP-encoded or is not doc-literal, a warning appears in this window and you cannot continue.
9) (Optional) To set up client authentication, perform one of the following actions:
   – To define HTTP or HTTPS authentication, select Requires HTTP/HTTPS Authentication, and then optionally select the credentials to accept for verification.
   – To define SOAP message authentication, select Requires Message Level Authentication, and then optionally select Accepts User Name and Password.
10) Click Finish.
11) In the Run At list, specify where the processing will take place:
   – To process the request on the client computer, select Client.
   – To process the request on the server, select Server.
   – To process the request on the client computer and server, select Client And Server.
12) To update the structure of the form and merge returned data after processing is completed, select Re-merge Form Data. You must use the Re-merge Form Data option if the response data is going to populate dynamic data (such as a repeating subform) that is bound to WSDL data, which may occur a variable number of times in the response. Dynamic data requires a full remerge to generate the correct number of repeating subforms. If the WDSL response data is bound to a fixed number of fields, the response data will populate the existing fields, such that you do not need to re-merge the form data.

To attach a database script to a button

You can use scripting to perform more complex data manipulation on an enterprise system. In this example, you can see how to add some simple script to button fields to do basic database operations, including moving to the first, previous, next, and last records, adding new records, updating existing records, and deleting records.

NOTE: This example is based on a particular database and configuration that is not included in the Designer samples.The example is intended as a reference for creating your own database lookup scripts.

1) In the Object Library palette, click the Standard category and drag eight button objects onto the form design. When you add a button object, the default control type is Regular. Accept the default setting.
2) Name the buttons as follows: First, Previous, Next, Last, Add New, Update, Delete, and Cancel.
3) Select the first button, in this case the one named First.
4) In the Script Editor, select `click` from the Show list, select JavaScript from the Language list, and select Client from the Run At list.
5) In the Script Editor, add the following JavaScript script:

```
xfa.sourceSet.DataConnection.first();
```

6) Repeat step 5 for each of the remaining buttons, using the following scripts:
   - Previous:`xfa.sourceSet.DataConnection.previous();`
   - Next:`xfa.sourceSet.DataConnection.next();`
   - Last:`xfa.sourceSet.DataConnection.last();`
   - AddNew:`xfa.sourceSet.DataConnection.addNew();`
   - Update:`xfa.sourceSet.DataConnection.update();`
   - Delete:`xfa.sourceSet.DataConnection.delete();`
   - Cancel:`xfa.sourceSet.DataConnection.cancel();`
7) Save the form design.
8) Test the form using the Preview PDF tab. If such a form were being filled in Adobe Reader, you would have to set usage rights for the form by using Acrobat Reader DC extensions in order for Adobe Reader to access the database.
9) Click the different buttons to display the data associated with each record. The example shows that the Last button causes the ID, part number, description, and unit price for the last record to appear in the applicable fields.

Using a button to populate fields from a database

You can use a database to provide the data for fields in your form, including text and numeric fields.

In this example, the Retrieve Record button and the list box are bound to the database (named DataConnection) shown in the Data View palette at the left. The button also has a FormCalc script attached to it to retrieve information based on the part number selected in the List Box field. The retrieved records appear in the ID, PART_NO, UNITPRICE, and DESCRIPTION fields.

1) Connect to the database.
2) In the Data View palette, drag the UNIT_PRICE, DESCRIPTION, PART_NO, and ID elements under DataConnection onto the form design. Each node dragged onto the form design creates a bound field.
3) In the Object palette, click the Standard tab and drag a button object onto the form design.
4) Enter a new name for the button. The example uses RETRIEVE RECORD.
5) Select the button and, in the Script Editor, select click from the Show list, select FormCalc from the Language list, and select Client from the Run At list.
6) Enter a FormCalc script similar to the following example. The script retrieves the data in the specified records and prints it in the corresponding fields when the user clicks the button.

```
//Change the commandType from TABLE to TEXT. TEXT
//is the equivalent of SQL Property
$sourceSet.DataConnection.#command.query.commandType = "text"

//Set the Select Node. Select in this case will be
//whatever SQL Property you want
$sourceSet.DataConnection.#command.query.select.nodes.item(0).value =
Concat("Select*from OfficeSupplies Where ID = ",
DataListBox1.rawValue,"")

//Reopen the Data connection
$sourceSet.DataConnection.open()
```

7) Save the form design as a PDF file.
8) Test the form using the Preview PDF tab to verify that the script works as expected.

Submitting data using a button

You can add a button to a form so form fillers can submit the information or data they enter into the various fields in the form. The Object Library provides three button objects you use to submit form data, the Button, Email Submit Button, and HTTP Submit Button objects.

To define the properties of a button for submitting data, you select the button object in the Layout Editor and select options on either the Submit or Field tab in the Object palette. With the Button object, you first select the Submit option in the Control Type area on the Field tab, to display the options for submitting data on the Submit tab. With the Email Submit Button and the HTTP Submit Button objects, the options for submitting data are located on the Field tab.

Each submit button provides options for different purposes. For example, if you want to submit data to a URL, use the Button or the HTTP Submit Button objects. However, if you want to submit the data as an XML Data Package (XDP) including attachments like annotations, PDF documents, or signatures, use the Button object.

Use the Button object to submit form data to a URL in XML Data Package (XDP), PDF, XML Data (XML), or URL-encoded data format. The Button object is the only button that provides the options to include attachments like annotations, templates, and PDFs. You can also use the Button object to submit form data by email using the mailto protocol.

Use the Email Submit Button object to submit form data to an e-mail address in XDP or PDF format.

Use the HTTP Button object to submit form data to a URL in URL-encoded data format.

All three buttons provide options for signing and encrypting submitted data.

RELATED LINKS:

Using digital signatures

Using XML encryption

Button properties in the Submit tab

Email submit button properties in the Field tab

HTTP submit button properties in the Field tab

To add a button that submits an XML Data Package (XDP) to a URL

1) Add a Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Control Type area, select Submit. Data will be submitted according to the settings in the Submit tab.
4) Click the Submit tab, and in the Submit to URL box, type the URL protocol.

To	Use this URL protocol	Example
Submit the package to an ftp site	ftp	ftp://ftp.gnu.org/gnu/GPL
Submit the package to a web server	http	http://myserver/cgi-bin/
Submit the package to a secure web server	https	https://myserver/cgi-bin/

NOTE: When submitting data to a URL, it is recommended that you specify an absolute target. Relative targets are interpreted relative to the user environment, which can vary from one user to the next at run time.

5) In the Submit As list, select XML Data Package (XDP).

These formats initiate server-side processing. The XML Data Package (XDP) option submits the form data and optionally includes other information, such as the form design, annotations, and signa-

tures, that is needed for Forms to subsequently render the form at run time. Attachments may be included with XDP files only.

6) (Optional) To apply a signature to the submitted data, select Sign Submission and then click Settings to select optional signature settings.

7) (Optional) To apply XML encryption to the form content, select Encrypt Submission and then click Settings to select optional encryption settings.

8) In the Include area, select the types of attachments to include:
   - To include review comments, tool tips, and any other special tags needed to capture screen reader text, select Annotations.
   - To include a PDF version of the form (including signatures) when it is submitted as an attachment, select PDF (Includes Signatures). Otherwise, a reference to an embedded PDF file is included.
   - To include a copy of the form design, select Template.
   - To include one or more <xdp> elements in the XDP source file, select Other. The specified elements must be separated by commas and white space is optional; for example: `xci, xslt, sourceset`.

9) In the Data Encoding list, select one of the data encoding schemes:
   - UTF-8
   - UTF-16
   - Shift_JIS
   - Big5
   - GBK
   - KSC_5601

To add a button that submits an embedded PDF to a URL

1) Add a Button object to the form design. See *To add objects to a form design*.

2) In the Object palette, click the Field tab.

3) In the Control Type area, select Submit. Data will be submitted according to the settings in the Submit tab.

4) Click the Submit tab and type the URL protocol in the Submit to URL box.

To	Use this URL protocol	Example
Submit the package to an ftp site	ftp	ftp://ftp.gnu.org/gnu/GPL
Submit the package to a web server	http	http://myserver/cgi-bin/
Submit the package to a secure web server	https	https://myserver/cgi-bin/
Submit the package to an email address	mailto	mailto:username@domain.com

NOTE: When submitting data to a URL, it is recommended that you specify an absolute target. Relative targets are interpreted relative to the user environment, which can vary from one user to the next at run time.

5) In the Submit As list, select PDF.

This format submits a package containing an embedded PDF file.

Choose this format if the form contains a signature field, or if a copy of the form together with its data needs to be saved by Forms or submitted to another type of target server. Do not choose this option if the form initiates server-side processing, and if Forms will be used to render HTML or dynamic forms at run time. Keep in mind that if you want to submit data in PDF format, ensure that the Save usage right is applied to the form using Acrobat Reader DC extensions. No data can be submitted if the Save useage right is not applied to the form.

6) (Optional) To apply a signature to the submitted data, select Sign Submission in the Field tab and then click Settings to select optional signature settings.

7) (Optional) To apply XML encryption to the form content, select Encrypt Submission and then click Settings to select optional encryption settings.

To add an email submit button that submits an embedded PDF to an email address

1) Add an Email Submit Button object the form design. See *To add objects to a form design*.

2) In the Object palette, click the Field tab.

3) In the Email Address box, type the email address you want to submit the PDF to.

4) (Optional) In the Email Subject box, type a subject line for the email message.

5) In the Submit As list, select PDF.

This format submits a package containing an embedded PDF file.

Choose this format if the form contains a signature field, or if a copy of the form together with its data needs to be saved by Forms or submitted to another type of target server. Do not choose this option if the form initiates server-side processing, and if Forms will be used to render HTML or dynamic forms at run time. Keep in mind that if you want to submit data in PDF format, ensure that the Save usage right is applied to the form using Acrobat Reader DC extensions. No data can be submitted if the Save useage right is not applied to the form.

6) (Optional) To apply a signature to the submitted data, select Sign Submission in the Field tab and then click Settings to select optional signature settings.

7) (Optional) To apply XML encryption to the form content, select Encrypt Submission and then click Settings to select optional encryption settings.

To add a button that submits XML data to a URL

1) Add a Button object to the form design. See *To add objects to a form design*.

2) In the Object palette, click the Field tab.

3) In the Control Type area, select Submit. Data will be submitted according to the settings in the Submit tab.

4) Click the Submit tab and type the URL protocol in the Submit to URL box.

To	Use this URL protocol	Example
Submit the package to an ftp site	ftp	ftp://ftp.gnu.org/gnu/GPL
Submit the package to a web server	http	http://myserver/cgi-bin/
Submit the package to a secure web server	https	https://myserver/cgi-bin/
Submit the package to an email address	mailto	mailto:username@domain.com

NOTE: When submitting data to a URL, we recommend that you specify an absolute target. Relative targets are interpreted relative to the user environment, which can vary from one user to the next at run time.

5) In the Submit As list, select XML Data (XML).

This format submits an XML data stream, which allows for the hierarchical representation of data and can be parsed by any generic XML parser. Choose this format if the server that communicates with the run-time user application program must receive an XML data steam.

6) (Optional) To apply a signature to the submitted data, select Sign Submission in the Field tab and then click Settings to select optional signature settings.

7) (Optional) To apply XML encryption to form content, select Encrypt Submission and then click Settings to select optional encryption settings.

8) In the Encoding list, select one of the data encoding schemes:
   - UTF-8
   - UTF-16
   - Shift_JIS
   - Big5
   - GBK
   - KSC_5601

To add an email submit button that submits XML data to an email address

1) Add an Email Submit Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Email Address box, type the email address you want to submit the PDF to.
4) (Optional) In the Email Subject box, type a subject line for the email message.
5) In the Submit As list, select XML Data (XML).

This format submits an XML data stream, which allows for the hierarchical representation of data and can be parsed by any generic XML parser. Choose this format if the server that communicates with the run-time user application program must receive an XML data steam.

6) (Optional) To apply a data signature to the submitted data, select Sign Submission in the Field tab and then click Settings to select optional signature settings.
7) (Optional) To apply XML encryption to form content, select Encrypt Submission and then click Settings to select optional encryption settings.

To add a button that submits XML data to an e-mail address

1) Add a Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Control Type area, select Submit. Data will be submitted according to the settings in the Submit tab.
4) Click the Submit tab and, in the Submit to URL box, type the mailto protocol as shown in the following example:

   `mailto:username@domain.com`
5) In the Submit As list, select XML Data (XML).

   This format sends an XML data stream, which allows for the hierarchical representation of data and can be parsed by any generic XML parser. Choose this format if the server that communicates with the run-time user application program must receive an XML data steam.
6) (Optional) To apply a data signature to the submitted data, select Sign Submission in the Field tab and then click Settings to configure optional signature settings.
7) (Optional) To apply XML encryption to form content, select Encrypt Submission and then click Settings to configure optional encryption settings.
8) In the Encoding list, select one of the data encoding schemes:
   - UTF-8
   - UTF-16
   - Shift_JIS
   - Big5
   - GBK
   - KSC_5601

To add a button that submits a text stream using the POST method

1) Add a Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Control Type area, select Submit. Data will be submitted according to the settings in the Submit tab.
4) Click the Submit tab and type the URL protocol in the Submit to URL box.

To	Use this URL protocol	Example
Submit the package to an ftp site	ftp	ftp://ftp.gnu.org/gnu/GPL
Submit the package to a web server	http	http://myserver/cgi-bin/

To	Use this URL protocol	Example
Submit the package to a secure web server	https	https://myserver/cgi-bin/
Submit the package to an email address	mailto	mailto:username@domain.com

NOTE: When submitting data to a URL, it is recommended that you specify an absolute target. Relative targets are interpreted relative to the user environment, which can vary from one user to the next at run time.

5) In the Submit As list, select URL-Encoded Data (HTTP Post).

   This format submits a text stream to the specified Uniform Resource Locator (URL) by using the `POST` method. The text stream can be parsed by an FTP server, a mail server, or a web server. To use this method, users must open the form in Adobe Reader 6.0 or later or a web browser unless the URL specifies the mailto protocol.

6) In the Encoding list, select one of the data encoding schemes:
   - UTF-8
   - UTF-16
   - Shift_JIS
   - Big5
   - GBK
   - KSC_5601

To add an email submit button that submits encrypted form content to an e-email

1) Add an Email Submit Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Email Address box, type the e-mail address you want to send the submission to.
4) (Optional) In the Email Subject box, type a subject line for the e-mail.
5) In the Submit As list, select PDF or XML Data (XML).
6) Select Encrypt Submission, and click Settings.
7) In the Encrypt Data And Submit Settings dialog box, select options as needed. See *Encrypt Data and Submit Settings dialog box*.

To add a button that submits encrypted and encoded form content to an e-mail address

1) Add a Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Control Type area, select Submit. Data is submitted according to the settings in the Submit tab.
4) Click the Submit tab and, in the Submit to URL box, type the mailto protocol as shown in the following example:

```
mailto:username@domain.com
```

5) In the Submit As list, select either PDF or XML Data (XML).
6) Select Encrypt Submission, and click Settings.
7) In the Encrypt Data And Submit Settings dialog box, select options as needed. See *Encrypt Data and Submit Settings dialog box*.
8) In the Encoding list, select one of the data encoding schemes:
   - UTF-8
   - UTF-16
   - Shift_JIS
   - Big5
   - GBK
   - KSC_5601

To add an HTTP button that submits encrypted form content to a URL

1) Add an HTTP Submit Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the URL box, type the URL that the form data will be posted to.
4) Select Encrypt Submission, and click Settings.
5) In the Encrypt Data And Submit Settings dialog box, select options as needed. See *Encrypt Data and Submit Settings dialog box*.

Using check boxes

If you want to provide users with individual choices that are either turned on or off, add check boxes to the form. The user can toggle the state of a check box. When the check box is selected, its state is on. When the check box is deselected, its state is off. Check boxes can be turned on and off in any combination and may be used to set the properties of a selection.

After you add a check box to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the check box. See *Formatting captions*.
- Set a border style for the box. See *To set the border style*.
- Define the object as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the check box. See *To specify a locale (language and country or region) for an object*.
- Specify the size for the check box.
- Specify the check style.
- Specify the states for the check box.
- Select the default state of the check box.
- Specify values for the on, off, and neutral states of the check box.
- Specify a binding method for storing and retrieving bound data.

Check boxes support scripting and calculations. If a user is to supply data, you can define whether the input is recommended or required, and you can set up messages to prompt users appropriately. All user input may be validated through scripting.

You can bind check box objects to data elements in a source data file to derive the On and Off values from the source file when the form is rendered

To specify the size for the check box

1) To set the size for a check box, in the Object palette, click the Field tab.
2) In the Size box, type a different value (in points) and press Enter.
   Depending on the size you enter, you may have to resize the check box.

To specify the check style

1) In the Object palette, click the Field tab.
2) Select an option in the Check Style list:
   – To set the check style to the default, which is an X, select Default.
   – To set the check style to a check mark, select Check.
   – To set the check style to a circle, select Circle.
   – To set the check style to a cross, select Cross.
   – To set the check style to a diamond, select Diamond.
   – To set the check style to a square, select Square.
   – To set the check style to a star, select Star.

To specify the states for the check box

1) In the Object palette, click the Field tab.
2) Under States, select an option:
   – To specify that the check box will have On (selected) and Off (clear) states, select On/Off.
   – To specify that the check box will have On (selected), Off (clear), and Neutral (not selected or clear) states, select On/Off/Neutral.

   Values for each state must be defined in the Binding tab of the Object palette.

To define the behavior of the check box

1) In the Object palette, click the Value tab and select one of these options:
   – To allow users to choose whether to enter data, select User Entered - Optional.
   – To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users will not be able to edit the calculated value.
   – To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

– To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users will not be able to edit the value.

2) If the value will be calculated, attach the calculation script to the object by using the Script Editor.
3) (Optional) If a calculated value can be overridden, type a message in the Override Message box.

TIP: You can dynamically populate a validation pattern or script message with a value from a data source. This allows you to ensure that users enter the correct value in the field.

To specify the default state of the check box

1) In the Object palette, click the Value tab.
2) Select a state from the Default list:
   – To set the default state to On, select On. The check box is selected initially.
   – To set the default state to Off, select Off. The check box is deselected initially.
   – To set the default state to Neutral, select Neutral. The box is filled with grey initially.

NOTE: The Default list is available only when the Type option is set to User Entered or Read Only. The Neutral option is available from the Default list only when the States area in the Field tab is set to On/Off/Neutral.

To change the values assigned to check box states

By default, check boxes use numbers to represent values in the form data. The default values are “1” for On, “0” for Off, and “2” for Neutral. You can change these values to match existing values in a data source or to provide more meaningful terms related to the purpose of the check box in the data file.

1) In the Object palette, click the Binding tab.
2) In the On Value box, specify the value of the check box’s On state in the data source
3) In the Off Value box, specify the value of the check box’s Off state in the data source.
4) If On/Off/Neutral was selected in the Field tab, in the Neutral Value box, specify the value of the check box’s Neutral state in the data source.

To define custom data-binding properties for a check box

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the check box.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the check box to its corresponding data node.

For information about how to bind objects to a data source, see *Binding fields to a data source*.

RELATED LINKS:

Check box properties in the Field tab

Check box properties in the Value tab

Check box properties in the Binding tab

To dynamically populate a validation pattern message

To dynamically populate a validation script message

Using circles, lines, and rectangles

The drawing objects in the Object Library palette provide you with a way to enhance forms graphically.

- Using the Circle object, you can draw circular, elliptical, or arc shapes anywhere on the page.
- Using the Line object, you can draw solid, dashed, or dotted lines anywhere on the page.
- Using the Rectangle object, you can draw rectangles that have 90° or rounded corners. Use rectangles to graphically frame an area of the form or provide a rectangular area as a background for other objects.

After you add a drawing object to the form design, you can manipulate the object's properties in the Draw tab of the Object palette. You can define these properties:

- Change the line style.
- Change the thickness of the line.
- Select a line color.
- Specify the shape of the circle.
- Specify the start and end points of an arc.
- Fill the circle or rectangle with color or a pattern.
- Change the slope of the line.
- Change the corners of the rectangle.
- Make the object visible, invisible, or hidden.
- Define the object as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.

To change the line style

1) Select the circle, line, or rectangle.
2) In the Object palette, select a style from the Line Style list.
   You may have to zoom in or increase the thickness of the line to see the style.

To change the thickness of the line

1) Select the circle, line, or rectangle.
2) In the Object palette, in the box to the right of the Line Style, type a new thickness.
   If you type a number that makes the line very thick (for example, 1 in), then the line will be difficult to select. Use the Hierarchy palette to select the line.

To select a line color

1) Select the circle, line, or rectangle.
2) In the Object palette, click the color selector button and select a color.

To specify the shape of the circle

1) Select the circle.
2) In the Object palette, select the shape of the object from the Appearance list.
   – To draw an elliptical shape, select Ellipse.
   – To draw a circular shape, select Circle.
   – To draw an arc, select Arc.

To specify the start and end points of an arc

1) Select the arc.
2) In the Object palette, type the start and end points of the arc (in degrees) into the Start and Sweep boxes respectively.

To fill the circle or rectangle with color or a pattern

1) Select the circle or rectangle.
2) In the Object palette, select an option from the Fill list.
3) Using the adjacent color selector buttons, select appropriate fill colors.

To change the slope of the line

You can change the slope of the line. This is useful if you have drawn a line that is crooked. You can use this option to straighten the line.

1) Select the line.
2) In the Object palette, click one of the Appearance buttons .

To change the corners of the rectangle

1) Select the rectangle.
2) In the Object palette, click one of the Corners buttons.
3) In the Radius box, define the corner radius.

RELATED LINKS:

Rectangle properties in the Draw tab

Circle properties in the Draw tab

Line properties in the Draw tab

Layout properties in the Layout palette

Using content areas

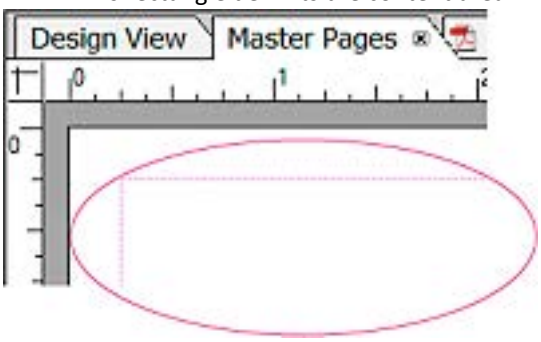
Content areas provide anchoring and layout management for all the objects in a form, including subforms.

It is usually unnecessary to modify the properties of a content area. However, if you are designing a form that contains subforms that are set to flow content, becoming familiar with the properties of content

areas will help you to understand the behavior of subforms. For information about the properties of content areas, see *Content area properties in the Content Area tab*.

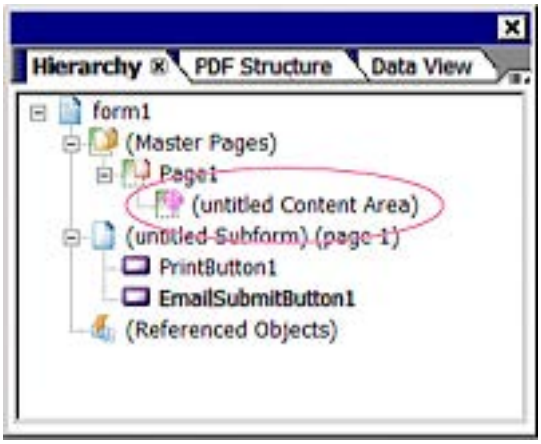
Content areas cannot be selected or manipulated through the Design View tab of the Layout Editor. If required, you can select, cut, copy, paste, move, delete, or resize content areas through the Master Pages tab of the Layout Editor. A rectangle on the master page delimits the area bounded by a content area.

This rectangle delimits the content area



New forms have a default content area. If required (for example, when designing a form that contains subforms set to flow content), you can add additional content areas to the form. All content areas are displayed in the Hierarchy palette.

Default content area



Data flow between content areas

When more than one content area has been defined, objects flow from one content area to the next when the form is rendered. You can select a flow direction for the objects and their data. The flow direction controls when an object receives the focus relative to the other objects in the form during a tabbing sequence. It also determines when an object receives merged data relative to the other objects in the form. Specifying the flow direction of one content area automatically sets the flow direction of all other content areas in the form.

When the form is rendered, if an object that merges data expands so much that it will no longer fit in one content area, the object spills into the next and subsequent content areas. Similarly, when you are designing a form, you cannot add more objects to a content area than will fit within the area bounded by the content area. As soon as the content area has been filled, the remaining objects overflow into the

next available content area. If another content area is not available to catch the overflow, the data will not be rendered properly.

To select a content area

You can select a content area using the Hierarchy palette or the Master Pages tab of the Layout Editor.

1) Select a content area by doing one of the following actions:
   – In the Layout Editor, click the Master Pages tab. Click the blue rectangle that delimits the content area.
   – In the Hierarchy palette, click the content area.

To rename a content area

- To rename a content area using the Hierarchy palette, right-click the object and select Rename Object. Type the new name and press Enter.
- To rename a content area using the Object palette, select the content area. In the Object palette, type a new name for the content area in the Name box and press Enter. Designer maintains the occurrence number automatically.

To define the tabbing and data-fill order for the subforms within the content area

1) In the Object palette, select one of these options from the Flow Direction list:
   – To set the default tabbing order and data-fill order starting from the top of the page and moving to the bottom of the page, select Top To Bottom.
   – To set the default tabbing order and data-fill order starting from the top of the page and moving to the right until the last object on the right edge of the page has been reached, select Western Text. When the right edge of the page has been reached, continue the tabbing order and flow direction at the next object down on the left side of the page.
   – To set the default tabbing order and data-fill order starting from the right of the page and moving to the left of the page, select Right to Left.

Copying, moving, and resizing content areas

You can cut, copy, paste, delete, resize, or reposition content areas through the Master Pages tab of the Layout Editor. You can only cut or delete content areas that you have added to a master page. A form must have at least one content area. The default content area cannot be deleted.

To cut, copy, paste, or delete a content area

1) In the Layout Editor, click the Master Pages tab. Right-click the content area and select the appropriate menu command.

To resize a content area

1) Resize the content area in the Master Pages tab of the Layout editor using one of these methods:
   – Click the border of the content area and drag any of the resizing handles to resize the object.
   – Select the content area, and then in the Layout palette, adjust the size settings.

To move a content area

1) In the Master Pages tab of the Layout editor, select the content area.
2) Move the content area using one of these methods:
   – Drag the content area to a new location.
   – Edit the **x** and **y** coordinates in the Layout palette.
   – Use an arrow key to move the content area slightly in one direction.

Adding content areas

You can add a content area using the Insert menu or the Object Library palette. Content areas can be added to master pages only.

To add a content area using the menu

1) With the master page of the form displayed, select Insert > Standard > Content Area.

To add a content area using the Object Library palette

1) Do one of the following actions:
   – In the Object Library palette, click the Standard category. Click the Content Area object and draw the object on the master page of the form.
   – From the Standard category of the Object Library palette, drag a Content Area object onto the master page of the form.
   – In the Object Library palette, click the Standard category. Double-click the Content Area object.

RELATED LINKS:

To name and rename objects

Layout properties in the Layout palette

Master page properties in the Master Page tab

Using date/time fields

If you want to provide the ability to record the date, the time, or both in a form, add a date/time field to the form. The value in a date/time field can be formatted and displayed according to predefined patterns and according to the locale specified for the field.

For example, you can add a date/time field to your form design so that the user can enter the date or select it from a calendar window.

After you add a date/time field to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the field. See *Formatting captions*.
- Set a border style for the field. See *To set the border style*.

- Define the field as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the field. See *To specify a locale (language and country or region) for an object*.
- Specify the comb format for the field. See *Using text fields*.
- Specify that Designer use the horizontal length of the text field to determine the maximum number of characters and numbers to allow.
- Define the display pattern.
- Define the edit pattern (which is the format in which the date is entered).
- Specify an initial default value to display.
- Define a run-time property (for example, insert the date and time on which the form is rendered).
- Define the validation pattern.
- Choose whether to display the date, the time, or both.
- Specify a data-binding pattern.
- Specify a binding method for storing and retrieving bound data.

Date/time fields support scripting and calculations. If a user is to supply data, you can define whether the input is recommended or required, and you can set up messages to prompt users appropriately. All user input may be validated through scripting.

RELATED LINKS:

Date/time field properties in the Field tab

Formatting field values and using patterns

Date/time fields

Date/time field properties in the Value tab

Date/time field properties in the Binding tab

To specify a default value

To dynamically populate a validation pattern message

To dynamically populate a validation script message

To add current date fields

You can use a date/time field to automatically show the current date by using the Runtime Property option in the Value tab of the Object palette.

NOTE: When using a current date field, the date updates to reflect the system's current date each time a user opens the form.

1) In the Object Library palette, click the Standard category and drag a Date/Time Field object onto the form design.
2) In the Object palette, click the Value tab and select Calculated - Read Only from the Type list.
3) Select the Runtime Property option.
4) View the form in the Preview PDF tab.

To define the behavior of date/time fields

1) In the Object palette, click the Value tab. Select one of these options:
   - To allow users to choose whether to enter data, select User Entered - Optional.
   - To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
   - To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
   - To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users cannot edit the calculated value.
   - To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.
   - To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users cannot edit the value.
2) If the value is recommended or required, type a prompt in the Empty Message box.
3) If the value will be calculated, attach the calculation script to the object by using the Script Editor.
4) (Optional) If a calculated value can be overridden, type a message into the Override Message box.

   TIP: You can dynamically populate a validation pattern or script message with a value from a data source. This allows you to ensure that users enter the correct value in the field.

To define custom data-binding properties for date/time fields

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the date/time field.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the field to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

To specify the data to display in date/time fields

You can choose to display the date, the time, or both.

1) In the Object palette, click the Binding tab.
2) From the Data Format list, specify what to display in the field:
   - **Date** Specifies the object as a date.
   - **Time** Specifies the object as a time.
   - **Date and Time** Specifies the object as a date and time.

To define the pattern for displaying date/time fields

1) In the Object palette, click the Field tab.
2) Click Patterns > Display.
3) In the Select Type list, select the pattern used to display data in the date/time field.

   NOTE: To display all date, time, and date/time formats in the Select Type list, you must first select Date and Time in the Data Format list of the Binding tab in the Object palette.

To specify the comb format for date/time fields

Use the comb format when you want to separate numbers added to date/time fields by border lines.

1) In the Object palette, click the Field tab.
2) Select Comb of Characters to includes border lines that separate each number within the date/time field.
3) Enter how many numbers will be separated by border lines within date/time text field.

To limit the numbers in date/time fields

You can limit how many numbers can be entered in date/time field by specifying that Designer use the horizontal length of the date/time field to determine how many numbers to allow.

1) In the Object palette, click the Field tab and select Limit Length to Visible Area.

Using decimal and numeric fields

Decimal fields are very similar to numeric fields. The following table describes the differences between decimal and numeric fields.

Field	Description
Decimal field	Displays numbers in decimal format. You can set the maximum number of leading and trailing digits. You can set a display pattern to restrict the formatted value with the leading and trailing digits. You cannot specify a data format for a decimal field; it is always stored as decimal. By default, the formatted value includes two digits following the radix character.
Numeric fields	Displays numbers in float or integer format. You cannot set the maximum number of leading and trailing digits. You can set a data pattern and data format (either float or integer). The formatted value includes only two digits following the radix character.

Use the decimal field in the following cases:

- If you want to set the number of leading or trailing digits
- If you want the precision of decimal or integer data to be shown exactly as it is entered (by ensuring the Limit Trailing Digits option is deselected) without any truncating

For example, a field called Qty is located within a repeating subform. In the first instance of the subform, the Qty field may have the value 3, and in the second instance of the subform, the Qty field may have the value 3.123:

- If the Limit Trailing Digits option is set to a maximum of 2, the first number is stored as 3 (because trailing 0s are removed) and the second number is stored as 3.12 (because the number is truncated).
- If the Limit Trailing Digits option is deselected, both numbers are stored exactly as they are entered (the first number is stored as 3 and the second number is stored as 3.123).

To learn more about the options for decimal fields, see the section on numeric fields because many of the options are identical.

NOTE: To use the decimal field object in an interactive form, you must have Adobe Reader 6.0.3 or later.

RELATED LINKS:

Numeric field properties in the Field tab

Decimal Field properties in the Field tab

Numeric patterns

Formatting captions

Formatting field values and using patterns

Numeric field properties in the Value tab

Numeric field properties in the Binding tab

To specify a default value

To dynamically populate a validation pattern message

To dynamically populate a validation script message

To specify a data pattern

About numeric fields

If you want to support the collection or display of float or integer data, including currency, add a numeric field to the form. The value in a numeric field can be formatted and displayed according to predefined patterns and according to the locale specified for the field.

After you add a numeric field to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the field. See *Formatting captions*.
- Set a border style for the field. See *To set the border style*.
- Define the field as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the field. See *To specify a locale (language and country or region) for an object*.
- Specify the comb format for the field. See *Using text fields*.
- Specify that Designer use the horizontal length of the text field to determine how many numbers to allow.
- Define the display pattern.

- Define the edit pattern.
- Specify an initial value to display.
- Define a run-time property (for example, insert the number of a rendered page).
- Define the validation pattern.
- Choose whether the field will support floating point or integer values.
- Specify a data-binding pattern.
- Specify a binding method for storing and retrieving bound data.

Numeric fields support scripting and calculations. If a user is to supply data, you can define whether the input is recommended or required, and you can set up messages to prompt users appropriately. All user input may be validated through scripting.

NOTE: Numeric fields have a maximum user-entered or calculated value of 2,147,483,647. This number is the largest number that can fit into a 32-bit signed integer on a computer.

Value ranges of XSD simple types mapped to numeric fields.

Type	Value Range
Boolean	True or False
byte	-128 to 127
unsignedByte	0 to 127
decimal, float	-1.7976931348623157e+308 to 1.7976931348623157e+308 Uses IEEE 754 double-precision format (also called as binary64).
integer, long, short	-2147483648 to 2147483647
unsignedShort, unsignedInt, positiveInteger, nonNegativeInteger, unsignedLong	0 to 2147483647
negativeInteger, nonPositiveInteger	-2147483648 to 0

RELATED LINKS:

Simple types

To define a radix (decimal point) alignment for numeric values

1) In the Paragraph palette menu, select Edit Value.
2) Click Radix Alignment .
3) In the adjacent box, type the amount of space to create between the radix point and the right edge of the fillable area.

4) Press Enter.

IMPORTANT: Acrobat 6.0.2 and Adobe Reader 6.0.2 do not support the radix alignment setting. If you intend to save the design as an Acrobat 6.0.2-compatible PDF form, do not use radix alignment. If you do, users cannot edit the values.

To define the behavior of decimal or numeric fields

1) In the Object palette, click the Value tab and select one of these options from the Type list:
   - To allow users to choose whether to enter data, select User Entered - Optional.
   - To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
   - To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
   - To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users cannot edit the calculated value.
   - To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.
   - To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users cannot edit the value.
2) If the value is recommended or required, type a prompt in the Empty Message box.
3) If the value will be calculated, attach the calculation script to the object by using the Script Editor.
4) (Optional) If a calculated value can be overridden, type a message in the Override Message box.

To define custom data-binding properties for decimal or numeric fields

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the field.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the field to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

To specify the data format for numeric fields

Numeric fields can save data in Float or Integer format. The default is Float.

By default, numeric data in a Decimal Field will only save a maximum of two digits after the decimal. Data beyond the second decimal place will be rounded.

1) In the Object palette, click the Binding tab.
2) Select the appropriate format from the Data Format list:

- To specify the number format as a three-part representation of a number that contains a radix character, select Float.
- To specify the number as any sequence of the digits 0 through 9, possibly preceded by a minus sign, select Integer.

To limit the numbers entered in decimal and numeric fields

You can limit how many numbers may be entered in decimal and numeric fields by specifying that Designer use the horizontal length of the decimal or numeric field to determine how many numbers to allow.

1) In the Object palette, click the Field tab and select Limit Length to Visible Area.

To perform a calculation with decimal fields by using JavaScript

You can perform calculations with decimal fields by using FormCalc. However, if the Limit Trailing Digits option is deselected, and you want to use JavaScript to perform a calculation with decimal fields, you must use a script. To maintain a high degree of precision, Designer stores the value of decimal fields as a string. The script indicates to the decimal field that it is a number.

For example, you have three decimal fields in your form design. The first is named `a`, the second `b`, and the third `Total`. You want to perform a calculation that adds `a` and `b`.

1) If the Script Editor is not already displayed, select Window > Script Editor.
2) (Optional) Drag the Script Editor palette bar until the palette is larger.
3) Select the decimal field that you want to use in a calculation.
4) Select Calculate from the Show list, JavaScript from the Language list, and Client from the Run At list.
5) In the Script Source field, type the following script:

```
Total.rawValue = Number(a.rawValue) + Number (b.rawValue)
```

   - Total is the name of the decimal field that you want to use in a calculation.
   - `a` is the first decimal field.
   - `b` is the second decimal field.

6) View the form in the Preview PDF tab.

To specify the comb format for decimal and numeric fields

Use the comb format when you want to separate characters and numbers added to decimal and numeric fields by border lines.

1) In the Object palette, click the Field tab.
2) Select Comb of Characters to includes border lines that separate each number within the decimal or numeric field.
3) Enter how many numbers will be separated by border lines within the decimal or numeric field.

Using extended decimal fields

You can use the extended decimal field to store and display decimal values larger than 15 and up to 31 digits. Values in an extended decimal field can have up to 14 trailing digits. Extended decimal fields use string data type to store values. You can set the maximum number of leading and trailing digits for a value in the extended decimal field. You can also specify data format (Object Window > Binding Tab > Data Format) for a value in the extended decimal field.

- If your number can contain more than 15 digits and less than 31 digits.
- If your number can contain more than 8 decimal digits (digits after the dot (.)) and less than 14 decimal digits.
- If you want to set the number of leading or trailing digits
- If you want the precision of decimal or integer data to be shown exactly as it is entered (by ensuring the Limit Trailing Digits option is deselected) without any truncating

You must have Adobe Reader DC or Acrobat DC version 19.010.20064.310990 or later to use the extended decimal field object in an interactive form.

An example of extended decimal field is, a field called Mass is located within a repeating subform. In the first instance of the subform, the Mass field may have the value 54372832348756945673, and in the second instance of the subform, the Mass field may have the value 54372832348756945673.567:

- If the Limit Trailing Digits option is set to a maximum of 2, the first number is stored as 54372832348756945673 (because trailing 0s are removed) and the second number is stored as 54372832348756945673.57 (because the number is truncated).
- If the Limit Trailing Digits option is deselected, both numbers are stored exactly as they are entered (the first number is stored as 54372832348756945673 and the second number is stored as 54372832348756945673.567).

Designer provides two objects or fields for the extended decimal functionality:

- **Standard Extended Decimal field**: You can drag and drop the Extended Decimal Field from **Object Library** > **Standard** panel to your form. It allows you to display extended decimal values. You can specify data format (Object Window > Binding Tab > Data Format) for the value of the field.
- **Custom ExtendedDecimal_JS object**: You can find the ExtendedDecimal_JS object in **Object Library**> **Custom** panel. Math functions of standard Java Script do not support values of an extended field. The Custom ExtendedDecimal_JS object provides a math library to perform calculation on extended decimal values.

  If you require an extended decimal field without any Java Script computation, use the Standard Extended decimal field. If you require Java Script computation, add the Custom ExtendedDecimal_JS object along with Standard Extended Decimal field. You can use the single ExtendedDecimal_JS object with multiple standard extended fields. See, section for information about performing calculations on extended decimal values using the Custom ExtendedDecimal_JS object.

To learn more about the options for decimal fields and extended decimal fields, see the section *Using decimal and numeric fields* on numeric fields because many of the options are identical. All the sections of decimal fields are applicable to extended decimal fields except the JavaScript calculation section.

Using drop-down lists and list boxes

Drop-down lists and list boxes provide users with a way to select a single choice from a list. The choices are represented by text. You define the list of default options.

Drop-down list	List box
Only one choice is visible until the user opens the list.	Any number of options can be visible at once. When the display area of the list box is too small to display all of the items in the list, a vertical scroll bar is displayed.
Accepts custom user entries.	Does not accept custom user entries.

Unlike check boxes and radio buttons, drop-down lists and list boxes support variable sizes of data and variable sets of choices. Clicking the drop-down arrow button displays the full set of choices, and clicking a choice in the list selects a single item. When the layout of the form prohibits the use of check boxes or radio buttons, consider using a list box to present options.

After you add a drop-down list or list box to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the list. See *Formatting captions*.
- Set a border style for the list. See *To set the border style*.
- Define the list as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the list. See *To specify a locale (language and country or region) for an object*.
- Add initial choices to the list and provide a default selection.
- Sort the choices in ascending or descending order.
- Enable or disable user input (drop-down lists only).
- Specify a locale for the list.
- Specify custom data values for each list item.
- Specify a binding method for storing and retrieving bound data.

TIP: You can dynamically populate a drop-down list or list box with values from a data source.

RELATED LINKS:

Working with Data Sources

To dynamically populate a drop-down list or list box

Drop-down list properties in the Field tab

Drop-down list properties in the Value tab

Drop-down list properties in the Binding tab

List box properties in the Field tab

List box properties in the Value tab

List box properties in the Binding tab

To dynamically populate a validation pattern message

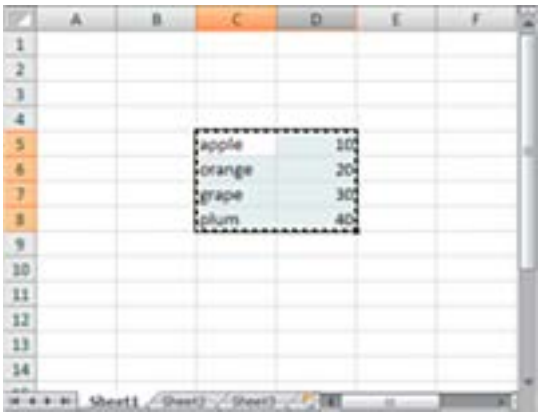
To dynamically populate a validation script message

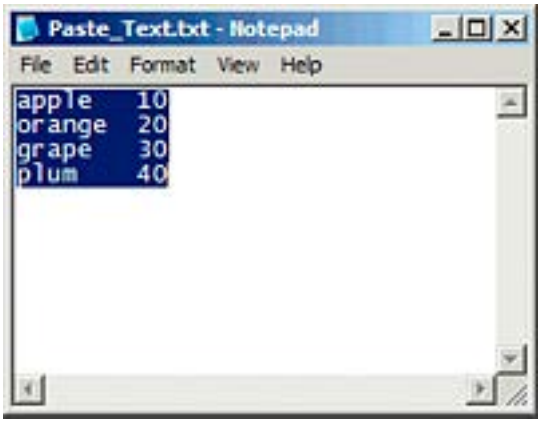
To specify the list of options for a drop-down list or list box

You can add a list of options to a drop-down list or list box by using either of these two ways:

- By copying the data from Microsoft Excel or a text editor
- Individually by using the Add Item button

The data that you are copying must be in tab-delimited format. You can copy and paste text or a text and value pair. If you choose to paste text, you place the text in a single column. If you want to paste a text and value pair, place the text in one column and the value in a second column. Here are examples of text and value pairs in Excel and Notepad. Always place the list items in the first column.





TIP: You can dynamically populate a drop-down list with values from a data source.

To add items from Excel or text editor

1) Select the drop-down list or list box.
2) Do one of the following actions:
   – In Excel, copy the spreadsheet cells that contain the text (or text and value pair) you want in the drop-down list or list box.
   – In a text editor, copy the text (or text and value pair) you want in the drop-down list or list box.
3) In the Object palette, click the Field tab, and then click Paste .

4) You can do any of the following actions:
   – To reorder an item in the list, select the item and click Move Up or Move Down .
   – To delete an item in the list, select the item and click Delete Item .
   – To sort the list in ascending order, select Sort Ascending .
   – To sort the list in descending order, select Sort Descending .
   – For a drop-down list only, to enable users to specify an item that does not appear in the list, select the Allow Custom Text Entry option.

To add items individually

1) In the Object palette, click the Field tab, and do one of the following actions:
   – Double-click below the Text column heading in the List Items box.
   – Click Add Item .
2) Type the text corresponding to the first option and press Enter.
3) Type the text corresponding to the next option and press Enter.
4) Repeat step 3 for each option that you want to add to the list.
5) You can do any of the following actions:
   – To reorder an item in the list, select the item and click Move Up or Move Down .
   – To delete an item in the list, select the item and click Delete Item .
   – To sort the list in ascending order, select Sort Ascending .
   – To sort the list in descending order, select Sort Descending .
   – For a drop-down list only, to enable users to specify an item that does not appear in the list, select the Allow Custom Text Entry option.

To define the behavior of a drop-down list or list box

1) In the Object palette, in the Value tab. From the Type list, select one of these options:
   – To allow users to choose to enter data or not, select User Entered - Optional.
   – To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
   – To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
   – To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users will not be able to edit the calculated value.
   – To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.
   – To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users will not be able to edit the value.

2) If the value is recommended or required, type a prompt into the Empty Message box.
3) If the value will be calculated, attach the calculation script to the object using the Script Editor.
4) (Optional) If a calculated value can be overridden, type a message into the Override Message box.

To display a default selection in the drop-down list or list box

You can set a default selection for the drop-down list or list box. You must first specify the list of options for a list.

1) In the Object palette, click the Value tab.
2) Select the default option from the Default list.

   NOTE: The Default list is not available when the Type option is set to Calculated - Read Only or Calculated - User Can Override.

   The options in the Default list correspond to the list items in the Field tab of the Object palette.

To define custom data-binding properties for a drop-down list or list box

Binding options enable you to build a form that captures data for enterprise infrastructures or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the drop-down list or list box.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the list to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

To specify list item values for a drop-down list or list box

Each item in a list is associated with a data value that represents the list item. By default, the values match the text for the list item. You can change these values, for example, to match existing values in a data source.

When you choose to specify item values for the first time, the values will change to integer values starting with “1” for the first item. If you deselect the Specify Item Values option, the values will revert to match the list item text.

TIP: You can dynamically populate a list box with values from a data source.

1) Select the drop-down list or list box object.
2) In the Object palette, click the Binding tab.
3) Select Specify Item Values.
4) Double-click the first value that you want to change.
5) Type the new value.
6) Press Enter to change the next value or double-click any other value to change it.

Using flash fields

Use the Flash Field object in interactive forms to deliver rich media content like embedded videos and data graphically displayed in charts.

NOTE: Flash Field objects are supported in Acrobat and Adobe Reader 10.0 or later. Remember to select this target version option on the Defaults panel in the Form Properties dialog box.

When you add a Flash Field object to a form, you specify the URL of the SWF file. You can also specify the location of an image file to use as a poster image. The poster image file is rendered any time the flash field is not in use or deactivated.

If you plan on using the form offline, you can embed flash data and poster image files in the form so that all components of the form are available. Keep in mind that embedding multiple SWF files and image files can significantly increase the size of the PDF file. To know more about the differences between embedding versus referencing objects, see this article.

You can display rich media content in a floating window, as well as add Flash assets like video, sound, image, text, XML, and SWC files. You can embed or reference Flash assets, or have a combination of both. You can also bind Flash Fields to a content provider such as an XML schema or a WSDL file.

In addition, you can configure various aspects of how rich media content is presented in a form. For example, in the Object palette you can define these properties:

- Change the caption for the field. See *Formatting captions*.
- Set a border style for the field. See *To set the border style*.
- Define the field as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the field. See *To specify a locale (language and country or region) for an object*.
- Position and position the field. See *Layout properties in the Layout palette*.

To specify the URL of a SWF file

1) In the Object palette, click the Field tab.
2) Do one of the following actions:
   - If you know the location of the SWF file, in the URL box, type the URL to the file and press Enter.
   - To browse to the location of the SWF file, click the Browse button to the right of the URL box, navigate to and select the SWF file, and click Open.

To embed flash data in the form

1) In the Object palette, click the Field tab and select Embed Flash Data.

   NOTE: If you are using the paper forms barcode object, do not select the Embed Flash Data option unless you have assigned a collection to the barcode, which does not include Flash Field objects. The embedded flash data is larger than the paper forms barcode object can hold.

To specify the location of a poster image

You can use BMP, JPG, GIF, PNG, or TIFF files as a poster image.

1) In the Object palette, click the Field tab.
2) Do one of the following actions:
   – If you know the location of the poster image file, in the Poster box, type the URL to the file and press Enter.
   – To browse for the location of the image file, click the Browse button to the right of the Poster box, navigate to the image file, and click Open.

To embed a poster image

1) In the Object palette, click the Field tab and select Embed Poster Image.

   NOTE: If you are using the paper forms barcode object, do not select the Embed Poster Image option unless you have assigned a collection to the barcode, which does not include Flash Field objects. The size of the embedded poster image file is larger than the paper forms barcode object can hold.

To display rich media content in a floating window

1) In the Object palette, click the Field tab and select Display Flash Content In Floating Window.
2) To configure the size and position of the floating window, in the Property list, enter values as needed. Values are measured in points. See *(Deprecated) Flash field properties in the Field tab.*

To add an additional asset

1) Click Edit.
2) Click Add (+).
3) In the Browse for Additional Asset dialog box, select the file you want to add, and click Open.
4) (Optional) To embed the file, select Embed Asset.
5) Click OK.

To unembed or delete an additional asset

1) Click Edit.
2) In the Additional Assets dialog box, do one of the following actions:
   – To unembed an asset, select the asset in the list and deselect Embed Asset.
   – To delete an asset, select the asset in the list and click Delete (+).
3) Click OK.

To define custom data-binding properties for a flash field

1) Select the flash field.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the check box to its corresponding data node. For information about how to bind objects to a data source. See *Binding fields to a data source*.

RELATED LINKS:

Flash field properties in the Field tab

Flash field properties in the Binding tab

Using image fields

Designer has two types of image objects.

- Image objects let you add a read-only image that users cannot edit on the form. This object is useful for enhancing the look of the form. For more information about this type of image object, see *Using images*.
- Image field objects let you add an image that can be changed in an interactive form.

Image fields can be bound to an external data source or they can be scripted. Image fields support the merging of external image data through binding. When the data source contains a value for an image field, the image is rendered in the form. With this capability, images can be selected and loaded dynamically. For example, a form can enable users to choose catalog items and, through scripting, a picture of each item could be displayed as a result. If the form contains an empty image field, the path to the image identifies which image to display in the form.

NOTE: Interactive images in image field objects can be updated when the form is opened in Acrobat and Adobe Reader 7.0.5 and later. Interactive images become read-only and cannot be updated when the form is opened in Acrobat 6.0.2 and Adobe Reader 6.0.2. However, through scripting in an interactive PDF or HTML form created by Forms, a user could request the same form again merged with a different image.

Image field objects support these file formats:

Windows Bitmap (BMP)

Designer supports BMP images.

Joint Photographic Experts Group variations (JPG)

Designer supports JPG images that include digital camera Exchangeable Image File (EXIF) data.

Graphics Interchange Format (GIF)

Although Designer supports GIF images, animated GIF files are not supported. Designer also supports transparency within GIF files to allow for overlaying images on forms.

Portable Network Graphics (PNG)

Designer supports one transparency color within PNG files to allow for overlaying images on forms.

Scalable Vector Graphics (SVG)

Designer supports vector image format for two-dimensional graphics with support for interactivity and animation. SVG supports graphic objects, such as vector graphic shapes, bitmap images, and text. You can group, style, transform and composite graphical objects into previously rendered objects. Designer supports the entire SVG feature set described in SVG 1.1 specifications (https://www.w3.org/TR/SVG/), except the following tags grouped by associate elements:

Tag Name	Description
Document Structure	
desc	Provides a mechanism to store the description of an element.
title	Provides a human readable name for container and graphics elements.
switch	Evaluates the requiredFeatures, requiredExtensions, and systemLanguage attributes on its direct child elements in order. Then, it processes and renders the first child for which these attributes evaluate to true.
Font	
glyph	Defines the graphics for a given glyph. The coordinate system for the glyph is defined by the various attributes in the 'font' element
font-face-uri	Points to a remote definition of the current font
Linking	
a	Used to create links in SVG images. SVG links work just like HTML links.
Clipping, Masking and Compositing	
mask	Used to quickly fade out an element using masks
Filter Primitive Elements	
felmage	Fetches image data from an external source and provides the pixel data as output (if the external source is an SVG image, it is rasterized)
Uncategorized Elements	
foreignObject	Allows for inclusion of a foreign XML namespace which has its graphical content drawn by a different user agent.
script	Equivalent to the script element in HTML and is the place for scripts
Animation Elements	
animate	Used to animate a single attribute or property over time
animateColor	Specifies a color transformation over time
animateMotion	Causes a referenced element to move along a motion path
mpath	Provides the ability to reference an external <path> element as the definition of a motion path

Tag Name	Description
set	Provides a simple means of just setting the value of an attribute for a specified duration
animateTransform	Animates a transformation attribute on a target element, allowing animations to control translation, scaling, rotation and/or skewing.

Tagged Image File (TIF)

Designer supports Monochrome (1 color component 1-bit depth), Greyscale (1 color component 8-bit depth), RGB (3 color components 8-bit depth), and Palette (1 color component 1,2,4,8-bit depths) TIF images.

An initial (default) image to insert at run time can be specified by either a file name or a Uniform Resource Locator (URL). Initially, the image is linked, which means that it is stored separately from the form and displayed when the form is opened. Alternatively, the image data can be embedded in the form when the form is created.

IMPORTANT: Linking to an image is not a secure way to protect sensitive business data. If the image data is considered sensitive, you should embed it in the form. Also, you can use a secure HTTPS connection in conjunction with a URL. However, HTTPS protects the transmission of the image data only while the data is being transmitted; it does not protect access to the image (that is, no authentication is performed).

After you add an image field to the form design, you can manipulate the object's properties in the Object palette. You can choose to define these properties:

- Change the caption for the field. See *Formatting captions*.
- Set a border style for the field. See *To set the border style*.
- Define the image as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the field. See *To specify a locale (language and country or region) for an object*.
- Embed the image in the form.
- Scale the image to size.
- Instead of or in addition to specifying a path to an initial (default) image file, specify a binding method for storing and retrieving bound image data.

Keep in mind that if the image file or a link to the image file is in the data file, the image only appears when the form is rendered and is not visible when you are designing the form. Also, if you plan on using the form offline, select the Embed Image Data option in the Field tab in the Object palette so that all components of the form are available.

RELATED LINKS:

Working around web browserlimitations

Image field properties in the Field tab

Image field properties in the Binding tab

To specify an initial (default) image to link

1) In the Object palette, click the Field tab
2) Do one of the following actions:
   – If you know the location of the image file, in the URL box, type the URL to the image file and press Enter.
   – To browse for the location of the image file, click the Browse button to the right of the URL box, navigate to the image file that you want to insert and click Open.

   NOTE: To use relative path names for retrieving linked images when the form is opened, the image files must be stored in a folder that is accessible to users. If Forms is available, the path must be relative to Forms.

To embed image data in the form when the form is created

1) In the Object palette, click the Field tab and select Embed Image Data.

   When you embed the image in the form, a copy of the image information is stored in the form. If the image field will be used to load images dynamically when the form is rendered, do not select the Embed Image Data option. To know more about the differences between embedding versus linking images, see this article.

   NOTE: If you are using the paper forms barcode object, you should not select the Embed Image Data option unless you have assigned a collection, that does not include image fields, to the barcode. The embedded image data is larger than the paper forms barcode object can hold.

To size an image

1) In the Object palette, click the Field tab.
2) In the Sizing list, select one of these options to determine what happens when the image is loaded:
   – To resize the image in the object, ensure that the aspect ratio of the image is preserved, and select Scale Image Proportionally.
   – To resize the image to match the dimensions of the object, select Scale Image to Fit Rectangle. The aspect ratio of the image is not preserved.
   – To preserve the actual size of the image, select Use Original Size. The image is not resized.

To define custom data-binding properties for an image field

You can use binding options to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time.

1) Select the image field.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the field to its corresponding data node. For information about how to bind objects to a data source, see "*Binding fields to a data source*.

Using images

If you want to enhance the way that a form looks, consider adding images to the form. You use image objects to contain images such as logos, icons, graphics, and photographs. An image object provides a graphic element that users cannot edit. Image objects support these file formats:

Windows Bitmap (BMP)

Encapsulated PostScript (EPS)

Joint Photographic Experts Group variations (JPG)

Designer supports JPG images that include digital camera Exchangeable Image File (EXIF) data.

Graphics Interchange Format (GIF)

Although Designer supports GIF images, animated GIF files are not supported. Designer also supports transparency within GIF files to allow for overlaying images on forms.

Portable Network Graphics (PNG)

Designer supports one transparency color within PNG files to allow for overlaying images on forms.

Tagged Image File (TIF)

Designer supports Monochrome (1 color component 1-bit depth), Greyscale (1 color component 8-bit depth), RGB (3 color components 8-bit depth), and Palette (1 color component 1,2,4,8-bit depths) TIF images.

The image to insert can be specified by either a file name or a Uniform Resource Locator (URL). By default, the image is linked, which means that it is stored separately from the form and displayed whenever the form is opened. You can also embed the image in the form.

IMPORTANT: Linking to an image is not a secure way to protect sensitive business data. If the image is considered sensitive, you should embed it in the form. Also, you can use a secure HTTPS connection in conjunction with a URL. However, HTTPS protects the transmission of the image data only while the data is being transmitted; it does not protect access to the image (that is, no authentication is performed).

After you add an image object to the form design, you can manipulate the object's properties in the Object palette. You can choose whether to define these properties:

- Embed the image in the form.
- Scale the image to size.
- Define the image as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.

RELATED LINKS:

Image properties in the Draw tab

Working around web browserlimitations

To specify the image to link

1) In the Object palette, use one of these methods:
   – If you know the location of the image file, in the URL box, type the URL to the image file and press Enter.
   – To browse for the location of the image file, click the Browse button to the right of the URL box, navigate to the image file that you want to insert and click Open.

   NOTE: To use relative path names for retrieving linked images when the form is opened, the image files must be stored in a folder that is accessible to users. If Forms is available, the path must be relative to Forms.

To embed an image into a form

1) In the Object palette, select Embed Image Data.

   NOTE: When you embed an image in the form, a copy of the image information is stored in the form.

To size the image

1) In the Object palette, click the Field tab.
2) In the Sizing list, select one of these options to determine what happens when the image is loaded:
   – To resize the image in the object, ensure that the aspect ratio of the image is preserved and select Scale Image Proportionally.
   – To resize the image to match the dimensions of the object, select Scale Image to Fit Rectangle. The aspect ratio of the image is not preserved.
   – To preserve the actual size of the image, select Use Original Size. The image is not resized.

About paper forms barcodes

A paper forms barcode electronically captures user-supplied data in an interactive PDF form. When an end user fills the form using Adobe Reader or Acrobat, the barcode is updated automatically to encode the user-supplied data. The user can then return the filled form by printing it and returning it by fax, mail, or hand. Upon receipt, the user-supplied data can be decoded using a scanning device.

To use the paper forms barcode object, your organization must have implemented Acrobat Reader DC extensions.

The barcoded forms solution provides a workflow where users completes the form using Adobe Reader and the data is automatically encoded into the barcode. After the form is printed and the paper copy returned to you, you can extract the form data by using a common barcode scanner. The result is 100% data accuracy, eliminating the need for rekeying data.

Using the paper forms barcode in a form eliminates the need for manual data entry or OCR-based forms processing. Data captured from fill-and-print paper forms can be reinserted into the electronic workflow quickly and accurately, with no loss of data caused by scanning or manual rekeying mistakes. Furthermore, you can retain a digital copy of the transaction, complete with ink signatures.

For example, a company has a PDF form that can be filled electronically. However, the form requires the user's signature; therefore, the completed form must be printed, signed, and returned by fax or mail. By including a paper forms barcode on the form, the data entered electronically is encoded into the barcode. When the printed form is returned, a barcode reader can read all of the captured data into a structured data file. Only the signature needs to be verified.

NOTE: The paper forms barcode object cannot be used on a form saved as an Acrobat 6.0-compatible PDF form. Users filling a form that contains a paper forms barcode require Acrobat 7.0 or later, or Adobe Reader 7.0 or later for PDF 417 barcodes, and Acrobat 7.0.5 or later, or Adobe Reader 7.0.5 or later for QR Code, DataMatrix, and GS1 DataMatrix barcodes in order for the barcode to encode the captured data.

RELATED LINKS:

To add objects to a form design

To control how a paper forms barcode encodes data

Paper forms barcode properties in the Field tab

Paper forms barcode properties in the Value tab

Layout properties in the Layout palette

Border properties in the Border palette

Accessibility properties in the Accessibility palette

How the paper forms barcode works

The paper forms barcode object is a two-dimensional (2D) barcode that is encoded with an industry standard PDF417, QR Code (Version 2), or DataMatrix (the ECC200 model) symbology. It includes an intelligent calculation script that encodes the data captured in a form's fields.

You select the form fields to be encoded by the paper forms barcode. However, because the barcodes have limited storage capacity, it is important that you select only required fields.

After you add a paper forms barcode to the form design, you specify its properties so that it behaves in the manner appropriate for your requirements. Using the Object palette, you can define these properties and others for the paper forms barcode:

- Barcode label
- Symbology encoded in the barcode
- Scanning method used to decode the paper forms barcode
- Error correction level, and the module width and height of the barcode (when you choose a custom scanning method)
- Object's visibility on the form
- Data compression options
- Form data to include in the barcode
- Data format that the paper forms barcode uses to encode data and the fields to be included

When you distribute your finished form, users fill the form, print it, and return it. After received, the user-supplied data is decoded by using a scanning device.

Consider the following points when designing forms that include the Paper Forms barcode object:

- The Paper Forms Barcode object encodes data that users type in a fillable PDF form. The use of paper forms barcodes for paper forms processing requires Acrobat Reader DC extensions.
- Users must use Acrobat or Adobe Reader 7.0.5 or later to fill a form using QR Code or Data Matrix Paper Forms Barcode. Therefore, form authors must not save the form as either Acrobat 7.0.5 Compatible or Acrobat 6.0.2 Compatible.
- Users must use Acrobat or Adobe Reader 7.0.5 or later to fill a form that includes the PDF417 Paper Forms Barcode.

  NOTE: To make use of the paper forms barcode, your organization must have implemented Acrobat Reader DC extensions.

Using paper forms barcodes

To specify a unique caption for the paper forms barcode

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Field tab, deselect Generate Label Automatically, and type a caption in the Label box.

To generate a caption for the paper forms barcode

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Field tab and select Generate Label Automatically.

To specify the symbology for the paper forms barcode

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Field tab and select one of the following options from the Symbology list:
   – PDF417
   – QR Code
   – Data Matrix

   NOTE: End users filling a form that contains a QR Code, DataMatrix, or GS1 DataMatrix barcode require Acrobat 7.0.5 or later, or Adobe Reader 7.0.5 or later. Decoding of QR Code and DataMatrix barcodes is supported by Barcoded Forms 7.0 but is not supported by barcoded forms 7.0 Stand-alone (ST).

To specify the paper forms barcode scanning method

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Field tab and select one of the following options from the Scanning Method list:
   – Hand-held Scanner

- Fax Server
- Document Scanner
- Custom

3) If you select Custom, you can also specify custom decoding settings.

To specify custom decoding settings for a paper forms barcode

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Field tab and select Custom from the Scanning Method list.
3) To change the decoding properties, select the value from the Value list beside the property.

   NOTE: If you are planning to decode the barcode by using a handheld barcode scanner, avoid creating barcodes wider than four inches. Taller and narrower barcodes generally work better with handheld scanners.

To control how a paper forms barcode encodes data

When using a paper forms barcode object in a form design, you must specify the format that the barcode will use when encoding the data, as well as which data will be encoded. You can specify the following types of data formats:

- XML
- Delimited

Both options provide a script for the paper forms barcode object that instructs it to use a particular data type when encoding the data.

You can provide your own script for encoding the data. For more information, see *To create a custom script for encoding data for a paper forms barcode*.

To define how the paper forms barcode object obtains data, use the options in the Value tab of the Object palette. You can also specify whether the data is compressed before encoding.

To automatically generate a script to encode barcode data

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Value tab and select Automatic Scripting.
3) In the Format list, select one of these options:
   - XML
   - Delimited.
4) (Optional) Select Include Field Names and Include Label.
5) In the Apply To list, select one of these options:
   - Entire Form Data
   - Collection Data. Select the collection from the Collection menu.

To compress the data before encoding

1) On the form design, select the paper forms barcode.
2) In the Object palette, click the Value tab and select Compress Data Before Encoding.

To create a custom script for encoding data for a paper forms barcode

In situations when you need to follow government or other specifications for encoding data, you may find that the preset data encoding options do not meet your requirements. Using the Custom option, you can provide your own script for encoding the data. The script could be quite simple, as in this example:

```
this.rawValue=NumericField1.rawValue
```

Alternatively, you could create a more complex script that determines which fields to encode based on certain conditions.

1) On the form design, select the paper forms barcode object.
2) In the Object palette, click the Value tab and deselect Automatic Scripting.
3) In the Script Editor, you can modify the sample custom script or write your own script.

For example, the following sample code illustrates the JavaScript script required to retrieve a form's field values, format a string representing a Comma Separated Value (CSV) instance containing the form fields, and update a paper forms barcode for an interactive PDF form. The form contains fields named`accountNum`, `formerFirstName`, `formerMiddleInitial`, `formerLastName`, `newFirstName`, `newMiddleInitial`, and `newLastName`.

```
// This function ensures that the barcode will update
// when a change is made to any field on the form.
function depends(node)
{
for (var i = 0; i < node.nodes.length; ++i)
{
var child = node.nodes.item(i);
if (child.isContainer)
depends(child);
}
}
// Return a field surrounded by quotes and followed with a separator
function fmtField(fieldName, separator)
{
  var str = "\"" + fieldName.rawValue + "\"" + separator;
  return str;
}
// Force all fields in the form to be updated in the dataset
depends(xfa.form);
// Generate the CSV string that will be encoded in the barcode
var comma = ",";
var newLine = "\n";
var s = fmtField(accountNum, comma);
s += fmtField(formerFirstName, comma)
s += fmtField(formerMiddleInitial, comma);
```

```
s += fmtField(formerLastName, comma);
s += fmtField(newFirstName, comma)
s += fmtField(newMiddleInitial, comma);
s += fmtField(newLastName, newLine);
// Assign the string to the barcode for encoding
this.rawValue = s;
```

To use legacy encoding format

By default, Designer uses a new encoding format that uses the XFA JavaScript function `xfa.record.saveXML()` to produce XML encoded data. However, you can still use the legacy encoding format that uses the old function `xfa.datasets.saveXML()`. For example, you may be processing forms with barcode data by using an application that can only parse the old XML format.

The legacy encoding format is used in paper forms barcode XML scripts in Designer 7.0 to 8.0. When you open a form that contains the legacy encoding format, the Use Legacy Format option is selected automatically.

1) On the form design, select the paper forms barcode object.
2) In the Object palette, click the Value tab.
3) In the Format list, select XML and then select Use Legacy Format.

To specify a delimiter

You can choose the delimiter character that is used to separate field data, such as field names, field values, and barcode labels.

With all delimiters other than Carriage Return, the first line starts with the optional barcode label heading, followed by the form object names, and ends with a delimiter and a new line character. The second line starts with the barcode label, which is a globally unique identifier (GUID) by default, followed by the form object data, and ends with a delimiter and a new line character.

With the Carriage Return delimiter, field names, barcode labels, and field values are listed over multiple lines, with field names and barcode labels listed first, each on a separate line, followed by either a carriage return, a new line character, and then with the field values, each of which are on a separate line.

If a field name or value in the form is empty, the barcode encoding script will insert a delimiter and encode the next field name or value. The barcode data will not contain a space, or any special marking, as the placeholder for the empty text.

NOTE: With all types of delimiters, you can control whether field names and barcode labels are included in the barcode content by selecting the Include Field Names and Include Label options in the Value tab of the Object palette.

To avoid incorrect decoding results when you choose the delimited format for a paper forms barcode, make sure that the delimiter you select is not a part of your form's textual contents. That is, do not choose a delimiter that may be a part of the paper forms barcode field value. For instance, if you choose tab, comma, or space as the delimiter, the contents of the form fields you are encoding cannot contain any tab, comma, or space characters because it will confuse the decoder.

You can always choose a different character as the delimiter if the form contents change and the character that was originally chosen to be the delimiter becomes a part of the contents.

NOTE: It is recommended that you do not select the Carriage Return option as a delimiter, because the carriage return character is already used in the paper forms barcode values as a delimiter to separate the header and value rows.

1) On the form design, select the paper forms barcode object.
2) In the Object palette, click the Value tab.
3) In the Format list, select Delimited and, in the Delimiter list, select the type of delimiter you want.

To specify character encoding

You can select the character encoding of the value that is encoded into a barcode.

1) On the form design, select the paper forms barcode object.
2) In the Object palette, click the Value tab.
3) In the Character Encoding list, select the option that suits your needs.

Design tips for using a paper forms barcode

Creating an effective paper forms barcode means designing a barcode so that information is encoded into it in a useful way that can be easily decoded into understandable information. Here are some things to keep in mind when designing a usable, decodable barcode.

Barcode design tip	Details
Design a barcode that is appropriate for its intended use.	Think about how the users will submit the form to you or your processing organization. Make sure the barcode is designed to survive the printing, mailing in, faxing, and scanning processes.
Use collections	Use Collections in paper forms barcodes instead of applying the barcode to the entire form. Collections ensure that you include the relevant fields in the barcode. ***NOTE:*** *If you add a List Box object with the Allow Multiple Selection option selected on the Field tab to a paper forms barcode collection, you must also select the Enforce Strict Scoping Rules in JavaScript option on the Defaults tab in the Form Properties dialog. Otherwise, any values that a form filler selects in the List Box object may not encode properly in the paper forms barcode.*
The Paper forms barcode collection name must be different from the object collection name.	Make sure that the paper forms barcode name is different from the object name. Otherwise, the paper forms barcode will not update at the run time.

Barcode design tip	Details
Paper forms barcodes in subforms with default names may cause unexpected results.	When you include a paper forms barcodes in multiple unnamed subforms, you may get unexpected results. Rename the subforms instead of using the default name.
Consider the placement of the barcode on the form design.	You can place a paper forms barcode anywhere on the form design. Always place the barcode where it can be seen and scanned easily. Keep barcodes at least a half-inch from the edge of the paper to ensure that they are included when the form is printed.
Size a barcode for the amount of data to be captured.	There is a limit to the amount of data that can be encoded into a paper forms barcode. The size of the barcode also affects the amount of data that it can encode. If the barcode is too small to hold the user-supplied data, a red marker appears on the upper-right corner of the barcode, with a message informing you that the data set to be encoded is too large for the barcode to contain. You should perform one of these steps: • Resize the barcode, making it bigger to accommodate more data. • Reduce the data set by changing the default data size for some form objects, such as text fields. For example, the default limit length of each text field is 255. However, if a text field will only contain a first name, last name, or even a full name, the limit could be reduced to 20 or 50 characters, potentially solving the barcode size issue. • Using extended characters and both uppercase and lowercase letters increases the size of the data. Write a script to restrict data to alphanumeric characters and uppercase letters. • If field names are included in the data, use shorter object names. • Use Form object collections.
Delimiter must not be part of textual content.	When you choose the delimited format for a paper forms barcode, make sure that the delimiter is not a part of your form's textual contents. For instance, if you choose Tab as the delimiter, the contents of the form fields you are encoding cannot contain tabs because it will confuse the decoder. To avoid this confusion and to ensure that the encoder will work correctly, if the character originally chosen to be the delimiter becomes a part of the contents, you can choose a different character (such as a pipe or a comma) as the delimiter instead.
Do not embed data for image fields.	Paper forms barcodes cannot hold enough data to embed data for images. If you have an image field object on the form, either do not select the Embed Image Data option for the image field object or exclude the object from the barcode data. If the image field is included in the barcode data, the barcode displays a warning that the data length cannot be calculated.

Barcode design tip	Details
Do not rotate Paper Forms barcodes.	Rotating a paper forms barcode will cause it to become clipped when the form is filled in Acrobat and Adobe Reader versions less than 8.0.
Test the form before distributing it.	It is critical that you test your completed form in Adobe Reader as well as in Acrobat. Most users of these types of forms will only have Adobe Reader installed.
Determine what version of Adobe Reader users will utilize to fill the form and ensure the correct right is applied to form.	When a form that contains one or more Paper Forms Barcode objects is opened in Adobe Reader 8.1 or later, the Paper Forms Barcode objects appear greyed-out when a user begins to fill the form, unless the 2DBarcode right has been applied to the form using Acrobat Reader DC extensions.
Ensure that users fill the form electronically	Make it clear to your users that they must fill the form electronically. If they print the form and then fill it, the paper forms barcode cannot encode any of the user-supplied data.
Test the maximum capacity of the paper forms barcode	You must test your forms in both Adobe Reader and in Acrobat to ensure that the capacity of each barcode is large enough to capture all of the required data. This can be done by filling each field with realistic entries of maximum length and ensuring that the barcode does not turn gray. You need to use Adobe Reader to test documents that have Acrobat Reader DC extensions barcode usage rights applied to them.

RELATED LINKS:

Using paper forms barcodes

To specify a unique caption for the paper forms barcode

Using form object collections

Using password fields

If you want to control access to a form through a password, add a password field to the form. When a user types in the password field, the characters are masked by a character that you select. If required, you can define a specific input pattern that the user must match to gain access to the form.

After you add a password field to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the field. See *Formatting captions*.
- Set a border style for the field. See *To set the border style*.
- Define the field as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the list. See *To specify a locale (language and country or region) for an object*.

- Specify that Designer use the horizontal length of the text field to determine how many characters to allow in the password field.
- Select the password display character.
- Define the edit pattern.
- Define the validation pattern.
- Specify a data-binding pattern.
- Specify a binding method for storing and retrieving bound data.

A password field can be present without forcing users to supply a password. You can define whether the password is optional, recommended, or required, and you can set up messages to prompt users appropriately. All user input can be validated through scripting.

RELATED LINKS:

Formatting captions

Formatting field values and using patterns

Password patterns

Password field properties in the Field tab

Password field properties in the Value tab

Password field properties in the Binding tab

To dynamically populate a validation pattern message

To dynamically populate a validation script message

To specify the password display character

1) In the Object palette, click the Field tab.
2) In the Password Display Character box, type the character that will be used to hide the password value when it is displayed at run time.

To define the behavior of the field

1) In the Object palette, click the Value tab and select one of these options from the Type list:
   – To allow users to choose whether to enter data, select User Entered - Optional.
   – To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
   – To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
2) If the value is recommended or required, type a prompt in the Empty Message box.

To define custom data-binding properties for a password field

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the password field.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the field to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

To limit the characters entered in password fields

You can limit the number of characters entered in password fields by specifying that Designer use the horizontal length of the decimal or numeric field to determine how many characters to allow.

1) In the Object palette, click the Field tab and select Limit Length to Visible Area.

Using radio buttons

Radio buttons provide a way to present users with a number of mutually exclusive choices. Users can toggle the state of any radio button in a group between on and off. When the radio button is selected, its state is On and its value is registered. When the radio button is cleared, its state is Off and its value is not registered. Only one radio button object in a group can be in the On state.

TIP: You can make a radio button look like a check box by setting its appearance characteristics in the Field tab of the Object palette. Before doing so, however, consider the expectations of users with respect to the behavior of check boxes. Do they expect check boxes to provide multiple choices? If so, you should maintain the standard and not apply radio button functionality to check box images.

After you add a radio button to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the radio button. See *Formatting captions*.
- Set a border style for the radio button. See *To set the border style*.
- Define the radio button as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the radio button. See *To specify a locale (language and country or region) for an object*.
- Assign a value to each radio button's On state in the exclusion group.
- Specify the style for the radio button.
- Specify the default selection for the group.
- Specify a binding method for storing and retrieving bound data.

Radio buttons support scripting and calculations. If a user is to supply data, you can define whether the input is recommended or required, and you can set up messages to prompt users appropriately. User input may be validated through scripting.

An *exclusion group* is a group of radio buttons. Only one radio button in the exclusion group can be selected at a time.

You can work with an exclusion group as one object. Some options on the Object palette apply to all of the radio buttons in the same exclusion group. You can move radio buttons between exclusion groups and create new exclusion groups. If you have several radio buttons that are part of the same exclusion group, you can easily break some of the objects out into another exclusion group.

When you add a new radio button to the form, if the last object you added to the same subform was also a radio button, the new button will be part of the same exclusion group. If the last object you added was not a radio button, the radio button will create a new exclusion group.

Exclusion groups automatically resize to fit all of the radio buttons within the group. If you drag a radio button to an empty area of the page, the exclusion group will expand to contain the radio button. If you manually resize the exclusion group, all of the contained radio buttons will be resized accordingly.

RELATED LINKS:

Formatting captions

To make radio buttons accessible

Radio button properties in the Field tab

Radio button properties in the Value tab

Radio button properties in the Binding tab

Working with Data Sources

To set the size for a radio button

1) In the Object palette, click the Field tab.
2) In the Size box, type a different value (in points) and press Enter.
   Depending on the size you enter, you may have to resize the radio button.

To specify the radio button style

1) In the Object palette, click the Field tab.
2) Select an option in the Button Style list:
   - To set the check style to the default, which is a filled circle, select Default.
   - To set the button style to a check mark, select Check.
   - To set the button style to a circle, select Circle.
   - To set the button style to a cross, select Cross.
   - To set the button style to a diamond, select Diamond.
   - To set the button style to a square, select Square.
   - To set the button style to a start, select Star.

To assign On values to radio buttons

Each radio button in an exclusion group is associated with an On value that represents the selected radio button in the form data. You can change these values, for example, to match existing values in a data source.

When you specify On values, the default values are integer values starting with “1” for the first radio button in the exclusion group that was added to the form. If you deselect the Specify Item Values option, the On values will match the radio button caption text.

1) Select the exclusion group or one of the radio buttons within the group.
2) In the Object palette, click the Binding tab and select Specify Item Values.
3) Double-click the first On value that you want to change.
4) Type the new On value.
5) Press Enter to change the next On value or double-click any other On value to change it.

To define custom data-binding properties for radio buttons

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties for radio buttons in the Binding tab of the Object palette.

1) Select the group.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the exclusion group to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

To create a new exclusion group

1) Select the radio buttons that you want to make part of a separate exclusion group in the Layout Editor or the Hierarchy palette.
2) Do one of the following actions:
   – Right-click one of the selected objects in the Layout Editor and select Wrap In New Radio Button Group.
   – Right-click one of the selected objects in the Hierarchy palette and select Wrap In New Radio Button Group.

   NOTE: After you add a group of radio buttons, add a different object to the page. (For example, add a text object.) Now, any new radio buttons you add to the form design start a new exclusion group.

To move a radio button to a different exclusion group

1) Select the radio button that you want to move to a different exclusion group in the Layout Editor or the Hierarchy palette.
2) Do one of the following actions:
   – To move the radio button on the form, drag the object in the Layout Editor to a position within the borders of another exclusion group.

– To change exclusion groups without moving the button on the page, drag the radio button to another exclusion group in the Hierarchy palette.

To move an exclusion group

1) Click the border of the exclusion group to select it.
2) Drag the group to a new position on the page. All of the radio buttons in the group will move together.

To merge exclusion groups

1) Click the border of the exclusion groups to select them.
2) Select Layout > Merge Radio Button Groups.

To define the behavior of the exclusion group

1) In the Object palette, click the Value tab and, in the Type list, select one of these options:
   – To allow users to choose whether to enter data, select User Entered - Optional.
   – To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
   – To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
   – To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users cannot edit the calculated value.
   – To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.
   – To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users cannot edit the value.
2) If the value is recommended or required, type a prompt in the Empty Message box.
3) If the value will be calculated, attach the calculation script to the object by using the Script Editor.
4) (Optional) If a calculated value can be overridden, type a message in the Override Message box.

To specify the default selection for the exclusion group

1) In the Object palette, click the Value tab.
2) In the Default list, select the value that identifies the radio button that you want to set to the On state.

NOTE: The Default list is available only when the Type option is set to User Entered or Read Only. The value of a radio button's On state can be changed in the Binding tab.

Using signature fields

If you want to authenticate the identity of a user and the document's content, add a signature field to the form. A digital signature stores information about the signing party and the state of the document when it is signed. Adobe signatures support the Public Key Cryptography Standard (PKCS) #7, using the RSA MD5, RSA SHA-256, DSA SHA-256, or DSA SHA-384 hash algorithm.

You can specify whether a signature covers an entire form or a collection of objects in a form. If you want the signature to apply to a collection of objects, the signing party must use Acrobat or Adobe Reader version 8.0 or later.

If the signature covers a collection of objects, the fields are locked and cannot be modified after the document is signed. The lock is inherited. For example, when a subform is locked, all the objects in the subform inherit the lock.

To sign the form, the user clicks the signature field. For more information about signing PDF forms in Acrobat, see *Acrobat Help*.

The signature field does not generate a digital signature directly. It invokes a third-party signature handler through scripting, and the signature handler provides the required digital signature functionality. Users cannot sign forms without an appropriate signature handler.

If required, you can change the signature field name in the Hierarchy palette after you add the object to the form design.

After you add a signature field to the form design, you can define the signature properties in the Signature tab of the Object palette. You can define whether the signature covers the entire form or a collection of field objects and specify default signature properties.

You can edit the caption text and manipulate the object’s properties in the Field tab of the Object palette. You can define these properties:

- Change the caption for the field. See *Formatting captions*.
- Set a border style for the field. See *To set the border style*.
- Define the field as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the field. See *To specify a locale (language and country or region) for an object*

RELATED LINKS:

Formatting captions

Layout properties in the Layout palette

Border properties in the Border palette

Drop-down lists

Using digital signatures

Using text

Text objects present read-only text that users cannot edit. You can use text objects to do these tasks:

- Label an area in the form
- Provide instructions for filling out the form

- Include a header and footer

After you add a text object to the form design, you can define these properties:

- Define the text as visible, invisible, or hidden. See *Making objects visible, invisible, or hidden*.
- Specify a locale for the text. *To specify a locale (language and country or region) for an object*.
- Type and style the text.
- Allow page breaks within the content of the field.
- Keep a text object with the next object in the document when a page break is introduced.
- Apply font properties to all of the text or selected portions of the text.

In addition, you can insert placeholders, such as floating text fields, and the values of run-time properties, such as page count, current page, and current date/time, into text objects.

The text object has the Make Fixed-Sized Text Objects Auto-fit While Editing option (Tools > Options > Workspace) selected by default. Therefore, the object automatically resizes at design time to accommodate the text you enter and the property settings you change, such as margins, borders, and font type and size.

To type characters into a text object

You can type characters into the object after you add the object to the form.

By default, the Make Fixed-Size Text Objects Auto-Fit While Editing option on the Workspace panel of the Tools > Options dialog is selected. This option allows the width and height of text objects to expand as you type characters. To expand only either the width or height of a text object, deselect Make Fixed-Size Text Objects Auto-Fit While Editing and then select the appropriate Expand To Fit option on the Layout palette.

1) Select Edit and ensure that the Lock Text and Lock Static Objects commands are not selected.
2) Double-click the text object and type the characters you want to add.

To insert a floating text field into a text object

You can embed a floating text field in a text object. Floating text fields support the merging of different text values within text objects. For example, you could embed a floating text field to generate a customer name in the salutation of a form letter.

A floating text field does not have a caption and can be edited as if it were a single character of text. The properties of the floating text field must be defined separately in addition to the properties of the text object.

NOTE: When merging floating fields with data, keep in mind that floating fields cannot span pages.

1) In the text object, place the cursor where you want the floating text field to be merged.
2) Select Insert > Floating Field.
3) Select the floating text field and edit its properties.

To insert the value of a run-time property into a text object

You can insert the values of run-time properties into a text object. When the form is viewed, the actual value is inserted automatically.

The inserted value can have a different font, font size, and font style than the text object.

1) In the text object, place the cursor where you want the value of the run-time property to be merged.
2) Select Insert > *[run-time property value]*.
   - To insert the current page number of the finished form, select Current Page Number.
   - To insert the total number of pages making up the finished, select Number of Pages.
   - To insert the current date/time, select Current Date/Time.
   - To insert the value of the locale setting for the application processing the form, select Viewer Locale.
   - To insert the name of the application processing the form, select Viewer Name.
   - To insert the version number of the application processing the form, select Viewer Version.
3) Select the run-time property value and edit its properties.

To merge text objects

When you import a PDF file into Designer, depending on the import options you choose, title and instruction text may be broken up into multiple text objects. This import behavior makes the text difficult to edit because individual words or individual letters may be broken into separate text objects. If you need to edit the text, you can merge text objects together to make the text easier to update.

1) Select the text objects that you want to merge.

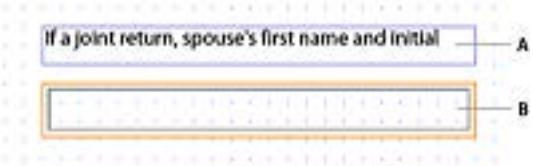
   If you also select an object of a different type, it will not be included in the merged object.
2) Select Layout > Merge Selected Text Objects.

   All selected text objects combine into a single object.

To create a caption by merging a text object and a field object

You can create a caption for a field object that does not have a caption by merging it with a text object.

For example, if you have a text object above a text field whose caption is set to none, you can merge the two objects to create one field. The text from the text object becomes the caption for the text field.



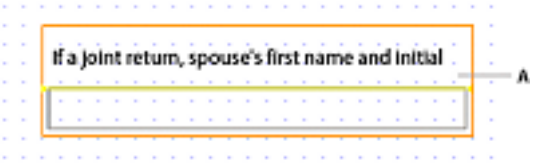
Before Merging: A.

Text object

B.

Text field

After Merging: Text object



Before you perform this task, you must ensure the following settings are in effect:

- Ensure you have one text object.
- Ensure you have one field object whose caption is set to none in the Layout palette or the caption is empty.

The field object can be one of the following objects:

- Check Box
- Date/Time Field
- Decimal Field
- Signature Field
- Drop-down List
- Image Field
- List Box
- Numeric Field
- Password Field
- Radio Button
- Text Field

You can merge only one text object to one field object.

1) Select the text object that you want to become the caption for the field object.
2) Ctrl+click or Shift+click to select the field object that you want to merge with the text object.

   The field object must have its caption set to none in the Layout palette or the caption must be empty.
3) Select Layout > Merge as Caption.

   The new object's caption becomes the text from the text object. The new object's size and position matches the total boundaries of the merged objects.

   You may have to edit the font of the caption and the value after you merge the objects.

To insert language-specific numbers in text or captions

Most regions of the world use a numbering system that is often called Roman or European (1, 2, 3, ...). However, the numbering system in Arabic is different; it is derived from Hindi or Indic numbers.

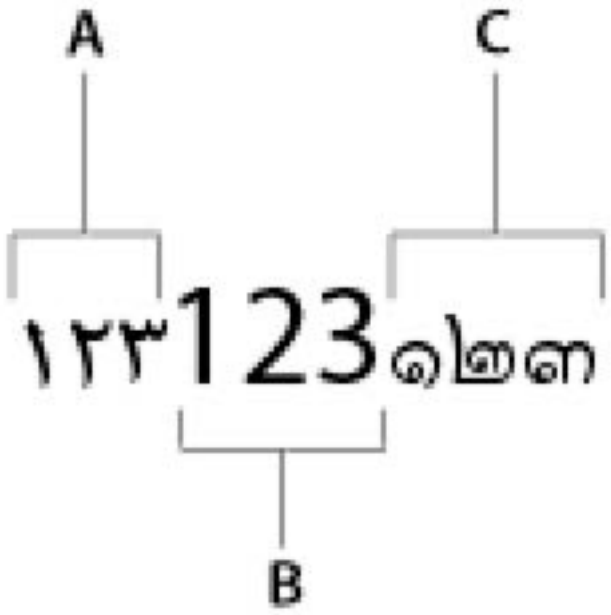
Although Arabic letters are written from right-to-left, numbers in Arabic are written from left-to-right. This numbering system is often called Arabic-Indic.

You can insert language-specific numbers in text or captions if your locale, such as Arabic (Egypt), supports it. The following locales support this feature:

- Arabic (all except Algeria, Morocco, and Tunisia)
- Thai (Thailand Traditional)

The options described in this Help topic are available only if support for the appropriate language is enabled through Microsoft Office Language Settings.

1) Select the text.
2) Right-click and ensure the language-specific Use Digits command is selected.
   If this menu item is not there, ensure the locale set for this object supports this feature.
3) Type the numbers.
   This example is of a caption that shows three types of language-specific numbers.
   A. Arabic (Egypt) **B.** English (USA) **C.***Thai (Thailand Traditional)*



To allow page breaks within a text object

The Allow Page Break Within Content option is enabled for a text object if the parent subform allows page breaks. This option is deselected by default.

Here are a few things to consider when allowing page breaks within a text object:

- The Allow Page Break Within Content option is enabled for a text object only when the parent object allows page breaks.
- Page breaks are not supported in rotated text objects.
- The Allow Page Breaks Within Content option is disabled for floating fields. Instead, select the Allow Page Break Within Content option for the text object referencing the floating field.
- A page break is not allowed when an object that can accept a page break is positioned beside an object that cannot.

- When the Make Fixed-Size Text Objects Auto-Fit While Editing option (in Tools > Options > Workspace) or the Expand to Fit options (in the Layout palette) are deselected, the boilerplate text does not expand at design time, which means that page breaks are prevented.

To allow page breaks within a text object

1) In the Object palette, click the Draw tab and select Allow Page Breaks Within Content.

To keep a text object with the next object in the form

The Keep with Next option is enabled for a text field if the parent object allows page breaks and is a flowed container.

The Keep With Next option is disabled for floating fields. Instead, select the Keep With Next option for the text object referencing the floating field.

1) In the Object palette, click the Draw tab and select Keep With Next.

RELATED LINKS:

Layout properties in the Layout palette

Border properties in the Border palette

Font properties in the Font palette

Paragraph properties in the Paragraph palette

Text properties in the Draw tab

To make objects expand to fit

Using date/time fields

Using text fields

Formatting captions

Using text fields

If you want to provide users with a way to enter textual data, add text fields to the form. Text fields enable users to type, select, edit, cut, copy, paste, and delete any of the text inside the field.

Text fields can accept one or more lines of wrapping text and support enhanced formatting capabilities. For example, text fields can display blocks of text in different typefaces and colors. Text can be stored and retrieved as plain text or XHTML. If required, the data in a text field can be formatted and displayed according to predefined patterns.

After you add a text field to the form design, you can edit the caption text and manipulate the object's properties in the Field, Value, and Binding tabs of the Object palette. You can define these properties:

- Change the caption for the field. (See *Formatting captions*.)
- Set a border style for the field. (See *To set the border style*.)
- Define the field as visible, invisible, or hidden. (See *Making objects visible, invisible, or hidden*.)
- Specify a locale for the field. (See *To specify a locale (language and country or region) for an object*.)
- Specify the comb format for the field.

- Enable or disable multiple lines of text.
- Enable or disable the field to be filled with rich-formatted text.
- Limit the number of characters in the field.
- Allow page breaks within the content of the field.
- Keep a text field object with the next object in the document when a page break is introduced.
- Define the display pattern.
- Define the edit pattern.
- Specify an initial value to display.
- Define a run-time property (for example, insert the page number).
- Define the validation pattern.
- Choose whether the field will support plain text or XHTML.
- Specify a data-binding pattern.
- Specify a binding method for storing and retrieving bound data.

Text fields support scripting and calculations. If a user is to supply data, you can define whether the input is recommended or required, and you can set up messages to prompt users appropriately. All user input may be validated through scripting.

RELATED LINKS:

Formatting captions

To lock objects

Formatting field values and using patterns

Text field patterns

Text field properties in the Field tab

Text field properties in the Value tab

Text field properties in the Binding tab

To specify a default value

To dynamically populate a validation pattern message

To dynamically populate a validation script message

To allow multiple lines of text in text fields

1) In the Object palette, click the Field tab and select Allow Multiple Lines.

To limit the number of characters in text fields

You can limit the number of characters in text fields in one of two ways:

- You can specify the maximum number of characters allowed.
- You can specify that Designer use the horizontal length of the text field to determine how many characters to allow.

To specify the number of characters in text fields

1) In the Object palette, click the Field tab.
2) Select the Limit Length and, in the Max Chars box, type the maximum number of characters that users will be permitted to enter into the text field.

To limit the number of characters to the width of text fields

1) In the Object palette, click the Field tab and select Limit Length to Visible Area.

Allowing page breaks within a text field

The Allow Page Break Within Content option is enabled by default for a text field if the parent subform allows page breaks.

Here are a few things to consider when allowing page breaks within a text field:

- The Allow Page Break Within Content option is available only if the parent object allows page breaks.
- The caption text of a text field object will not break between pages when Top or Bottom is selected for the position for the caption in the Field tab of the Object palette.
- Page breaks are not supported in rotated text field objects.
- A page break is not allowed when an object that can accept a page break is positioned beside an object that cannot.

To allow page breaks within a text field

1) In the Object palette, click the Field tab and select Allow Page Breaks Within Content.

To keep a text field with the next object in the form

The Keep with Next option is enabled for a text field if the parent object allows page breaks and is a flowed container.

The Keep With Next option is disabled for floating fields. Instead, select the Keep With Next option of the text field object referencing the floating field.

1) In the Object palette, click the Field tab and select Keep With Next.

To enable the field to be filled with rich-formatted text

By default, a text field is set to display and capture plain text. In this case, if the data value associated with the text field includes text formatting information, the formatting is ignored and the data value is displayed as plain text. You can set it so that any rich-text formatting of the data is preserved.

1) In the Object palette, click the Field tab and select Rich Text from the Field Format list.

To define the behavior of the field

1) In the Object palette, click the Value tab and, in the Type list, select one of these options:
   – To allow users to choose whether to enter data, select User Entered - Optional.

- To prompt users to enter data and make the field recommended, select User Entered - Recommended and type a custom message in the Empty Message box.
- To prompt users to enter data and make the field required, select User Entered - Required and type a custom message in the Empty Message box.
- To make the field read only and display a data value that is calculated and displayed through an attached script, select Calculated - Read Only. Users cannot edit the calculated value.
- To make the field editable and display a data value that is calculated and displayed through an attached script, select Calculated - User Can Override. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.
- To make the field read only and display a data value that is merged or calculated and displayed at run time, select Read Only. Users cannot edit the value.

2) If the value is recommended or required, type a prompt in the Empty Message box.
3) If the value will be calculated, attach the calculation script to the object by using the Script Editor.
4) (Optional) If a calculated value can be overridden, type a message in the Override Message box.

To define custom data-binding properties for text fields

Binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. Set data-binding properties in the Binding tab of the Object palette.

1) Select the text field.
2) Enable the form to connect to the data source when the form is opened.
3) Bind the field to its corresponding data node. For information about how to bind objects to a data source, see *Binding fields to a data source*.

To specify the data format for text fields

Text fields can save data in plain text only or with XHTML text formatting information included as part of the value. The default is plain text. If you change the data format to XHTML, the Rich Text option in the Field Format list in the Field tab of the Object palette is automatically selected.

1) Select the text field object.
2) In the Object palette, click the Field tab.
3) Select the appropriate format from the Field Format list.

To specify the comb format for text fields

Use the comb format when you want to separate characters and numbers added to text fields by border lines.

1) In the Object palette, click the Field tab.
2) Select Comb of Characters to includes border lines that separate each character within the text field.
3) Enter the number of characters that will be separated by border lines within the text field.

Using form object collections

You can select only the required objects to be covered by a signature field or encoded in a paper forms barcode, and save them as a collection.

A collection applies to either signature fields or to paper forms barcodes, but not to both, because the objects are represented differently. The collections that are available for signature fields appear in the Signature tab in the Object palette. The collections that are available for paper forms barcodes appear in the Value tab in the Object palette.

RELATED LINKS:

Using digital signatures

Using paper forms barcodes

To add a signature field

Signature field properties in the Signature tab

Design tips for using a paper forms barcode

Paper forms barcode properties in the Value tab

Collection Editor dialog box

Collection List dialog box

Signature field properties in the Field tab

To create a form object collection

You can create a collection for signature fields or for paper forms barcodes. You can create one or many collections and select the one you want to use for a particular signature field or paper forms barcode.

NOTE: If you add a List Box object with the Allow Multiple Selection option selected on the Field tab to a paper forms barcode collection, you must also select the Enforce Strict Scoping Rules in JavaScript option on the Defaults tab in the Form Properties dialog. Otherwise, any values that a form filler selects in the List Box object may not encode properly in the paper forms barcode.

1) Select File > Form Object Collections.
2) Click New and type a name for the collection in the box.
3) Click Modify and select the appropriate option from the Collection Type List. The list displays the available objects for the selected collection type.
4) Select the objects to include in the collection. To hide unsupported objects for the type of collection, select Hide Unsupported Nodes.
5) Click OK.

To create a form object collection

You can create a collection for signature fields or for paper forms barcodes. You can create one or many collections and select the one you want to use for a particular signature field or paper forms barcode.

1) On the form design, select the signature field or paper forms barcode.
2) Do one of the following actions:

- To create a collection for a paper forms barcode, in the Object palette, click the Value tab, select Collection Data from the Apply To list and select New/Manage Collection from the Collection list.
- To create a collection for a signature field, in the Object palette, click the Signature tab, select Lock Fields After Signing, and select <New/Manage Collection>.

3) Click New and type a name for the collection in the box.
4) Click Modify and select the appropriate option from the Collection Type List.
5) Select the objects to include in the collection. To hide unsupported objects for the type of collection, select Hide Unsupported Nodes.
6) Click OK.

To copy a form object collection

1) Select File > Form Object Collections.
2) Select a collection from the list and click Duplicate.
3) Double-click the collection and type a name.
4) To specify the position of the collection, click Move Up or Move Down.

To copy a form object collection

1) On the form design, select a signature field or a paper forms barcode.
2) Do one of the following actions:
   - To copy a collection for a paper forms barcode, in the Object palette, click the Value tab, select Collection Data from the Apply To list, and select New/Manage Collection from the Collection list.
   - To copy a collection for a signature field, in the Object palette, click the Signature tab, select Lock Fields After Signing, and select <New/Manage Collection>.
3) Select a collection from the list and click Duplicate.
4) Double-click the collection and type a name.
5) To specify the position of the collection, click Move Up or Move Down.

To rename a form object collection

1) Select File > Form Object Collections.
2) Double-click the collection in the list and type the new name.

To rename a form object collection

1) On the form design, select a signature field or a paper forms barcode.
2) Do one of the following actions:
   - To rename a collection for a paper forms barcode, in the Object palette, click the Value tab, select Collection Data from the Apply To list, and select New/Manage Collection from the Collection list.

– To rename a collection for a signature field, in the Object palette, click the Signature tab, select Lock Fields After Signing, and select New/Manage Collection.

3) Double-click the collection in the list and type the new name.

To modify a form object collection

1) Choose one of these methods:
   – Select File > Form Object Collections and go to step 3.
   – Select a signature field or paper forms barcode object on the page.
2) Do one of the following actions:
   – To modify a collection for a paper forms barcode, in the Object palette, click the Value tab.
   – To modify a collection for a signature field, in the Object palette, click the Signature tab.
3) In the Collection list, select New/Manage Collection.
4) Select the collection that you want to modify and click Modify.
5) Select the appropriate option from the Collection Type List.
6) Select the objects to remove or include in the collection. To hide unsupported objects for the type of collection, select Hide Unsupported Nodes.
7) Click OK.
8) To specify the position of the collection, click Move Up or Move Down.

To modify a form object collection

1) On the form design, select a signature field or a paper forms barcode.
2) Do one of the following actions:
   – To modify a collection for a paper forms barcode, in the Object palette, click the Value tab.
   – To modify a collection for a signature field, in the Object palette, click the Signature tab.
3) In the Collection list, select New/Manage Collection.
4) Select the collection that you want to modify and click Modify.
5) Select the appropriate option from the Collection Type List.
6) Select the objects to remove or include in the collection. To hide unsupported objects for the type of collection, select Hide Unsupported Nodes.

To delete a form object collection

1) Select File > Form Object Collections.
2) Select a collection from the list, click Delete, and then click Yes.

To delete a form object collection

1) On the form design, select a signature field or a paper forms barcode.
2) Do one of the following actions:
   – To delete a collection for a paper forms barcode, in the Object palette, click the Value tab.

– To delete a collection for a signature field, in the Object palette, click the Signature tab.

3) In the Collection list, select New/Manage Collection.
4) Select a collection from the list, click Delete, and then click Yes.

Using custom objects

In addition to the objects in the Standard category of the Object Library palette, Designer includes some predefined custom objects, which are configured to provide functionality that form authors frequently need in a form solution. These custom objects have properties and scripts that you can adjust to suit your requirements.

The following predefined custom objects are located in the Custom category of the Object Library palette.

Object	Description
Address Block	Accepts and displays addresses in United States Postal Service format. The Address Block is a group of text fields that lets users enter their name, address, city, state, zip code, and country. Designed for any type of form.
Process Fields	A block of objects designed for Process Management forms. It includes a submit button, an action drop-down list, and other hidden fields used to transport data needed by Process Management to route the form data correctly.
Countries	Displays a drop-down list populated with country names. You can add, move, and delete any country from the list. Designed for interactive forms.
Current Date	A date/time field with a script that displays the current date according to the viewer's system locale. Designed for any type of form.
Data Drop-down List	A drop-down list with a script that populates the list from an OLEDB data source. Designed for interactive forms.
Data List Box	A list box with a script that populates the list from an OLEDB data source. Designed for interactive forms.
Email Address	A text field that captures and validates an email address. Designed for interactive forms.
ExtendedDecimal _JS	A script object. It provides a math library to perform standard math operations on Extended Decimal Fields.

Object	Description
Form Bridge	A hidden object that enables communication to the PDF outside of Acrobat.
Name	A group of text objects that accepts and displays name information. End users can enter their last name, first name, and initial. For any type of form.
Page n of m	A text field with a script that displays the current page and total page count of the form at run time. Example: Page 1 of 30 For any type of form.
Page Navigation	A group of buttons with scripts to navigate to first, last, previous, and next pages. Designed for interactive forms.
Phone Number - UK	A text field that formats input as a United Kingdom (UK) telephone number. Designed for interactive forms.
Phone Number - North America	A text field that formats input as a North American telephone number. Designed for interactive forms.
Sheet n of m	Displays a text object that indicates the value of the current piece of paper (sheet) within the range of sheets required for a form. Example: Sheet 1 of 5 For any type of form.
Signature - Print and Sign	Displays a space on a printed form where a user can provide a signature. Designed for print forms.
Survey Question	A text object for a survey question and a group of radio buttons for selecting the answer. Designed for interactive forms.
U.S. Social Security Number	A text field that lets a user enter a United States social security number. Designed for interactive forms.
U.S. States	Displays a drop-down list populated with U.S. state names. You can add, move, and delete any state from the list. Designed for interactive forms.

Most of the custom objects are ready-to-use, requiring only minor changes such as changing the captions, formatting numbers, or changing the font type to meet your requirements. Simply drag the custom objects onto the form design and make the changes. When you place a custom object on the

form design, notice that it also appears in the Hierarchy palette. For custom objects that consist of several standard objects formatted in a particular manner and grouped together, such as the address block custom object, you will see the individual text field objects (Name, Address, City, State, Zip Code, Country) in the Hierarchy palette.

Two of the custom objects, the data drop-down list and the data list box, consist of a single object that has a custom script associated with it. That script enables you to populate two columns with data from an OLEDB data connection. When you place the object on the form design, you can see the script in the Script Editor. The script includes comments that explain which modifications are needed to make the script compatible with your particular run-time environment. For more information, see *To dynamically populate a drop-down list or list box from an OLEDB data connection*.

The Process Fields object is a special object that is designed strictly for use with forms workflow.

RELATED LINKS:

To add objects to a form design

Object Library palette

Managing library palettes

Objects that support scripting and calculations

Objects that support scripting and calculations

Layout properties in the Layout palette

Border properties in the Border palette

Font properties in the Font palette

Paragraph properties in the Paragraph palette

About custom objects for the SAP environment

You can use this predefined set of custom objects in the SAP environment. The custom objects appear in the Object Library palette.

Object	Description
Check Fields	A button with a script that sends the form's data to the SAP system for verification. Designed for interactive forms built for Adobe Reader 8.0 or later. Requires the Web Dynpro environment.
Enumerated Drop-down List	A drop-down list where the list of data values is supplied in the data. It uses data binding values to locate a list of elements in the data source. This object selects the first item in the list if the default value is not found. Designed for interactive forms built for Acrobat 8.0 or later and for use in the Web Dynpro environment. For more information, see *To dynamically populate a drop-down list in the Web DynPro environment*. You can set up dynamic properties for drop-down lists and list boxes. (See *To dynamically populate a drop-down list or list box*.)

Object	Description
Decimal Field Value Help	The Decimal Field combined with Value Help. Value Help contains a script that queries the SAP system for help in filling a field. Designed for interactive forms built for Adobe Reader 8.0 or later.
Enumerated Drop-down List (no select)	A drop-down list (no select) where the list values come from the data. It uses data binding values to locate a list of elements in the data source. This object does not select the first item in the list if the default value is not found. Designed for interactive forms built for Adobe Reader 8.0 or later and for use in the Web Dynpro environment. For more information, see *To dynamically populate a drop-down list in the Web DynPro environment*. You can set up dynamic properties for drop-down lists and list boxes. (See *To dynamically populate a drop-down list or list box*.)
Hidden Bound Text Field	A hidden field that contains script for detecting changes in unbound form data for forms opened in Adobe Reader 7.0.5 or earlier.
Launch URL	A button that contains a script that launches a hard-coded URL. Designed for interactive forms. For more information, see *About URL links*.
Localized Friendly Date	A date/time field that contains a script for entering the date in three date formats. Users can enter dates in the short, medium, and long date formats. The format of the field depends on the default locale. For example, if the locale is set to English (U.S.), the fields follows the M/D/YY|MMM D, YYYY|MMMM D, YYYY format. Example of short date display for U.S. display: 1/6/08 Example of medium date display for U.S. display: Jan 6, 2008 Example of long date display for U.S. display: January 6, 2008 Designed for interactive forms.
Localized Full Date	A date/time field that contains a script for entering the date in the full date format. The format of the field depends on the default locale. For example, if the locale is set to English (U.S.), the field follows the EEEE, MMMM D, YYYY format. Example of U.S. display: Thursday, January 6, 2008 Designed for interactive forms.
Localized Long Date	A date/time field that contains a script for entering the date in the long date format. The format of the field depends on the default locale. For example, if the locale is set to English (U.S.), the field follows the MMMM D,YYYY format. Example of U.S. display: January 6, 2008 Designed for interactive forms.

Object	Description
Localized Medium Date	A date/time field that contains a script for entering the date in the medium date format. The format of the field depends on the default locale. For example, if the locale is set to English (U.S.), the field follows the MMM D, YYYY format. Example: Jan 6, 2008 Designed for interactive forms.
Localized Short Date	A date/time field that contains a script for entering the date in the short date format. The format of the field depends on the default locale. For example, if the locale is set to English (U.S.), the field follows the M/D/YY format. Example: 1/6/08 Designed for interactive forms.
Numeric Field Value Help	The Numeric Field combined with Value Help. Value Help contains a script that queries the SAP system for help in filling a field. Designed for interactive forms built for Adobe Reader 8.0 or later.
RFID Fields	RFID label support for hexadecimal data that enables binary data as hexadecimal values in RFID fields. You can use hexadecimal data for Zebra RFID.
Set Fields Properties	A script object that sets the properties of grouped fields. Designed for any type of form on the server side. For more information, see *To group interactive fields and set their properties*.
Subform Instance Controls: Add	Lets a user add new input fields to a form on the client. A button with a script that creates a new instance of a repeating subform. For example, use this object on an order form for a number of customers. The customers can use Adobe Reader to add as many line items as they need. The script consists of a single button with a `click` event specified to run at the client. See the script for instructions. Designed for forms whose layout adjusts to accommodate data. For more information, see *To add repeating fields*.
Subform Instance Controls: Insert Remove Move	Lets a user insert, remove, and move new input fields on a form on the client. The object consists of four buttons, each containing a script. The scripts contain `click` events specified to run at the client. See the script for instructions. For example, use this object on an order form for a number of customers. Using Adobe Reader, the customers add, remove, and move as many line items as needed. Designed for forms whose layout adjusts to accommodate data. For more information, see *To add repeating fields*.
SubmitToSAP	A button that contains a script that submits the form data to the SAP system. Designed for interactive forms built for Adobe Reader 8.0 or later.

Object	Description
SubmitSuccessClose	A hidden field that contains a script that closes the PDF form when a user successfully submits form data. This script executes when new data arrives from the server. Designed for interactive forms. For more information, see *About setting a PDF form action*.
SubmitSuccessReadOnly	A hidden field that contains a script that sets all fields to read-only when an end user successfully submits form data. This script executes when new data arrives from the server. Designed for interactive forms. For more information, see *About setting a PDF form action*.
TableCalcs	A script object that counts, adds, and averages fields in a table on any number of pages. For more information, see *To use the TableCalcs custom object*.
Text Field Value Help	The Text Field combined with Value Help. Value Help contains a script that queries the SAP system for help in filling a field. Designed for interactive forms built for Adobe Reader 8.0 or later.
Value Help	A script that queries the SAP system for help in filling a field. You must customize this script. Designed for interactive forms built for Adobe Reader 8.0 or later. Requires the Web Dynpro environment.
Value Help Drop-down List	A drop-down list field that is enabled so that users can make a Value Help request to the SAP server. Designed for interactive forms built for Adobe Reader 8.0 or later. Requires the Web Dynpro environment.

To create a custom object

To create a custom object

Designer enables you to create your own custom objects that you can save and reuse. Creating a custom object involves adding one or more objects from the Standard tab to the form design and then defining the properties of the objects to suit your requirements.

You can save the object in the Custom category of the Object Library palette, or you can create your own category in the Object Library palette and save the object there. Creating your own Object Library categories is a good way to organize the custom objects that you create and keep them separate from the predefined custom objects provided with Designer. For information about creating your own categories, see *To add a category to the Object Library palette*.

1) In the Object Library palette, drag the object that you want to customize onto the form design.
2) Define the properties of the object.
3) In the Object Library palette, click the Custom category.

4) Drag your object to the Custom category in the Object Library palette.
5) In the Add Library Object dialog box, type a name and description for the custom object, and click OK.

NOTE: Multiple objects can be saved as a single custom object. When you drag multiple objects to the Custom tab together, a single object is created in the Custom tab.

10.3. Adding, copying, and deleting objects

To add objects to a form design

You can add objects to a form design in several ways. After you add an object, you can define the object's properties.

To add an object using the Insert menu

1) Do one of the following actions:
   – Select Insert > Standard > *[object]*.
   – Select Insert > Barcodes >*[object]*.
   – Select Insert > Custom >*[object]*.

To add an object from the Object Library palette

1) Do one of the following actions:
   – Drag the object onto the form design.
   – Double-click the object.
   – Click the object and drag in the form design to draw the object.

To add several objects of the same type

1) Do one of the following actions:
   – In the Object Library palette, click the object and then select View > Keep Drawing. You can use the Keep Drawing command for any object. To disable this feature, select View > Keep Drawing again or press Escape.
   – Click the Draw Fields button in the Tools toolbar and then select the object from the drop-down list. The Draw Fields button is only available for those objects in the drop-down list. To disable this feature, click the Select Tool or the Hand Tool button in the Tools toolbar.

   When these commands are selected, you can insert as many copies of an object as you require without selecting the object from the Object Library palette each time.

To cut, copy, and paste objects

You can cut, copy, and paste objects by using the menu or keyboard shortcuts.

Duplicating an object is the equivalent of copying and pasting an object. By default, the duplicate is placed just below and to the right of the original. If you drag the duplicate to a new position and then duplicate it, the next duplicate and any subsequent duplicates are placed the same distance and direction away from the last duplicate. You can make as many duplicates as will fit in the active content area.

To cut or copy and paste an object

1) Select one or more objects.
2) Do one of the following actions:
   – To cut an object, select Edit > Cut.
   – To copy an object, select Edit > Copy.
3) Select Edit > Paste.

If the pointer is on a page, the object is pasted beneath the pointer. Otherwise, the object is pasted in the upper left corner of the active content area.

To copy a group of objects

1) In the Hierarchy palette, select the group node.
2) Select Edit > Copy.

To make multiple copies of one or more objects

1) In the form design or in the Hierarchy palette, select one or more objects.
2) Select Edit > Copy Multiple.
3) Type a number in the Number Of Copies box. You can make as many copies as will fit in the active content area.
4) Select options for Vertical Placement and Horizontal Placement.
5) To increase the amount of vertical space between the copied and original objects, in the Vertical Spacing area, select Offset By and type the distance in the adjacent box.
6) To increase the amount of horizontal space between the copied and original objects, in the Horizontal Spacing area, select Offset By and type the distance in the adjacent box.
7) Click OK. The copies are added to the page and all of the objects, both original and copied, are selected.

To duplicate objects

1) Select one or more objects.
2) Select Edit > Duplicate.

To duplicate an object by dragging it

1) Drag the object to the spot where you want to place the duplicate.
2) Press Ctrl and release the mouse button.

To delete objects

You can delete any selected object.

1) Select one or more objects.
2) Do one of the following actions:
   – Select Edit > Delete.
   – Press Delete.

10.4. Selecting, grouping, and moving objects

RELATED LINKS:

Keys for selecting text

Keys for manipulating objects

Using subforms

Layout properties in the Layout palette

To select objects

You can select an object by using familiar mouse and keyboard methods:

- To select a single object, click the Select Tool button in the toolbar and click the object.
- To select multiple objects, Ctrl+click each of the objects you want to select or lasso (drag the pointer over) the objects you want to select.
- To select all objects, select Edit > Select All.
- To select all static objects, select Edit > Select All Static Objects.
- To select all fields, select Edit > Select All Fields.

To select objects using the Hierarchy palette

The Hierarchy palette assists with the selection and manipulation of objects.

When you select an object in the Hierarchy palette, the object is selected in the form simultaneously. However, when you make a selection in the Hierarchy palette, all keystrokes (for example, F2 or Shift+click) are directed to the Hierarchy palette until you work in the form again.

- To display the Hierarchy palette, select Window > Hierarchy.
- To select a single object in the Hierarchy palette, click the object.

- To select multiple adjacent objects in the Hierarchy palette, click the first object and then Shift+click the last object.
- To select multiple non-adjacent objects in the Hierarchy palette, Ctrl+click each of the objects.
- To select a group of objects in the Hierarchy palette, click the group node.

NOTE: If you select an item in the Hierarchy palette and click the Design View tab, the object is selected on the form design, and you can also use the keyboard to manipulate the selected object.

To group and ungroup objects

You can group and ungroup objects by using the Layout menu, the Hierarchy palette, or the Layout toolbar:

- To group objects, in the form design or Hierarchy palette, select the objects you want to group and then either select Layout > Group or click Group .
- To ungroup objects, in the form design or Hierarchy palette, select the group of objects you want to ungroup and then either select Layout > Ungroup or click Ungroup .
- To add a new object to a group by using the mouse, in the form design, drag the object to the group.
- To add a new object to a group by using the Hierarchy palette, drag the object to an existing group node.

To group interactive fields and set their properties

You can use the Set Fields Properties custom object to group multiple interactive fields and set their properties. The properties of the groups of interactive fields are enabled when a user opens the PDF form. For example, in a guided procedure, you could have a three-step form where the first step contains visible and editable areas, and the second and third steps remain read-only.

You can set these properties on groups of interactive fields on a PDF form:

Visible

Makes the fields visible.

Invisible

Makes the fields invisible.

Open

Lets users edit or modify the fields.

Not Editable

Makes the fields read-only so that users cannot edit them.

Mandatory

Users must fill the fields before submitting the form.

Optional

Users can submit the form without filling the fields.

The Set Fields Properties object consists of a script object nested inside an untitled subform. The script object called *groupedFields* defines a function called `setProperty`.

See the object's script for more details.

To use the object, a form developer must write a function call to invoke the object's `setProperty` function:

1) In the Object Library palette, click the category that contains the Set Fields Properties object and drag it onto the existing form design.
2) Write a script that invokes the function of the script object. Here is the list of function calls to set the status of grouped fields:

```
groupedFields.setProperty(this.parent, "presence", "visible")
groupedFields.setProperty(this.parent, "presence", "invisible")
groupedFields.setProperty(this.parent, "access", "open")
groupedFields.setProperty(this.parent, "access", "readOnly")
groupedFields.setProperty(this.parent, "mandatory", "disabled")
groupedFields.setProperty(this.parent, "mandatory", "error")
```

   NOTE: The SOM expression may vary depending on the position of the script object in the form hierarchy.
3) Save, preview, and test the form in the Preview PDF tab. You may want to enable the JavaScript Debugger to catch any errors while testing the form. See "JavaScript Debugger in Acrobat Professional" in *Scripting Using Designer* for more information about the debugger.

   NOTE: An invalid property value will not generate an error message.

To move objects

You can move objects by dragging them with the mouse, using the arrow keys on the keyboard, or entering precise values in the Layout palette.

To move or duplicate objects by pasting

1) Select one or more objects.
2) Select Edit > Cut to move the selection or Edit > Copy to duplicate the selection.
3) (Optional) To paste an object into another file, open the file.
4) Select Edit > Paste to paste the object into the active window.

To move an object by dragging it

1) Select one or more objects.
2) Drag the object to a new location.

To move an object by using the arrow keys

1) Select one or more objects.
2) Press the arrow key for the direction in which you want to move the object. The default distance is 1 point (1/72 of an inch, or .3528 millimeter).
3) To move the object 10 points at once, press Shift+arrow.

To move objects by using x and y coordinates in the Layout palette

1) Select one or more objects.
2) In the Layout palette, enter new values in either the X or the Y box, or both.

To stack objects

Designer stacks successively drawn objects, beginning with the first object drawn. How objects are stacked determines how they appear when they overlap. You can change the stacking order of objects in your form design at any time by using the arrange commands in the Layout menu.

1) Select the object.
2) Use one of these methods:
   - To bring the object forward, select Layout > Bring Forward, or click Bring Forward .
   - To bring the object to the front, select Layout > Bring To Front, or click Bring To Front .
   - To send an object backward, select Layout > Send Backward, or click Send Backward .
   - To send an object to the back, select Layout > Send To Back, or click Send To Back .

To lock objects

Locking the objects in a form design prevents you from selecting, moving, and editing the objects in the Layout Editor.

You can lock the following parts of your form design:

Text

Includes text in a text object or text in the captions of objects. When you select Lock Text, you must double-click the appropriate text or caption to edit it. This makes it not as easy to modify the text. If Lock Text is not enabled, you click in the caption or text object to start editing the text. Using Lock Text is particularly useful if you want to select and move objects but not make any changes to the text. You can single-click the object to select it, but you must double-click the caption or text to make changes.

Static objects

Includes text, circles, lines, and rectangles.

Field objects

Includes buttons, check boxes, date/time fields, decimal fields, signature fields, drop-down lists, image fields, list boxes, numeric fields, paper forms barcodes, password fields, radio buttons, and text fields.

After locking static objects and field objects, you must unlock them to add new static objects and field objects.

The lock commands are useful in situations when you have finished formatting specific objects in the form design and want to continue working with other types of objects. For example, your form includes a large number of text field objects, as well as text objects that provide instructions about filling the form. You have positioned and formatted all the text objects and need to apply a different font to the text field objects. You can easily do this by locking the text objects and then using Edit > Select All Fields to select all the text field objects at once and then change the font.

- To lock text, select Edit > Lock Text. To unlock text, select Edit > Lock Text.
- To lock static objects, select Edit > Lock Static Objects. To unlock static objects, select Edit > Lock Static Objects.
- To lock fields, select Edit > Lock Fields. To unlock fields, select Edit > Lock Fields.

To manipulate objects in a form design by selecting them in the Hierarchy palette

If you have multiple overlaid objects in a form design, you can use the Hierarchy palette to select and manipulate an object. This may be easier than selecting the object directly on the form design.

1) In the Hierarchy palette, click the object you want to manipulate. The object will be selected in the Hierarchy palette and on the form design.
2) Click the Design View tab.
3) Use the keyboard to manipulate the object on the form design. For example, to nudge an object 2 points, press the arrow key twice for the direction you want to move the object. The object on the form design will move in the required direction.

10.5. Aligning and sizing objects

To align objects to each other

When you align objects to each other, the last object you select determines the point of alignment. For example, if you Shift+click several objects to left align them, the objects are aligned to the left edge of the object that you selected last. In a group of selected objects, the last object selected has solid resizing handles.

RELATED LINKS:

To align objects to a grid

To center objects on the page

To distribute objects

Keys for manipulating objects

To align the edges of objects

1) Select the objects that you want to align.
2) Do one of the following actions:
   – To align the left edges of the objects, select Layout > Align > Left or click Align Left .
   – To align the right edges of the objects, select Layout > Align > Right or click Align Right .
   – To align the tops of objects, select Layout > Align > Top or click Align Top .
   – To align the bottoms of objects, select Layout > Align > Bottom or click Align Bottom .

To align the vertical centers of objects

1) Select the objects that you want to align.
2) Select Layout > Align > Vertical Center or click Align Vertical Center .

To align the horizontal centers of objects

1) Select the objects that you want to align.
2) Select Layout > Align > Horizontal Center or click Align Horizontal Center .

To align objects in subforms that flow content

You can align objects, except table rows and table cells, in subforms that flow content.

1) Select the objects that you want to align.
2) In the Layout palette, select one of the alignment buttons under Content Alignment in a Flowed Container.

To align objects to a grid

You can align objects to the closest grid increment. The grid settings can be adjusted in the Drawing Aids palette.

1) Select the objects that you want to align.
2) Select Layout > Align > To Grid.

RELATED LINKS:

Using Snap to Grid

Keys for manipulating objects

Drawing Aids palette

To center objects on the page

You can position an object in the horizontal or vertical center of the page. The center is based on the defined page size.

1) Select the objects that you want to center.
2) Select Layout > Center In Page > Horizontally, or Layout > Center In Page > Vertically.

RELATED LINKS:

To align objects to a grid

To align objects to each other

To distribute objects

To rotate objects

You can rotate an object around its anchor point in a 90°, 180°, or 270° increment. The anchor point is defined by the X and Y coordinates of an object and provides a starting place for rotating the object.

1) Select the object.
2) In the Layout palette, select the position of the anchor point relative to the perimeter of the object from the Anchor list.
3) Click one of the rotation buttons .

RELATED LINKS:

Layout properties in the Layout palette

To distribute objects

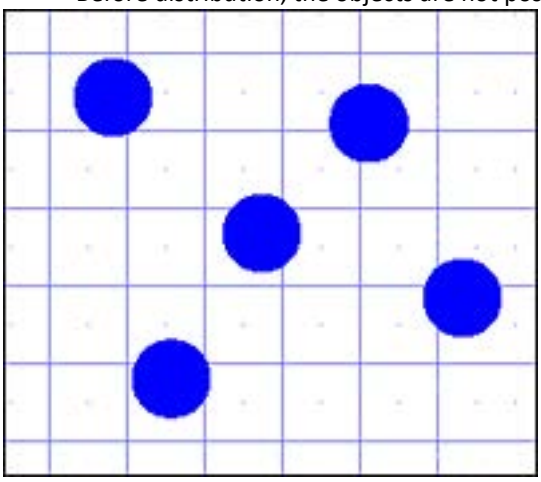
Distributing objects spaces them evenly within the area bounded by the current selection. The selected objects can be spaced evenly across or down, or arranged in rows and columns. When you distribute objects in the current selection area, the objects are spaced evenly between the leftmost and rightmost edges of the objects in the selection area.

Before distribution, the amount of space between each object differs.

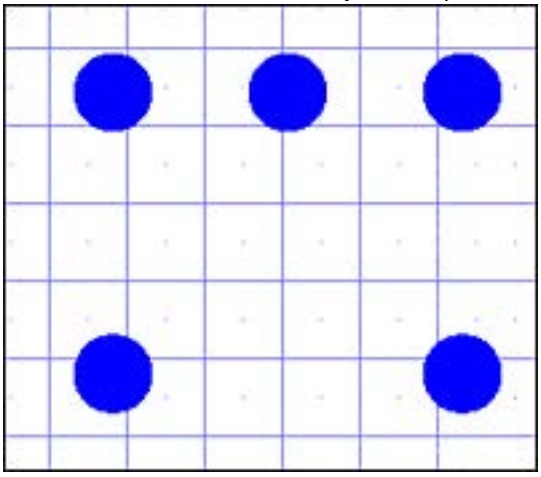
After distribution, the amount of space between each object is the same.

When you distribute a number of objects in rows and columns, the objects closest to the outer edges of the selection area determine the positions of the leftmost and rightmost columns and the top and bottom rows.

Before distribution, the objects are not positioned in columns and rows



After distribution, the objects are positionedin columns and rows



TIP: For complex layouts, distributing a single row or column at a time provides more control than distributing all of the objects in rows and columns at once. For the best results, you could try distributing the leftmost column first, followed by the topmost row. Afterward, select and left align each row with the leftmost column.

1) Select the objects that you want to distribute.
2) Do one of the following actions:
   - To distribute the objects evenly across the current selection area, select Layout > Distribute > Across or click Distribute Evenly Across .
   - To distribute the objects evenly down the current selection area, select Layout > Distribute > Down or click Distribute Evenly Down .
   - To distribute the objects in rows and columns, select Layout > Distribute > In Rows & Columns or click Distribute Evenly In Rows And Columns .

RELATED LINKS:

To align objects to each other

To align objects to a grid

To center objects on the page

Keys for manipulating objects

To resize objects

You can change the size of an object or a group of objects by using the mouse. When you resize more than one object at a time, the objects are resized proportionally. To change the size of several objects by different amounts, you select and change the size of each object individually.

You can also change the size of an object by editing the settings in the Layout palette.

NOTE: You can only resize an image field if its Sizing option (located in the Field tab of the Object palette) is set to an option other than Use Image Size.

1) Select each object that you want to resize.
2) Position the pointer over one of the selection handles, and when the pointer becomes a double-headed arrow, do one of the following steps:
   – To make the object larger, drag the handle away from the selection.
   – To make the object smaller, drag the handle toward the middle of the selection.

   As you drag the handle, the status bar displays the size of the object.

RELATED LINKS:

To make objects the same size

Layout properties in the Layout palette

To make objects the same size

You can make two or more objects the same width, height, or both. The dimensions of the last object you select determine which width or height are used. For example, if you Shift+click several objects to make the same width, the objects are resized to match the width of the object that you selected last. In a group of selected objects, the last object has solid resizing handles.

NOTE: You can only resize an image field if its Sizing option (located in the Field tab of the Object palette) is set to an option other than Use Image Size.

1) Select the objects that you want to make the same size.
2) Select Layout > Make Same Size > *[dimension]*.

RELATED LINKS:

To align objects to each other

To resize objects

To make objects expand to fit

When data is merged, objects can grow in both width and height from the anchor point. Because any expansion of the object occurs in the opposite direction from the anchor point, the anchor point may restrict the direction in which an object may grow. For example, if you choose an anchor point of Top Middle, the object may grow to the left, right, and down.

IMPORTANT: The Expand To Fit option should not be selected for Acrobat 6-compatible forms. Users cannot edit the data in expandable fields in Acrobat 6 compatible forms. Instead, you should position and size fields to accommodate the longest possible user-entered data value.

To manually expand objects, you can also select the Show Text Overflow Indicators option on the Wizards and Tips panel in the Options dialog box. See *Wizards and Tips (Options dialog box)*.

1) Select the object.
2) In the Layout palette, select the Expand To Fit option for the width, height, or both.

RELATED LINKS:

Layout properties in the Layout palette

To position objects

You can specify the precise position of an object using the X and Y coordinates in the Layout palette. The X coordinate defines the horizontal position of the object relative to the left edge of the subform. The Y coordinate defines the vertical position of the object relative to the top edge of the subform. The overall size of an object includes the caption (if any) and the associated fillable area.

1) Select the object.
2) From the Layout palette menu, select either of these options:
   – Absolute Coordinates
   – Relative Coordinates
3) In the Layout palette, specify the X and Y coordinates of the object.

RELATED LINKS:

Using Snap to Grid

Layout properties in the Layout palette

Using Snap to Grid

You can use the Snap to Grid option to automatically snap objects that you place on the form design to the nearest point on the grid. The Snap to Grid option is available in the Drawing Aids palette and from the View menu.

When you move an object, Snap to Grid ensures that the X and Y coordinates of the object's anchor point are evenly divisible by the grid unit. If an object's width or height is not evenly divisible by the current grid unit, you cannot position the object beyond that last grid point where it will not exceed the extent of the page content area.

If you want to position an object tightly against the right or bottom edge of a page, keep the following points in mind:

- If Snap to Grid is selected, the only way to position an object tightly against the right or bottom edge of the page is to ensure that its width and height are evenly divisible by the grid unit. The simplest way to achieve this is to resize the object by dragging the bottom-left corner while Snap to Grid is selected. Snap to Grid will restrict the size of the object to even multiples of the grid unit.

- If Snap to Grid is not selected, you can manually position the object as close as possible to the right and bottom edge of the page.

RELATED LINKS:

To align objects to a grid

Drawing Aids palette

Using to Snap to Guideline

Use the Snap to Guideline option to automatically place objects on guidelines you define. The Snap to Guideline option is available in the Drawing Aids palette and from the View menu. Use the Drawing Aids palette to add or delete guidelines.

When Snap to Guideline is enabled, an object that you move to within one grid space of a guideline snaps to the guideline automatically. If additional snapping options are enabled, the object must be closer to the guideline before it snaps to the guideline.

When you select multiple objects, the objects are treated as one object for snapping.

RELATED LINKS:

Drawing Aids palette

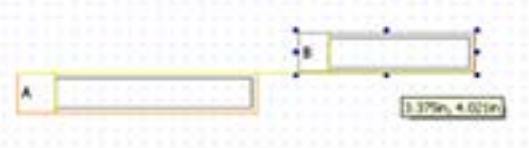
To use the drawing aids

Using Snap to Object

Use the Snap to Object option to automatically position the edge of an object using the edge of another object. The Snap to Object option is available in the Drawing Aids palette and from the View menu.

When Snap to Object is enabled, an object that you move to within one grid space of another object snaps to the other object automatically. If additional snapping options are enabled, the objects must be closer to each other before the object you are moving snaps.

Snap to Object snaps objects to other objects even if the objects do not touch. For example, in the following illustration, the bottom edge of object B snaps to the top edge of object A.



When the Snap to Object option is enabled, you can disable it temporarily by pressing the Alt key when you position the object. Pressing the Alt key temporarily enables the option when it is disabled.

NOTE: Because the Alt key disables the arrow keys, you cannot use it to toggle Snap to Object on or off when you use arrow keys to move objects.

When you select multiple objects, the objects are treated as one object for snapping.

RELATED LINKS:

Drawing Aids palette

To use the drawing aids

Using Snap to the Center of the Page

Use Snap to the Center of the Page to automatically place the center of an object at the vertical or horizontal center of the page. The Snap to the Center of the Page option is available in the Drawing Aids palette and from the View menu.

The center lines are not objects. They are displayed only when you are positioning objects near them and Snap to the Center of the Page is enabled.

When Snap to the Center of the Page is enabled, an object that you move to within one grid space of a page center line snaps to the line automatically. If additional snapping options are enabled, the object must be closer to the center line before it snaps to the line. An object can snap to the vertical and horizontal center lines at the same time, which places it at the exact center of the page.

When you select multiple objects, the objects are treated as one object for snapping.

RELATED LINKS:

Drawing Aids palette

To use the drawing aids

10.6. Formatting

Formatting text

You can define font properties for text objects, object captions, and data values.

The Font palette contains all of the font properties available. You can define these font properties in the Font palette:

- Type of font
- Font size and style
- Baseline shift
- Vertical and horizontal scale of font
- Letter spacing
- Auto kern

You can use the Font toolbar to quickly define common text formatting properties, including font type, size, and style.

NOTE: If you are creating objects that may display text in languages that use non-Latin-1 characters, make sure that you choose a font for the object that supports the full character set of the language and the appropriate locale by using the Field tab of the Object palette.

RELATED LINKS:

Font properties in the Font palette

Font palette

Formatting paragraphs

Paragraph properties in the Paragraph palette

To apply a different font

1) Select the text you want to change or select the object.
   If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   – To edit the caption and data value, in the Font palette, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, in the Font palette, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, in the Font palette, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, from the Font list, select a font name.

To change the font size

1) Select the text you want to change or select the object.
   If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   – To edit the caption and data value, in the Font palette, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, in the Font palette, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, in the Font palette, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, from the Font Size box, type or select a point size.

To apply bold, italic, underline, or strikethrough formatting

1) Select the text you want to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   – To edit the caption and data value, in the Font palette, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, in the Font palette, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, in the Font palette, click the Currently Editing palette menu and select Edit Value.

3) In the Font palette, do one of the following actions:
   – To apply bold formatting, click Bold.
   – To apply italic formatting, click Italic.
   – To apply underline formatting, click Underline.
   – To apply strikethrough formatting, click Strikethrough.

To change the color of text

1) Select the text you want to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   – To edit the caption and data value, in the Font palette, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, in the Font palette, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, in the Font palette, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, click the color selector and select the color you want.

To change the baseline shift of text

Use the baseline shift option to specify the amount of space to move a character (or group of characters) up or down relative to the baseline. Positive numbers shift characters up, and negative numbers shift characters down.

You can use baseline shift to make small adjustments to text such as positioning copyright and trademark symbols (© and ™), or to create superscript or subscript such as a number or figure used in a formula or mathematical expression, which is smaller than the adjacent text, and is set slightly above or below the line of type.

Adjusting the baseline shift does not change the line spacing or font size of the characters. When you change the spacing between lines or the font size, the baseline-shift position is maintained proportionally. You can change font sizes in the Font palette, and adjust line spacing in the Paragraph palette.

1) Select the text to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   – To edit the caption and data value, in the Font palette, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, in the Font palette, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, in the Font palette, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, in the Baseline Shift box, type the amount of shift to apply.

To change the letter spacing of text

Use the letter spacing options to adjust the amount of space between the letters in a word or group of words. Designer uses EM units of measure (proportional width of a letter *m*) to space letters. The minimum value you can enter is -1 000 (-1em). The maximum value you can enter is 10 000 (10em). The default value is 0 (no spacing). You must enter whole numbers; fractions are rounded off to the nearest whole number.

NOTE: The Letter Spacing box displays Mixed when you apply different letter-spacing values within one or more selected objects.

1) Select the text to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) In the Font palette, do one of the following actions:
   - To edit the caption and data value, click the Currently Editing palette menu and select Edit Caption and Value.
   - To edit only the caption, click the Currently Editing palette menu and select Edit Caption.
   - To edit only the data value, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, in the Letter Spacing box, type the amount of letter spacing to apply.

To change the vertical or horizontal scale of text

Use the vertical and horizontal scale options to increase or decrease the size of text. Designer calculates font scaling in percentage units. The default value for the vertical and horizontal scale is either 100% or no scaling. The minimum value you can enter is 1%. The maximum value you can enter is 1000%. Negative values are not applicable. You can enter values that have up to two decimal places (for example, 50.75%).

The percentage symbol (%) is automatically generated when the value in the box is validated. If you type an invalid value, the last correct value is displayed instead.

NOTE: The Vertical Scale and Horizontal Scale boxes display Mixed when you apply different vertical or horizontal scale values within one or more selected objects.

1) Select the text to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) In the Font palette, do one of the following actions:
   - To edit the caption and data value, click the Currently Editing palette menu and select Edit Caption and Value.
   - To edit only the caption, click the Currently Editing palette menu and select Edit Caption.
   - To edit only the data value, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, in the Vertical Scale box, or in the Horizontal Scale box, or in both, type the percentage of scaling to apply.

To auto kern text

Use the auto kern option to reduce the amount of space between certain letters. Auto kerning minimizes uneven spacing and maintains a uniform distance between the letters within a word or group of words. For example, letter combinations, such as WA, MW, and TA, are typically kerned for a better appearance.

1) Select the text to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) In the Font palette, do one of the following actions:
   – To edit the caption and data value, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, click the Currently Editing palette menu and select Edit Value.
3) In the Font palette, select Auto Kern.

Formatting paragraphs

You can define paragraph properties for text objects, object captions, and data values. To define these properties, you must first select the object.

The Paragraph palette contains all of the paragraph properties available in Designer. You can define these paragraph properties in the Paragraph palette:

- Alignment and spread
- Lists
- Indentation
- Line spacing
- Hyphenation

You can use the Text Formatting toolbar to quickly access the most common paragraph formatting commands.

RELATED LINKS:

Hyphenate text

Paragraph properties in the Paragraph palette

To align paragraphs

1) Select the text you want to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   – To edit the caption and data value, in the Paragraph palette, click the Currently Editing palette menu and select Edit Caption and Value.
   – To edit only the caption, in the Paragraph palette, click the Currently Editing palette menu and select Edit Caption.
   – To edit only the data value, in the Paragraph palette, click the Currently Editing palette menu and select Edit Value.

3) Do one of the following actions:
   – To align text left of right, in the Paragraph palette, click Align Left or Align Right.
   – To center text, in the Paragraph palette, click Align Center.
   – To justify text, in the Paragraph palette, click Justify.
   – To align text to the top of the area reserved for the caption and value, in the Paragraph palette, click Align Top.
   – To align text to the middle of the area reserved for the caption and value, in the Paragraph palette, click Align Middle.
   – To align text to the bottom of the area reserved for the caption and value, in the Paragraph palette, click Align Bottom.

To create a bulleted list

1) Select the text that you want to add bullets to, and then click Bulleted List.
2) To select a bullet style, click the arrow next to Bullets.
3) Do one of the following actions:
   – To increase the list indentation, click Increase Indent.
   – To decrease the list indentation, click Decrease Indent.

To create a numbered list

1) Select the text that you want to add numbering to, and then click Numbered List.
2) To select a number style, click the arrow next to Numbered List.
3) Specify the starting number in the Start box.
4) Do one of the following actions:
   – To increase the list indentation, click Increase Indent.
   – To decrease the list indentation, click Decrease Indent.

To add compound tags to a numbered list

1) Click anywhere within the numbered list.
2) Select the Compound Tags checkbox.
3) Select the items in the numbered list that you want to apply the compound tags to, and then click Increase Indent.

For example, in the following list:

1. Item A.

2. Item B.

3. Item C

4. Item D

increasing the indent of Item B and Item C, results in the following numbered list:

1. Item A.

    1.1. Item B.

    1.2. Item C

2. IItem D

To change the indentation of paragraphs

1) Select the text you want to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   - To edit the caption and data value, in the Paragraph palette, click the Currently Editing palette menu and select Edit Caption and Value.
   - To edit only the caption, in the Paragraph palette, click the Currently Editing palette menu and select Edit Caption.
   - To edit only the data value, in the Paragraph palette, click the Currently Editing palette menu and select Edit Value.
3) Do one of the following actions:
   - To increase or decrease the left indent of text, in the Paragraph palette, under Indents, enter the indent you want in the Left box.
   - To change the right indent of text, in the Paragraph palette, under Indents, enter the indent you want in the Right box.
   - To create a first-line indent, in the Paragraph palette, in the First list, select First Line. In the By box, type the measurement to indent.
   - To create a hanging indent, in the Paragraph palette, in the First list, select Hanging. In the By box, type the measurement to indent.

To change the spacing of paragraphs

1) Select the text you want to change or select the object. If Edit > Lock Text is enabled, double-click the text.
2) Do one of the following actions:
   - To edit the caption and data value, in the Paragraph palette, click the Currently Editing palette menu and select Edit Caption and Value.
   - To edit only the caption, in the Paragraph palette, click the Currently Editing palette menu and select Edit Caption.
   - To edit only the data value, in the Paragraph palette, click the Currently Editing palette menu and select Edit Value.
3) Do one of the following actions:
   - To change the spacing before or after text, in the Paragraph palette, under Spacing, enter the spacing you want in the Above or Below box.

– To space lines, in the Paragraph palette, from the Line Spacing list, select an option.

Hyphenation in selected paragraphs

Although the hyphenation values you select in the Form Properties and Options dialog boxes apply to the text in the entire form, you can use the Hyphenate option in the Paragraph palette to add or remove hyphenation in individual paragraphs. Use the Hyphenate option to manually adjust text layout on an object-by-object basis.

You can hyphenate the text in the caption area of objects such as text fields, decimal fields, and numeric fields, and the value area of text field objects (default text and text the form filler enters).

It is recommended that you become familiar with the various options you need to select for objects that contain hyphenated text. (See *Hyphenate text*.)

TIP: When the Hyphenate New Items is selected in the Options dialog box on the Formatting page, the Hyphenation option is automatically selected with new objects added to a form.

RELATED LINKS:

Hyphenate text

Considerations for setting hyphenation

Setting and removing hyphenation in forms

Paragraph properties in the Paragraph palette

Formatting (Options dialog box)

Formatting (Form Properties dialog box)

To hyphenate the text in a selected paragraph

1) To enable) the Hyphenate option in the Paragraph palette, do one of the following actions:
   – (New forms) Select Tools > Options > Formatting, and then choose Allow Hyphenation in Text and Field Captions, or Allow Hyphenation in Text Field Values, or both.
   – (Current form) Select File > Form Properties > Formatting > and choose Allow Hyphenation in Text and Field Captions, or Allow Hyphenation in Text Field Values, or both.
2) Select the text to hyphenate.
3) (Does not apply to text objects) In the Paragraph palette, click the Currently Editing palette menu and do one of the following actions:
   – To hyphenate caption and data value text, select Edit Caption and Value.
   – To hyphenate only caption text, select Edit Caption.
   – To hyphenate only data value text, select Edit Value.
4) To hyphenate the text in the value area of text field objects, in the Object palette, on the Field tab, select Allow Multiple Lines.
5) In the Layout palette, select Expand to Fit (Width), or Expand to Fit (Height), or both.
6) In the Paragraph palette, select Hyphenate. The change is applied.

To remove hyphenation from a selected paragraph

1) Click the text to remove hyphenation from and, in the Paragraph palette, deselect Hyphenate. The change is applied.

Formatting objects

RELATED LINKS:

Font properties in the Font palette

Paragraph properties in the Paragraph palette

Layout properties in the Layout palette

Border properties in the Border palette

To add space around an object

All objects have a certain amount of white space or margin around them.

1) Select the object.
2) In the Layout palette, set the margins for Left, Right, Top, and Bottom.

To add a border around an object

To define border properties for an object that supports borders, you must first select the object. The properties that you define in the Border palette apply to the whole object.

1) Select the object.
2) In the Border palette, from the Edges list, select an editing option.
   – To apply a border to each edge separately, select Edit Individually. This option does not apply to three-dimensional borders.
   – To apply the same border to all edges, select Edit Together.
3) Beneath the Edges list, select a style, width, and color for the borders.
4) To have notched corners, click one of the Corners buttons and, in the Radius box, define the corner radius (in inches, centimeters, or points).
5) To fill the object with color or a pattern, select an option from the Style list and, using the adjacent color selector buttons, select appropriate fill colors.

To add a border around the fillable area

You can also specify the border properties of the fillable areas of an object. For example, for a text field, you can apply one type of border and fill to the overall object and another type of border and fill to the value area that displays data. These objects have fillable areas:

- Command buttons
- Check boxes
- Signature Fields

- Drop-down lists
- List boxes
- Numeric fields
- Password fields
- Radio buttons
- Text fields

To add a border around the fillable area:

1) Select the object.
2) In the Object palette, click the Field tab. From the Appearance list, select Custom.
3) Select a style for the border of the fillable area. The options are the same as those in the Border palette.
4) If applicable for the object, type a border width and select a color for the border.
5) To have notched corners, click one of the Corners buttons and in the Radius box, define the corner radius (in inches, centimeters, or points).
6) To fill the fillable area with color or a pattern, select an option from the Style list and using the adjacent color selector buttons, select appropriate fill colors.

To change the default formatting for new objects

You can set the default formatting for new objects and save it in the Object Library. For example, you can change the default font for all new Text objects to Times New Roman.

1) Insert an object.
2) Make the changes to the object. For example, add a border, apply a different font, change the size of the text.
3) Drag the object into a category in the Object Library palette.
4) In the Add Library Object dialog box, type the name of the object and click OK.

   If another object of the same name already exists in that tab, Designer asks you to confirm that you want to replace the existing object.

Using leaders in text

Use leaders to guide the reader from one piece of information to another across a page, such as in a table of contents, price list, or invoice. You can add leaders to a line or block of text in text objects and in the caption area of objects such as text fields, decimal fields, and numeric fields. When the text object or caption area is in edit mode, the Leader command is available in the Insert menu and the Context menu.

You can define and edit leader properties such as the end position, pattern, line thickness or dot spacing, and alignment. You can also adjust the positioning, size, and appearance of leaders by using the options in the Font and Paragraph palettes.

When you add a leader, Designer draws a leader from the insertion point to the end position that you specify. The leader end position is based on ruler increments. The default end position extends the leader

to the right edge of the caption or static text object. You can only define or edit one leader at a time. All changes are applied immediately.

You can define different leaders for each paragraph of text. However, it is important to understand that a new paragraph inherits the leader properties from the previous paragraph. In other words, when you create a new paragraph, any leaders defined in the previous paragraph apply to the new paragraph.

NOTE: Designer can not anticipate the content of run-time data. If you intend to populate a form with rich text that contains leaders, you must set the target version to Acrobat 9 when you design the form.

RELATED LINKS:

Delete Tab Stops dialog box

Font properties in the Font palette

Font palette

To add a leader

1) Place the insertion point where you want the leader to start within the text object or caption area, and then select Insert > Leader.
2) In the Leader End Position box, type the ruler position where you want the leader to end.
3) In the Leader Pattern list, select the pattern to fill the leader.
4) In the Leader Dot Spacing or Leader Underline Weight list, select the amount of space to appear between dots or dashes, or select the thickness of the line.
5) Set the alignment for the leader by doing one of these actions:
   – To align the left side of the text that follows the leader (immediately before the first letter) with the specified end position, click Left Align Leader.
   – To align the right side of the text that follows the leader (immediately after the last letter) as the end position for the leader, click Right Align Leader.
6) Click anywhere in the Layout Editor to close the dialog box.

To edit a leader

1) Select the leader and edit it.
2) Click anywhere in the Layout Editor to close the dialog box.

To delete a leader

The Delete Leader command removes the selected leader (tab), the associated tab stops, as well as any leader properties, such as pattern, spacing, and alignment.

NOTE: The Delete Leader command is available on the Edit menu after you select a leader. However, if the Delete Leader command remains unavailable after you select a leader, this means that the leader is not associated with a tab stop. When this is the case, you can use the Delete key to remove the leader.

1) Select the leader to delete.
2) Select Edit > Delete Leader.

To enable a visual clue for No Leader

When you select the No Leader option while inserting Leaders, the leaders are not displayed in both, the Preview PDF and the final PDF form. You can turn on a visual indicator to display tab stops when No Leader is selected. As this value is read while opening the document, close and reopen already open documents for this value to take effect.

1) In Tools > Options > Formatting, check Show non blank tab stop in design view option.
2) Specify a character to see in place of blanks.

To delete tab stops

If you use the Delete key on the keyboard to delete a selected leader, only the tab character is removed. Any associated tab stops or leader properties such as pattern, spacing, and alignment are not removed.

To delete unused tab stops, you must use the Delete Tab Stops dialog box.

NOTE: The Delete Tab Stops command is available on the Edit menu when there are one or more tab stops in the current paragraph. If there are two or more paragraphs within a selection, the dialog is unavailable.

1) Select Edit > Delete Tab Stops.
2) Complete one of the following actions:
   – To delete selected tab stops, select one or more tab stops from the list, and then click Clear.
   – To delete all tab stops, click Clear All.
3) Click OK.

10.7. Creating an insertion point

Use insertion points in forms that are assembled on a server by using the Assembler services.

An *insertion point* is a subform that acts as a placeholder for a fragment that is inserted into the form when the form is assembled. You can add one or more insertion points to the body or the master pages of a form.

Use one of these methods to create an insertion point:

- Insert an insertion point object into a form:
  – Inserts a subform with the default name *Insertion Point*.
  – Inserts an insertion point subform that contains an insertion point placeholder (a text object that contains temporary placeholder content).
  – Sets the Data Binding (Open, Save, Submit) option on the Binding tab to No Data Binding.
- Insert an insertion point into an existing subform:
  – Inserts an insertion point subform within the selected subform. The default name is (*untitled Subform*)
  – Inserts an insertion point placeholder into the insertion point subform.

  - For the existing subform, sets the Data Binding (Open, Save, Submit) option on the Binding tab to Use Name (*<object name>*). For the insertion point subform, sets the Data Binding (Open, Save, Submit) option on the Binding tab to No Data Binding.
  - Selects the Expand To Fit options on the Layout tab for the insertion point placeholder (text object). Therefore, the insertion point placeholder can expand according to the amount of text contained in the insertion point placeholder.

- Define an existing subform as an insertion point:
  - Makes the select subform an insertion point.
  - Maintains all existing option settings for the subform. For example, if the Data Binding (Open, Save, Submit) option on the Binding tab is set to Use Name (<object name>) before you define the subform as an insertion point, this option setting is retained after the change.
- Does not insert an insertion point placeholder. You can manually add a text object as an insertion point placeholder.

You can insert one or more insertion points into an insertion point. By default, each new (child) insertion point occupies the same area as the parent insertion point. As such, multiple insertion points can overlap. Therefore, it is recommended that you use the Hierarchy palette to select the individual objects that make up an insertion point. The *Hierarchy palette* is a graphical representation of the contents in the Design View and Master Pages tabs. What you select in the Hierarchy palette is also selected in the body or master page that it is associated with.

In the Draw Aids palette, you can select various border styles for an insertion point on the Drawing Aids palette.

You can optionally name an insertion point subform in the Name box on the Subform tab. To relate the insertion point subform to the associated fragment when the form is assembled, use the same name for the insertion point and the fragment. It is recommended that you type the name in medial capitals, a mix of uppercase and lowercase (for example, LiveCycle or InDesign).

Insert an insertion point object into a form

1) In the Hierarch palette, select where to insert the insertion point object.
2) Click Insert > Custom > Insertion Point.
3) (Optional) In the Name box, type a name for the insertion point subform.
4) Position the insertion point within the form as required.

Insertion point into an existing subform

1) In the Hierarch palette, select the subform to insert an insertion point into.
2) Click Insert > Insertion Point.
3) (Optional) In the Name box, type a name for the insertion point subform.

Define a subform as an insertion point

1) In the Hierarchy palette, select the subform to define as an insertion point.
2) In the Object palette, click the Subform tab and select This Subform Is An Insertion Point.
3) (Optional) In the Name box, type a name for the insertion point subform.

Add an insertion point placeholder (to a subform that is defined as an insertion point)

1) In the Hierarchy palette, select a subform that is defined as an insertion point, and click Insert > Standard > Text.
2) In the Object palette, on the Draw tab, select This Is An Insertion Point Placeholder.
3) Click inside the text object and type the text to identify the subform as an insertion point. For example, type `This is temporary placeholder content`.

10.8. Formatting captions

A number of the standard field objects in the Object Library palette support captions. Captions provide textual information to the viewers about the form field.

When you add an object that supports captions, Designer assigns a default name for the caption. The default name is the name of the object. For example, if you add a list box object to the form design, the default caption is List Box.

When creating the form design, you can see the caption for the object in two places: on the form design next to the object (the default position) or in the Field tab of the Object palette.

You can edit the caption text by using the Field tab of the Object palette or by editing the caption text directly on the page. If the text wraps as you type, increasing the size of the object will display the text properly. If the object displays a white cross in a red box, the object either needs to be resized or have the caption reserve adjusted. A reserve of 0 sets the caption area to auto-fit so that it adjusts to fit the entire caption. The caption is protected and users cannot be modified it when filling the form.

NOTE: When you bind a caption to a data source, the caption is always shown in the form, even if you have set Caption Position in the Layout tab to None.

If you need to rearrange field objects, but do not want to risk modifying their captions, you can use the Lock Text command. Using this command lets you to move the field objects around easily but makes the caption not as easy to modify.

NOTE: You cannot select or edit captions if a field object is locked. See *To lock objects*.

RELATED LINKS:

Formatting

To create a caption by merging a text object and a field object

Font properties in the Font palette

Paragraph properties in the Paragraph palette

Layout properties in the Layout palette

To lock objects

To dynamically populate a caption

To set a default font for captions in new forms

1) Click Tools > Options.
2) Click Default Fonts.
3) Under Default Caption Font Properties For New Forms, select Typeface, Size, and Style options, as needed.

To set a default font for captions in an existing form

1) Click File > Form Properties.
2) Click Default Fonts.
3) Under Default Caption Font Properties, select Typeface, Size, and Style options. as needed.

To rename a caption

1) Select the object.
2) Do one of the following actions:
   – To change the caption in the Layout Editor, select the text of the caption and type a new caption. If Edit > Lock Text is enabled, double-click the text.
   – To change the caption by using the Object palette, click the Field tab type a new caption in the Caption box.

   TIP: You can dynamically populate a caption with a value from a data source.

To position a caption

Captions are positioned relative to the fillable area of an object.

1) Select one or more objects.
2) In the Layout palette, select a position for the caption from the Position list.

   NOTE: When you bind a text field caption to a data source, the text field's caption label is always shown in the form, even if you selected None for the caption position in the Layout tab

To specify the amount of space for a caption

You can adjust the amount of white space between the caption and the fillable area.

1) Select one or more objects.
2) In the Layout palette, type a new value in the Reserve box.

   NOTE: If you enter a reserve space before selecting a position for the caption, Designer adjusts the reserve space automatically.

To set the caption area to auto-fit

1) Select one or more objects.
2) In the Layout palette, type 0 in the Reserve box. This number sets the caption area to auto-fit so that it adjusts to fit the entire caption.

To hide a caption

1) Select one or more objects.
2) In the Layout palette, select None from the Position list.

10.9. Making objects visible, invisible, or hidden

Depending on the purpose of the form, you may want to control which objects are displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form from within Acrobat or Adobe Reader. With the various options in the Presence list in the Object palette, you can selectively make different objects in a PDF form visible, invisible, or hidden when the form is viewed on-screen or printed from within Acrobat or Adobe Reader. You can also specify that an object only print when the form is printed on the front or both sides of the printed page.

Keep in mind the following factors when applying presence options to objects:

- The Invisible and Hidden options are unavailable for groups, content areas, master pages, page sets, and subform sets.
- The presence option you apply to container objects such as subforms overrides the presence option applied to each contained object. For example, if you apply the Visible (Print Only) option to a subform set, the subform and the collected objects will not appear on-screen but will appear in the printed form, regardless of the presence setting of the individual objects.
- If you want to print a form in Acrobat or Adobe Reader, do not apply the Visible (Screen Only) option to any content area objects that contain objects you want to display in the printed form. When you apply the Visible (Screen Only) option to a content area object, the objects within the content area are only displayed onscreen and do not appear in the printed form. A warning marker appears in the upper-right corner of the master page when a content area or master page object is not present in both the screen and the print views. If you want to display one set of objects when a form is printed and another when the form is viewed onscreen, you can create one master page with two content area objects, one for each set of objects. You apply the Visible (Print Only) option to one content area object, and the Visible (Print Only) option to the other.
- When you apply the Visible (Print Only) option to an object, the object will appear in the Design View and Master Page tabs but will not appear in the Preview PDF tab.

The following list describes some of the ways you can use the presence options:

Visible

The object is visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or the server), and occupies space in the form layout. For example, if you want the

form to look the same on-screen and in print, you can apply the Visible option to all of the objects in the form. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form if printed from within Acrobat or Adobe Reader, and occupies space in the form layout. The object is visible in the printed form if printed from the server. For example, if you have a form that users will fill, submit online, and then print, you can apply the Visible (Screen Only) option to Print and Submit buttons so that the buttons are visible on-screen but not visible in the printed form.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form if printed from within Acrobat or Adobe Reader, and occupies space in the form layout. The object is visible in the printed form if printed from the server. For example, if you need to include some instructional text in the printed form that is not required on-screen, such as mailing instructions, you can apply the Visible (Print Only) option to Text Field objects that contain the instructions so that the text is visible in the printed form but not visible on-screen.

Inactive

Inactive presence indicates that an object is hidden and excluded from event processing. Calculations, validations, and other events do not trigger for inactive objects. The enumeration of the presence attribute determines which of the form processing stages a form object participates in. Inactive objects participate in the step of merging data with template to create Form DOM. ***NOTE:*** *Presence Inactive applies to all Designer objects.*

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout. For example, if you want an area in a form to remain invisible until a user selects a certain option, such as the section for payment-type information in a billing form, you can apply the Invisible option to the text field objects used for gathering credit card information so that they remain invisible until the user selects a Radio button that indicates the credit card type.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout. For example, if you want a section in a form to remain hidden until a user selects a certain option and to not occupy space in the form layout when the section is hidden, you can apply the Hidden (Exclude from Layout] option to the subform that contains the various objects in the section so that the form layout adjusts as needed.

NOTE: The Hidden (Exclude from Layout) option works as described only when you apply this option to objects that are placed within a subform that is set to Flowed.

One-sided Printing Only

The object is visible on-screen, visible on the front side of each printed page of the form (when printed from within Acrobat, Adobe Reader, or the server), and occupies space in the form layout.

For example, to have the page number appear at the lower-right of the pages, apply the One-sided Printing Only option to the page number object.

Two-sided Printing Only

The object is visible on-screen, visible on the both sides of each printed page of the form (when printed from within Acrobat, Adobe Reader, or the server), and occupies space in the form layout. For example, to have the page number appear at the lower-right of each odd-numbered printed page and at the lower-left of each even numbered printed page when double-sided printing, you must create two master pages. In the first master page, you place the page number object at the lower-right and apply the Visible option. In the second master page, you create two page number fields. Place one at the lower-left and apply the Two-sided Printing Only option; place the other at the lower-right and apply the One-sided Printing Only option.

To make an object visible, invisible, or hidden

The Presence list appears in a different tab, depending on the object selected.

1) Select the object.
2) In the Object palette, click the appropriate tab.
3) In the Presence list, select one of the following options:
   - To make the object visible on-screen, visible in the printed form, and occupy space in the form layout, select Visible.
   - To make the object visible on-screen, not visible in the printed form, and occupy space in the form layout, select Visible (Screen Only).
   - To make the object not visible on-screen, visible in the printed form, and occupy space in the form layout, select Visible (Print Only).
   - To make the object not visible on-screen, not visible in the printed form, and occupy space in the form layout, select Invisible.
   - To make the object not visible on-screen, not visible in the printed form, and not occupy any space in the form layout, select Hidden (Exclude from Layout).
   - To make the object visible on-screen, visible on only the front side of each sheet in the printed form, and occupy space in the form layout, select One-sided Printing Only.
   - To make the object visible on-screen, visible on both sides of each sheet in the printed form, and occupy space in the form layout, select Two-sided Printing Only.

10.10. Formatting field values and using patterns

Depending on the requirements of your situation, you can specify one or more of the following patterns to control how field values, such as text fields, numeric fields, and date/time fields are formatted at run time:

- A display pattern, which describes how data will be displayed in the form. If you define an initial default value, it is formatted according to the display pattern. The display pattern is also responsible for formatting user input and any bound values retrieved at run time.

- An edit pattern, which describes the syntax for entering data into a date/time field, numeric field, text field, or password field at run time.
- A validation pattern, which is used to validate user input at run time.
- A data pattern, which describes the syntax of bound or saved data.

The formatting options that you choose will depend on the purpose of your form. For example, if you are designing an interactive form, for each field you should define an edit pattern to process user input and a validation pattern to validate the input. You would only define a data pattern if the fields are bound to a data source.

Keep in mind that if you specify only an Edit pattern for a Numeric Field or Decimal fields object, form fillers can still enter alphabetic characters in the field. To avoid this behavior, do one of the following actions:

- Do not specify just an Edit pattern. Ensures that Acrobat and Adobe Reader filter out unwanted alphabetic characters.
- Specify Edit and Display patterns. Ensures that the data is formatted correctly according to the Display pattern.
- Specify Edit and Validation patterns. Ensures that the value is rejected and the field is cleared when a form filler enters an alphabetic character.

When to use patterns

Use patterns to control how field values are processed at run time. For example, users can enter letters and numbers into a text field and any special punctuation or spacing can be applied automatically according to a predefined pattern before the value is displayed.

Capturing and displaying user input

If you are creating a form to capture data, you can specify how data should be formatted. You specify how the data should appear using a display pattern. If you do not specify a display pattern, the data appears according to Designer defaults.

If users will be entering data that does not match the Designer defaults, you must specify an edit pattern. The edit pattern describes the syntax of the user input. Given the pattern, the run-time application converts the user input into a raw value and then formats the value according to the display pattern.

If you are designing an interactive form, consider what user input must be validated. For example, a text field may or may not require validation depending on usage. A multiple line text field allowing the form filler to enter a comment does not need to be validated. Similarly, a numeric field will automatically prevent the form-filler from entering any non-numeric data. However, if the data has to be restricted to a specific range of numbers, you will want to validate the user input. You can choose to display a custom message to prompt users for a correct value at run time. If you do not specify a custom message, the system generates one automatically.

Remember that by using the options on the Form Validations tab in the Form Properties dialog box, you can configure how Acrobat displays validations messages, highlights failed or mandatory fields that contain invalid data or no data, and sets the focus on the first field that fails to validate. See *Displaying validation errors in Adobe Acrobat*.

NOTE: User input can be processed through FormCalc formulas and JavaScript scripts (for example, a script can request the raw value of a field). Because formulas and scripts operate on raw and formatted values, it is important to validate those fields where input is restricted.

One example of how an edit and validation pattern may be used together is a credit card or social security number entry. You could define a text field with the following edit patterns:

text{9999-9999-9999-9999}|text{9999 9999 9999 9999} for credit cards

or

text{999-99-9999}|text{999 99 9999} for a US social security number

In both cases, the user may enter the number with hyphen(-), space ( ), or just the 16 or 9 digit number. The canonical, or simplest form of the number is the 16 or 9 digit number.

You may also choose to add the following validation pattern:

text{9999999999999999}

or

text{999999999}

In this case, only the number is stored and the validation checks for the correct number of digits. However, in this case, it might be more useful to specify a validation script rather than a pattern. There are algorithms that will checksum a credit card number to ensure that it looks like a valid credit card number and not just a random 16 digit number. An example is the Luhn Algorithm for credit cards.

The result is a form that has a text field where the edit pattern allows user entry in one of three typical ways for typing a credit number, and the validation runs a script that validates that the number looks like a valid credit card number.

Retrieving and displaying bound data

If bound data will be merged with a form, you can specify how the data should be formatted for display using a display pattern. If you do not specify a display pattern, the data is displayed according to Designer defaults.

If the bound data does not match Designer defaults, you must specify a data pattern. The data pattern describes the syntax of the bound data. Given the pattern, the run-time application converts the retrieved data into raw values and then formats them for display.

Defaults for value formatting

Default values must conform to the following rules, depending on the type of field.

Field	Rule
Date/Time Field	A default date/time value must conform to the short format for the locale specified for the date/time field. However, by default, Designer displays the default value in the medium locale format at both design time and run time. For example, consider a form with a Date/Time Field set to use the German (Germany) locale. You enter the default value for a date in the short format DD.MM.YY. After you change the focus to another field, the value specified in the field on the page is displayed in the medium format DD.MM.YYYY. The formatted value also appears in the medium format if you view the form in the Preview PDF tab. ***NOTE:*** *At run time, by default, form fillers must edit the value of date/time fields using the short format for the locale specified for the field. If you specify an Edit Pattern on the Edit tab in the Patterns dialog box (Field tab > Patterns), that pattern overrides the short format, and users must enter data that conforms to the Edit Pattern.*
Numeric Field or Decimal Field	A default numeric value can be any integer or any decimal number that contains a single radix point. The radix character can be either a "." (period) or "," (comma) depending on the locale selected. Thousands separators (or grouping symbols) and currency symbols are not valid as part of the default value. For example, if a numeric field is set to the locale English (USA), and you specify the default value $1,234.56, both the currency symbol "$" (dollar sign) and the thousands separator "," (comma) are not valid.
Text Field	A default text value (including passwords) can be any alphanumeric text string, including spaces.

NOTE: Only those fields listed in the table have default values that must conform to locale-specific formatting.

To specify a default value

Date/time fields, numeric fields, and text fields can display an initial (default) value when the form is opened. The value can be derived from a run-time property, or you can specify the value explicitly in Designer. The value can also be derived from an external data source through binding. At run time, Designer formats field default values according to the locale specified for each field.

1) Select a date/time field, decimal field, numeric field, or text field.
2) In the Object palette, click the Field tab. Select a locale from the Locale list.
3) In the Object palette, click the Value tab. Type the value into the Default box.

   The default value must be specified in locale-sensitive format.

   NOTE: If the data is bound and a data pattern has been specified, the value must match the data pattern specified in the Binding tab.

To specify a display pattern

At run time, Designer displays date, time, and numeric field values in locale-sensitive format. If you want to display a field value in a format other than the default, you can specify the custom pattern by clicking the Patterns button on the Field tab.

NOTE: Drop-down lists support custom user entries, but a display pattern for custom user entries cannot be specified. You can write a script to format the user input if required.

Because the display pattern describes how data will be displayed in the form, all default values, user-entered values, and values retrieved from a database are converted to the format described by the display pattern.

NOTE: Dates earlier than January 1, 1900 are not formatted by the display pattern.

1) Select the date/time field, numeric field, or text field.
2) In the Object palette, click the Field tab.
3) Click Patterns and either select one of the predefined display patterns from the Select Type list or type a custom pattern in the Pattern box.

To prompt users to enter data

Prompts are useful for situations where users are expected to enter data or make a selection. You can write a message to prompt users to enter a value into a date/time field, numeric field, text field, password field, or drop-down list, or prompt users to select an option from a drop-down list, list box, or radio button group.

Recommending that users enter data

You can recommend that users enter data in a field but still let them submit the form if they do not. If a user enters data in the field, leaves the field and then clears it, a message box appears. A custom message appears if one written in the Empty Message box. A standard empty field message appears if you do not type a custom message. A message only appears if there was data in the field, the value was deleted, and the user exited the field without re-entering data. If the user never attempts to enter data in the field and tries to submit the form, a field is required message appears. The user can choose to ignore the message and submit the form. Choose User Entered - Recommended to recommend that users enter data in a field.

Requiring that users enter data

You can make it mandatory for users to enter data in a field before they can submit a form. If a user enters data in the field, tabs out, and then returns to clear it, a message box appears. A custom message appears if one written in the Empty Message box. A standard empty field message appears if you do not type a custom message. A message only appears if there was data in the field, the value was deleted, and the user exited the field without re-entering data. If the user never attempts to enter data in the field and tries to submit the form, a field is required message appears. Choose User Entered - Required to make it mandatory that users enter data in a field.

Remember that by using the options on the Form Validations tab in the Form Properties dialog box, you can configure how Acrobat displays validations messages, highlights failed or mandatory fields that

contain invalid data or no data, and sets the focus on the first field that fails to validate. See *Displaying validation errors in Adobe Acrobat*.

NOTE: If users do not enter a value into the field and try to submit the form, the error message field is required appears. However, users can save and close a PDF form without providing recommended or required values. In this case, no messages appear to prompt users for input.

1) Select the field, drop-down list, list box, or radio button group.
2) In the Object palette, click the Value tab. From the Type list, select one of these options:
   – User Entered - Recommended
   – User Entered - Required
3) In the Empty Message box, type the prompt. If applicable, the prompt should specify the required input format. For example, if you defined an edit pattern, the user input must conform to the edit pattern.

To specify an edit pattern

At run time, Designer displays date, time, numeric, and decimal field values in locale-sensitive format. If you want to permit form fillers to edit field values in a format other than the locale-sensitive default, you can specify an Edit Pattern on the Field tab. If the user's input does not conform to the edit pattern, the data is input as-is.

If you specify only an Edit pattern for a Numeric or Decimal field object, form fillers can still enter alphabetic characters in the field.

The edit pattern can be different than the display pattern. For example, because it is easier for users to enter short dates and read long dates, you could consider specifying a short date for a date/time field's edit pattern and a long date for its display pattern. When the display and edit patterns are different, the value is formatted to match the display pattern as soon as the user exits the field.

NOTE: This option is not available when the Type option in the Value tab of the Object palette is set to Protected, Calculated - Read Only or Read Only.

1) Select the date/time field, numeric field, text field, or password field.
2) In the Object palette, click the Field tab.
3) Click Patterns, click the Edit tab, and either select one of the predefined display patterns from the Select Type list or type a custom pattern in the Pattern box.

To validate user input

Three separate validations are possible for any field. The order of initiation of these validations is as follows:

- Test the field for null content.
- Verify the format of the field value against a specific field pattern. For more information about field patterns, see *Simple patterns*.
- Invoke a validation script.

You can define a validation pattern to validate user input for date/time fields, numeric fields, text fields, and password fields. By default, null entries are not accepted when a value is required. Raw values are compared to the validation pattern directly and, if the raw value matches the validation pattern, it is formatted for display.

If the user-entered value does not match the validation pattern, a programming error or warning appears. The error/warning is trapped by Acrobat, Adobe Reader, or Forms, which returns a message to the user automatically. If an edit pattern has not been specified and the user input does not match Designer defaults, validation fails.

A validation message appears if objects that require values contain null values and the user attempts to submit data to Forms.

NOTE: Users can save and close a PDF form without providing required values. In this case, no validation is performed.

If needed, you can write a custom validation pattern message to replace the default error or warning message.

In addition to a validation pattern, or in cases where a validation pattern is not supported (for example, for radio button groups and check boxes), you can validate user input by using a validation script. Validating input through a script ensures that the data is acceptable for your application. A custom message and run-time error or warning is also supported in this case.

Remember that by using the options on the Form Validations tab in the Form Properties dialog box, you can configure how Acrobat displays validations messages, highlights failed or mandatory fields that contain invalid data or no data, and sets the focus on the first field that fails to validate. (See *Displaying validation errors in Adobe Acrobat*.)

TIP: You can dynamically populate a validation pattern message with a value from a data source. This option allows you to ensure that users enter a valid value in the field.

To define a validation pattern and custom message

1) Select the date/time field, numeric field, text field, password field, drop-down list, or list box.
2) In the Object palette, click the Value tab.
3) Click Validation Pattern and either select one of the predefined validation patterns from the Select Type list or type a custom pattern in the Pattern box.
4) In the Validation Pattern Message box, type a message to prompt users to enter the correct value. The message should specify the required input format. To start a new line in the message, press Ctrl+Enter.
5) To have a programming error to appear instead of a warning, select the Error option.

To display a message when an attached script detects unacceptable input

1) Select the date/time field, numeric field, text field, password field, drop-down list, list box, check box, or radio button group.
2) In the Object palette, click the Value tab. In the Validation Script Message box, type the message.
3) To have a programming error appear instead of a warning, select the Error option.

To specify a data pattern

Data binding options enable you to build a form that captures data for enterprise infrastructures and/or use an external data source to populate a form at run time. For example, given appropriate binding information (see *Binding fields to a data source*) and access to the data source (see *Working with Data Sources*), Acrobat and Adobe Reader can import and display data from an OLEDB database when the form is opened. Objects can also be bound to an XML schema, an XML file, or a WSDL data source.

Acrobat, Adobe Reader, and Forms interpret the data-binding properties to store captured data and parse retrieved data. By default, an object's data is stored and merged according to Adobe data-merging rules. When a form opens in Acrobat or Adobe Reader, or is rendered by Forms, the field values are populated from the data source. Any changes to a field's value by the user are committed to the associated data source when the form is saved in Acrobat or Adobe Reader or the data is submitted to Forms.

If the data is not bound to a data source (for example, if the form data will be returned by email), the data pattern specifies the format that the data is saved in. If you do not create a data pattern, the data will be saved in canonical format. If a form may be filled by end users in a variety of locales or if the data may be returned to more than one locale, having the data in canonical format helps ensure that it is interpreted the same way by all users.

You can specify data patterns for date/time fields, numeric fields, text fields, and password fields. If the data pattern prevents Acrobat or Adobe Reader, or Forms from parsing a retrieved value, the value appears in the form unchanged (it is not formatted for display).

1) Select the date/time field, numeric field, text field, or password field.
2) In the Object palette, click the Field tab.
3) Click Patterns, click the Data tab, and either select one of the predefined data-binding patterns from the Select Type list or type a custom pattern in the Pattern box.

Simple patterns

Simple patterns can be used to format the values of date/time fields, numeric fields, text fields, and password fields. They each have their own rules governing the valid formation of patterns. There is a limited set of characters that you can use in a pattern, and the syntax of a valid pattern differs among date/time fields, numeric fields, text fields, and password fields.

For information about the valid characters that you can use in a pattern and examples of valid constructs, see one of the sections listed below. For information about complex patterns for a date/time field, numeric field, or text field, see *Complex field patterns*.

Locales

A *locale* is a standard term used when developing international standards to identify a particular nation (language, country or region). For the purposes of FormCalc, a *locale* defines the format of dates, times, numeric, and currency values relevant to a specific nation or region so that users can use the formats they are accustomed to.

Each locale is comprised of a unique string of characters called a *locale identifier*. The composition of these strings is controlled by the international standards organization (ISO) Internet Engineering Task Force (IETF), a working group of the Internet Society (www.isoc.org).

Locale identifiers consist of a language part, a country or region part, or both. The following table lists valid locales for this release of Designer.

Language	Country or Region	ISO Code
Arabic	Algeria	ar_DZ
Arabic	Bahrain	ar_BH
Arabic	Egypt	ar_EG
Arabic	Iraq	ar_IQ
Arabic	Jordan	ar_JO
Arabic	Kuwait	ar_KW
Arabic	Lebanon	ar_LB
Arabic	Libya	ar_LY
Arabic	Morocco	ar_MA
Arabic	Oman	ar_OM
Arabic	Qatar	ar_QA
Arabic	Saudi Arabia	ar_SA
Arabic	Sudan	ar_SD
Arabic	Syria	ar_SY
Arabic	Tunisia	ar_TN
Arabic	United Arabian Emirates	ar_AE
Arabic	Yemen	ar_YE
Armenian	Armenia	hy_AM
Azerbaijani-Cyrillic	Azerbaijan	az_Cyrl_AZ
Azerbaijani-Latin	Azerbaijan	az_Latn_AZ
Basque	Spain	eu_ES
Bosnain	Bosnia and Herzegovina	bs_BA
Bulgarian	Bulgaria	bg_BG
Catalan	Spain	ca_ES

Language	Country or Region	ISO Code
Chinese	People's Republic of China (Simplified)	zh_CN
Chinese	Hong Kong S.A.R., China	zh_HK
Chinese	Taiwan (Traditional)	zh_TW
Croatian	Croatia	hr_HR
Czech	Czech Republic	cs_CZ
Danish	Denmark	da_DK
Dutch	Belgium	nl_BE
Dutch	Netherlands	nl_NL
English	Australia	en_AU
English	Belgium	en_BE
English	Canada	en_CA
English	Hong Kong S.A.R., China	en_HK
English	India	en_IN
English	India Rupee	en_IN_RUPEE
English	Ireland	en_IE
English	New Zealand	en_NZ
English	Philippines	en_PH
English	Singapore	en_SG
English	South Africa	en_ZA
English	United Kingdom	en_GB
English	United Kingdom Euro	en_GB_EURO
English	United States of America	en_US
English	U.S. Virgin Islands	en_VI
Estonian	Estonia	et_EE
Finnish	Finland	fi_FI
French	Belgium	fr_BE
French	Canada	fr_CA

Language	Country or Region	ISO Code
French	France	fr_FR
French	Luxembourg	fr_LU
French	Switzerland	fr_CH
German	Austria	de_AT
German	Germany	de_DE
German	Luxembourg	de_LU
German	Switzerland	de_CH
Greek	Greece	el_GR
Hebrew	Israel	he_IL
Hindi	India	hi_IN
Hungarian	Hungary	hu_HU
Indonesian	Indonesia	id_ID
Italian	Italy	it_IT
Italian	Switzerland	it_CH
Japanese	Japan	ja_JP
Kazakh	Kazakhstan	kk_KZ
Khmer	Cambodia	km_KH
Korean	Korea	ko_KR
Korean	Korea Hanja	ko_KR_HANI
Lao	Laos	lo_LA
Latvian	Latvia	lv_LV
Lithuanian	Lithuania	lt_LT
Malay	Malaysia	ms_MY
Norwegian - Bokmal	Norway	nb_NO
Norwegian - Nynorsk	Norway	nn_NO
Persian	Iran	fa_IR
Polish	Poland	pl_PL
Portuguese	Brazil	pt_BR

Language	Country or Region	ISO Code
Portuguese	Portugal	pt_PT
Romanian	Romania	ro_RO
Russian	Russia	ru_RU
Serbian-Cyrillic	Serbia and Montenegro	sr_Cyrl_CS
Serbian-Latin	Serbia and Montenegro	sr_Latn_CS
Slovak	Slovakia	sk_SK
Slovenian	Slovenia	sl_SI
Spanish	Argentina	es_AR
Spanish	Bolivia	es_BO
Spanish	Chile	es_CL
Spanish	Columbia	es_CO
Spanish	Costa Rica	es_CR
Spanish	Dominican Republic	es_DO
Spanish	Ecuador	es_EC
Spanish	El Salvador	es_SV
Spanish	Guatemala	es_GT
Spanish	Honduras	es_HN
Spanish	Mexico	es_MX
Spanish	Nicaragua	es_NI
Spanish	Panama	es_PA
Spanish	Paraguay	es_PY
Spanish	Peru	es_PE
Spanish	Puerto Rico	es_PR
Spanish	Spain	es_ES
Spanish	United States of America	es_US
Spanish	Uruguay	es_UY
Spanish	Venezuela	es_VE
Swedish	Sweden	sv_SE

Language	Country or Region	ISO Code
Tagalog	Philippines	tl_PH
Thai	Thailand	th_TH
Thai	Thailand Traditional	th_TH_TH
Turkish	Turkey	tr_TR
Turkish (Turkey Lira)	Turkey	tr_TR_LIRA
Ukrainian	Ukraine	uk_UA
Vietnamese	Vietnam	vi_VN

Usually, both elements of a locale are important. For example, the names of weekdays and months, in English, for Canada and Great Britain are formatted identically, but dates are formatted differently. Therefore, specifying an English language locale is insufficient. Also, specifying only a country as the locale is insufficient. For example, Canada has different date formats for English and French. For information about how to set the locale in Designer, see *To specify a locale (language and country or region) for an object*.

In general, every application operates in an environment where a locale is present. This locale is known as the *ambient locale*. In some circumstances, an application might operate on a system, or within an environment, where a locale is not present. In these rare cases, the ambient locale is set to a default of English United States (en_US). This locale is known as a *default locale*.

Epoch

Date values and time values have an associated origin or *epoch*, which is a moment in time from which time begins. Any date value and any time value prior to its epoch is invalid.

The unit of value for all date functions is the number of days since the epoch. The unit of value for all time functions is the number of milliseconds since the epoch.

Designer defines day one for the epoch for all date functions as Jan 1, 1900, and millisecond one for the epoch for all time functions is midnight, 00:00:00, Greenwich Mean Time (GMT). This definition means that negative time values can be returned to users in time zones east of GMT.

Date formats

A *date format* is a shorthand specification of how a date appears. It consists of various punctuation marks and symbols that represent the formatting that the date must use. The following table lists examples of date formats.

Date format	Example
MM/DD/YY	11/11/78

Date format	Example
DD/MM/YY	25/07/85
MMMM DD, YYYY	March 10, 1964

The format of dates is governed by an ISO standard. Each country or region specifies its own date formats. The four general categories of date formats are short, medium, long, and full. The following table contains examples of different date formats from different locales for each of the categories.

Locale identifier and description	Date format (Category)	Example
en_GB English (United Kingdom)	DD/MM/YY (Short)	08/12/92 08/04/05
fr_CA French (Canada)	YY-MM-DD (Medium)	92-08-18
de_DE German (Germany)	D. MMMM YYYY (Long)	17. Juni 1989
fr_FR French (France)	EEEE, ' le ' D MMMM YYYY (Full)	Lundi, le 29 Octobre, 1990

Time formats

A *time format* is a shorthand specification to format a time. It consists of punctuations, literals, and pattern symbols. The following table lists examples of time formats.

Time format	Example
h:MM A	7:15 PM
HH:MM:SS	21:35:26
HH:MM:SS 'o''clock' A Z	14:20:10 o’clock PM EDT

Time formats are governed by an ISO standard. Each nation specifies the form of its default, short, medium, long, and full-time formats. The locale identifies the format of times that conform to the standards of that nation.

The following table contains some examples of different date formats from different locales for each of the categories.

Locale identifier and description	Time format (Category)	Example
en_GB English (United Kingdom)	HH:MM (Short)	14:13

Locale identifier and description	Time format (Category)	Example
fr_CA French (Canada)	HH:MM:SS (Medium)	12:15:50
de_DE German (Germany)	HH:MM:SS z (Long)	14:13:13 -0400
fr_FR French (France)	HH ' h ' MM Z (Full)	14 h 13 GMT-04:00

Date and time patterns

The following symbols must be used to create date and time patterns for date/time fields. Certain date symbols are only used in Chinese, Japanese, and Korean locales. These symbols are also specified below. For more information, see *Examples of date/time patterns*.

NOTE: The comma (,), dash (-), colon (:), slash (/), period (.), and space ( ) are treated as literal values and can be included anywhere in a pattern. To include a phrase in a pattern, delimit the text string with single quotation marks ('). For example, `'Your payment is due no later than' MM-DD-YY` can be specified as the display pattern.

Date symbol	Description	Formatted value for English (USA) locale where the locale-sensitive input value is 1/1/08 (which is January 1, 2008)
D	1 or 2 digit (1-31) day of the month	1
DD	Zero-padded 2 digit (01-31) day of the month	01
J	1, 2, or 3 digit (1-366) day of the year	1
JJJ	Zero-padded, three-digit (001-366) day of the year	001
M	One- or two-digit (1-12) month of the year	1
MM	Zero-padded, two-digit (01-12) month of the year	01
MMM	Abbreviated month name	Jan
MMMM	Full month name	January
E	One-digit (1-7) day of the week, where (1=Sunday)	3 (because January 1, 2008 is a Tuesday)
EEE	Abbreviated weekday name	Tue (because January 1, 2008 is a Tuesday)

Date symbol	Description	Formatted value for English (USA) locale where the locale-sensitive input value is 1/1/08 (which is January 1, 2008)
EEEE	Full weekday name	Tuesday (because January 1, 2008 is a Tuesday)
YY	Two-digit year, where numbers less than 30 are considered to fall after the year 2000 and numbers 30 and higher are considered to occur before 2000. For example, 00=2000, 29=2029, 30=1930, and 99=1999	08
YYYY	Four-digit year	2008
G	Era name (BC or AD)	AD
w	One-digit (0-5) week of the month, where week 1 is the earliest set of four contiguous days ending on a Saturday	1
WW	Two-digit (01-53) ISO-8601 week of the year, where week 1 is the week containing January 4	01

Several additional date patterns are available for specifying date patterns in Chinese, Japanese, and Korean locales.

Japanese eras can be represented by several different symbols. The final four era symbols provide alternative symbols to represent Japanese eras.

CJK date symbol	Description
DDD	The locale's ideographic numeric valued day of the month
DDDD	The locale's tens rule ideographic numeric valued day of the month
YYY	The locale's ideographic numeric valued year
YYYYY	The locale's tens rule ideographic numeric valued year
g	The locale's alternate era name. For the current Japanese era, Heisei, this pattern displays the ASCII letter H (U+48)
gg	The locale's alternate era name. For the current Japanese era, this pattern displays the ideograph that is represented by the Unicode symbol (U+5E73)
ggg	The locale's alternate era name. For the current Japanese era, this pattern displays the ideographs that are represented by the Unicode symbols (U+5E73 U+6210)

CJK date symbol	Description
g	The locale's alternate era name. For the current Japanese era, this pattern displays the full width letter H (U+FF28)
g g	The locale's alternate era name. For the current Japanese era, this pattern displays the ideograph that is represented by the Unicode symbol (U+337B)

Time symbol	Description	Locale-sensitive input value	Formatted value for English (USA) locale
h	One- or two-digit (1-12) hour of the day (AM/PM)	12:08 AM or 2:08 PM	12 or 2
hh	Zero-padded 2 digit (01-12) hour of the day (AM/PM)	12:08 AM or 2:08 PM	12 or 02
k	One- or two-digit (0-11) hour of the day (AM/PM)	12:08 AM or 2:08 PM	0 or 2
kk	Two-digit (00-11) hour of the day (AM/PM)	12:08 AM or 2:08 PM	00 or 02
H	One- or two-digit (0-23) hour of the day	12:08 AM or 2:08 PM	0 or 14
HH	Zero-padded, two-digit (00-23) hour of the day	12:08 AM or 2:08 PM	00 or 14
K	One- or two-digit (1-24) hour of the day	12:08 AM or 2:08 PM	24 or 14
KK	Zero-padded, two-digit (01-24) hour of the day	12:08 AM or 2:08 PM	24 or 14
M	One- or two-digit (0-59) minute of the hour ***NOTE:*** *You must use this symbol with an hour symbol.*	2:08 PM	8
MM	Zero-padded, two-digit (00-59) minute of the hour ***NOTE:*** *You must use this symbol with an hour symbol.*	2:08 PM	08
S	One- or two-digit (0-59) second of the minute ***NOTE:*** *You must use this symbol with an hour and minute symbol.*	2:08:09 PM	9

Time symbol	Description	Locale-sensitive input value	Formatted value for English (USA) locale
SS	Zero-padded, two-digit (00-59) second of the minute ***NOTE:*** *You must use this symbol with an hour and minute symbol.*	2:08:09 PM	09
FFF	Three- digit (000-999) thousandth of the second ***NOTE:*** *You must use this symbol with an hour, minute, and seconds symbol.*	2:08:09 PM	09
A	The part of the day that is from midnight to noon (AM) or from noon to midnight (PM)	2:08:09 PM	PM
z	ISO-8601 time-zone format (for example,Z, +0500, -0030, -01, +0100) ***NOTE:*** *You must use this symbol with an hour symbol.*	2:08:09 PM	-0400
zz	Alternative ISO-8601 time-zone format (for example, Z, +05:00, -00:30, -01, +01:00) ***NOTE:*** *You must use this symbol with an hour symbol.*	2:08:09 PM	-04:00
Z	Abbreviated time-zone name (for example, GMT, GMT+05:00, GMT-00:30, EST, PDT) ***NOTE:*** *You must use this symbol with an hour symbol.*	2:08:09 PM	EDT

Reserved symbols

The following symbols have special meanings and cannot be used as literal text.

Symbol	Description
?	When submitted, the symbol matches any one character. When merged for display, it becomes a space.
*	When submitted, the symbol matches 0 or Unicode white space characters. When merged for display, it becomes a space.
+	When submitted, the symbol matches one or more Unicode white space characters. When merged for display, it becomes a space.

Complex field patterns

In addition to defining simple patterns for date/time fields, numeric fields, and text fields, you can define a locale-specific pattern or handle variable patterns.

Locale-specific patterns

If you want to force a locale on a pattern, regardless of the locale that has already been assigned to an object, you can define a locale-specific pattern. The syntax of a locale-specific pattern is defined as follows:

```
category_name(locale_name){pattern}
```

where

- `category_name` can be `date`,`time`,`num`, or`text`.
- `locale_name` is identified by a language and/or country or region code, as defined in RFC 1766 (Tags for the Identification of Languages, 1995).
- `pattern` is the simple pattern for processing values.

For example, to force a date/time field to translate a date into the French language according to France's country code, you would define the pattern as follows:

```
date(fr_FR){DD MMMM, YYYY}
```

Variable patterns

In cases where the user input or bound data is available in more than one format (for example, telephone numbers may or may not have a three-digit area code), you can define a pattern that accounts for the differences. The syntax for defining a number of acceptable patterns is as follows:

```
category_name{pattern}|category_name{pattern}|category_name{pattern}
```

where each pattern is separated by a vertical bar (|). You can specify an unlimited number of patterns. For example, the following construct handles two different text patterns:

```
text{999*9999}|text{999*999*9999}
```

To set a default font for values in new forms

1) On the Tools menu, select Options.
2) Click Default Fonts.
3) Under Default Value Font Properties For New Forms, select Typeface, Size, and Style options. as needed.

To set a default font for values in an existing form

1) Click File > Form Properties.
2) Click Default Fonts.
3) Under Default Value Font Properties, select Typeface, Size, and Style options. as needed.

RELATED LINKS:

Form Validation (Form Properties dialog box)

To dynamically populate a validation pattern message

Numeric patterns

Text field patterns

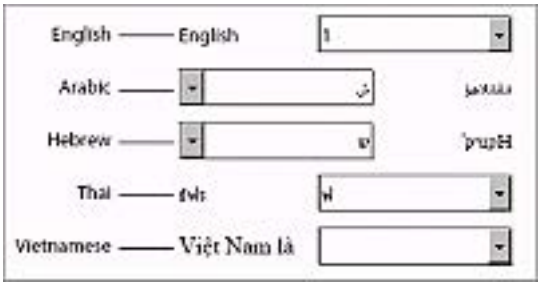
Password patterns

10.11. Setting up an object for other languages

To set up an object for Arabic and Hebrew

Objects, which are the building blocks of every form, have different appearances depending on the language the form is designed for.

For example, a drop-down list in an English form has the caption typically on the left and the drop-down arrow on the right. However, in Arabic and Hebrew, the drop-down list has the caption on the right and the drop-down arrow on the left because these are right-to-left languages.



To set up an object for Arabic and Hebrew, you must do the following steps:

- Set the locale to Arabic or Hebrew.
- Position the caption on the right (except for check boxes and radio buttons, which need to be positioned on the left).
- Right-align the caption and value.
- Set the font to one that the language supports.

Setting up an object for Arabic and Hebrew applies to the following objects:

- Check boxes
- Date/time fields
- Decimal fields
- Signature Fields
- Drop-down lists
- List boxes

- Numeric fields
- Password fields
- Radio buttons
- Text fields

NOTE: The options described in this Help topic are available only if support for the appropriate language is enabled through Microsoft Office Language Settings.

1) Add the object.
   For example, from the Object Library palette, drag the Drop-Down List object onto the form design.
2) In the Object palette, click the Field tab.
3) From the Locale box, select a language and country or region option.
   For example, select Arabic (Bahrain).
4) Select the object.
5) In the Font palette, select the font that supports the locale you selected.
   For example, Adobe Arabic supports Arabic and Adobe Hebrew supports Hebrew. You can locate the font you need for your language on the Internet if it is not already on your system.
6) Select the object.
7) In the Paragraph palette, click Right Align to right-align the caption and value.
8) In the Layout palette, select the appropriate option from the Position list for the caption.

Object	Position
Check boxes	Left
Date/time fields	Right
Decimal fields	Right
Signature Fields	Right
Drop-down lists	Right
List boxes	Right
Numeric fields	Right
Password fields	Right
Radio buttons	Left
Text fields	Right

TIP: View the object in the Preview PDF tab to ensure that it appears properly.

To set up an object for Thai and Vietnamese

To set up an object for Thai and Vietnamese, you must do the following steps:

- Set the locale to Thai or Vietnamese.
- Set the font to one that the language supports.

  NOTE: The options described in this Help topic are available only if support for the appropriate language is enabled through Microsoft Office Language Settings.

  a) Add the object.

  For example, from the Object Library palette, drag the Drop-Down List object onto the form design.

  b) In the Object palette, click the Field tab.

  c) From the Locale box, select a language and country or region option.

  For example, select Thai (Thailand Traditional).

  d) Select the object.

  e) In the Font palette, select the font that supports the locale you selected.

  For example, Adobe Thai supports Thai and Myriad® Pro and Minion® Pro support Vietnamese. You can locate the font you need for your language on the Internet if it is not already on your system.

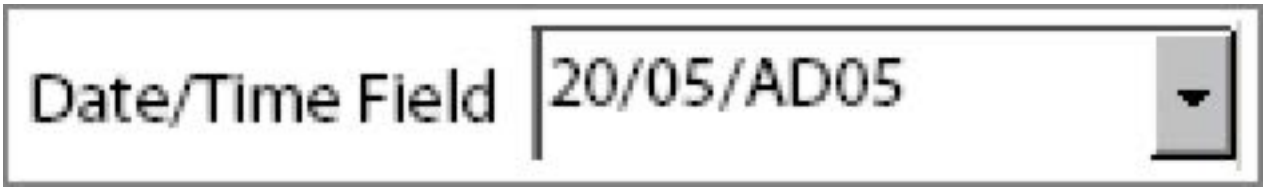
  f) View the object in the Preview PDF tab to ensure that it appears properly.

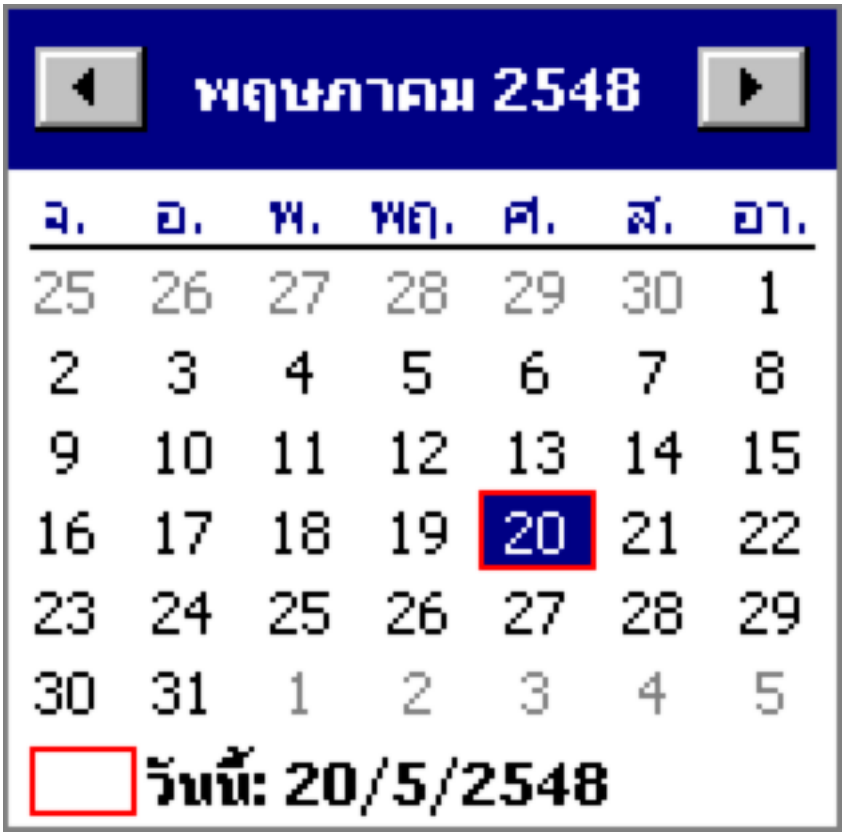
To set up a date field for other languages

1) Ensure that the regional and language options are set to the appropriate language in the Microsoft® Windows® Control Panel.
2) In Designer, drag the Date/Time field object onto the form design from the Object Library palette.
3) In the Object palette, select the Field tab and click the Patterns button.
4) On the Display tab, in the Pattern box, type `DD/MM/YY`.

   For example, for Thai dates use the following:

   – `g` shows B.E
   – `gg` shows the abbreviation of B.E in Thai
   – `ggg` shows the full name of B.E in Thai
5) In the Preview PDF tab, the Date/Time field and calendar look like this.





10.12. Using special objects

In addition to the standard and custom objects in the Object Library palette, Designer includes a number of objects and/or properties that provide special functionality in a form at run time. These objects and properties include:

- Global fields
- Floating fields
- Run time properties
- Variables

To define a global field

A global field contains information that appears in multiple locations on your form. For example, an invoice contains an invoice number that must appear in multiple places on the form. In the form design, you could set the invoice number to be a global field and reuse it elsewhere as needed.

Global fields are extremely useful when you have information that you know will be repeated in multiple places. Not only does the use of global fields reduce the amount of data that needs to be sent, they help ensure that exactly the same data appears in the necessary areas of the form.

Using the Global binding property, you can apply the same value to all objects in the form that have the same name.

When you apply the global setting to an object, all objects with the same name will be bound to the same data at run time. Because identically named global objects are linked to the same data value, the data displayed in one global object is automatically displayed in all other global objects having the same name. You cannot have more than one identically named object in a form where some, but not all, of the objects are set to global.

You can apply the global binding setting to the following types of objects:

- check boxes
- date/time fields
- drop-down lists
- image fields
- list boxes
- numeric fields
- decimal fields
- password fields
- radio buttons
- text fields
- barcoded fields (excluding Paper Form Barcode)

When you apply the global setting to an object, Designer automatically applies the global value to all other objects in the form with the same name. Conversely, if you remove the global setting from an object, Designer removes the value from all other objects with the same name and setting.

NOTE: When you apply global binding to an object, you cannot add run-time properties such as the current page, number of pages, and current date/time to that object.

1) Add the required objects to the form design.
2) Give each object the same name.
3) Select one of the like-named objects.
4) In the Object palette, click the Binding tab and select Use Global Data from the Data Binding list. Designer displays a message confirming that global binding will be applied to all objects that have the same name as the selected object.

RELATED LINKS:

Working with Data Sources

To name and rename objects

To insert a floating field

You can insert a floating field into a text object to display different types of information. Floating fields are objects that support the merging of text, numeric values, run-time properties, and scripting within a text object when the form is rendered. You can insert floating fields into text objects only.

Floating fields are text field objects by default, but you can change the object type from the Type list in the Field tab of the Object palette.

You can also bind floating fields to a data source to display specific text or numeric values. For example, you can insert a floating field to render individual customer names in the introductory greeting of a form letter. The floating field appears as a TextField object, as shown in this example.

```
(Dear Mr./Ms. {TextField})
```

1) Click inside the Text object in which you want to place the floating Text Field object.
2) Place your cursor on the line where you want Designer to insert the floating Text Field object.
3) Select Insert > Floating Field.
4) To set the properties of the Floating Field, click the Floating Field marker inside the Text object, and then select options as needed. It is good practice to rename the floating Text Field object and specify the necessary binding.

   TIP: Set the properties of the floating field object independently of setting the properties of the text object. By default, floating field objects are set to Hidden (Exclude from layout) in the Presence list.

If you want to dynamically update the value of a floating field inside Adobe Reader (through scripting, for example), you must use Adobe Reader 9.1 or later. Set the target version to Acrobat and Adobe Reader 9.1 or later and save the form as a dynamic PDF form. If you expect a large amount of text to exceed the bounds of the floating field, set the *Expand to Fit* option for both the floating field and its parent text container to make them growable.

RELATED LINKS:

To insert run-time properties

Using text

Using text fields

To insert run-time properties

Using run-time properties in a form design, you can display certain types of information in the form at run time. The types of information you can display are the current page, page count, current date/time, viewer locale, viewer name, and viewer version.

You can insert run-time properties into the following objects:

- Numeric field
- Text
- Text field

You can only insert run-time properties into the value portion of the field and not the field's caption.

You use the commands in the Insert menu to add run-time properties to a text object. Alternatively, you can use the Type list to add run-time properties to either a text field object or a numeric field object. Designer adds the run-time property to the selected object, along with a calculate script that generates the intended value and a default value based on the property.

NOTE: When you add the current page run-time property to an object, do not apply the global binding option as well because Designer will apply the same page number on every page instead of unique values.

To add a run-time property to a text object

1) Click the text field.
2) Select Insert and select the run-time property you want to insert.

To add a run-time property to a text field or numeric field object

1) Click the text field or numeric field.
2) On the Object palette, click the Value tab and select Calculated - Read Only from the Type list. The Runtime Properties option appears.
3) Select Runtime Properties and, in the list below it, select the run-time property you want to insert.

RELATED LINKS:

About numeric fields

Using text

Using text fields

To add variables to a form

Use variables to insert varying text or values. A variable is a segment of text or values that Designer can automatically update or change (on demand) using scripting.

The variables element can hold any number or any kind of separate data items. Because each data item is individually named with its own name attribute, they can be individually addressed by scripts.

You can define text variables and then insert them into captions or other text. For example, you could apply variables to product terminology that may change at some point. If the variable (term) needs to change, you can open the affected form and update the variable value, and Designer will automatically update all instances of the term.

To create a variable, you must name the variable and provide the value you want Designer to use. For example, you might create a variable called ProductName that contains the current name of a product, which you can update if needed.

Variable values are saved with the form.

RELATED LINKS:

Variables tab (Form Properties dialog box)

10.13. Adding repeating fields in a form design

About repeating text fields

You can use the Subform Instance Controls custom objects to add repeating text fields in a form design. For example, a company sends an order form to their end users. The end user clicks a button to add as many line items to the form as required.

There are two Subform Instance Controls custom objects:

Subform Instance Controls - Add

Adds a new instance of a repeating subform. The object consists of a single button with a click event specified to run at the client.

Subform Instance Controls - Insert Remove Move

Inserts, removes, and moves a new instance of a repeating subform. The object consists of four buttons with a click event specified at the client.

Moves the current instance up and down in the form hierarchy.

Inserts one instance below the current one.

Removes the current instance.

To add repeating fields

The following instructions provide guidance on how to add repeating text fields to a form design. You must customize the script in the Script Editor to make it work for your form design.

1) Add the Subform Instance Controls - Add and Subform Instance Controls - Insert Remove Move objects to your existing form design.
2) Edit the caption text and define the object's properties in the Field tab of the Object palette.

NOTE: Do not change the button's type, because it will delete the script.

1) In the Hierarchy palette, move the Add Item button under the repeating subform's header. In doing so, the button remains visible, even when all instances of the repeating subform are removed from the form.
2) If there is a possibility that the repeating subform will spill over onto a new page, move another instance of the Add Item button under the overflow leader subform in the Hierarchy palette.
3) In the Hierarchy palette, move the insert, remove, and move buttons under the repeating subform.
4) Customize the buttons' scripts in the Script Editor. Here are some examples:
   – For the Add Item button, provide the SOM expression of the repeating subform:`var sSubformSOM = "xfa.form.form1.Subform1";` The SOM expression may vary upon the position of the button relative to the repeating subform.
   – For the Add Item, Insert, and Remove buttons, set the bCalc variable to true if the fields of the new subform instance are referenced by other calculations.

5) Save, preview and test the form in the PDF Preview tab. You may want to enable the JavaScript Debugger to catch any errors while testing the form. See “Debugging Calculations and Scripts” in *Scripting Using Designer* for more information about the JavaScript Debugger.

RELATED LINKS:

Using text fields

Text field properties in the Binding tab

To define custom data-binding properties for text fields

10.14. Adding URL links to a PDF form

About URL links

You can use the Launch URL custom object to add URL links to a PDF form. At run time, a user can click the button to launch the URL in a new browser window.

The Launch URL custom object contains a button and a hidden text field nested inside a subform with a binding set to none. The button’s click event script launches the URL. The URL is specified by invoking the`launchURL() App` method in the Acrobat JavaScript object model.

Button

The button contains script that launches the URL, displays the button tool tip, and displays an error message if the URL is not available.

Hidden text field

The hidden text field contains the script that sets the tool tip text of URLButton1. It is a`form:ready` script that looks up the hidden field’s raw value to set the button’s tool tip.

You make a URL launch in a PDF form by hard-coding the URL in the form design or passing the URL from data.

RELATED LINKS:

Binding fields to a data source

To hard-code a URL link

To pass a URL from data

To hard-code a URL link

When adding the Launch URL custom object to a form design, you can hard-code a URL link by typing the URL value in the hidden field, URL Field1. See *To align objects to each other* for more information about the Launch URL custom object.

1) In the Object Library palette, click the tab that contains the Launch URL object and drag it onto the form design.
2) Select the hidden field, URLField1.
3) In the Object Palette, click the Value tab and type the URL in the Default box.
4) If you want to prevent the URL from being submitted with the data, do the following steps:
   – Click the Binding tab.
   – In the Default Binding box, select None.
5) Save, preview, and test the form in the PDF Preview tab. You may want to enable the Acrobat JavaScript Debugger to catch any errors while testing the form. See "Debugging Calculations and Scripts" in *Scripting Using Designer* for more information about the JavaScript Debugger.

To pass a URL from data

When adding the Launch URL custom object to a form design, you can pass the URL value from data by binding the hidden field, URLField1, explicitly or implicitly. Binding explicitly means binding the field to an XML schema. Binding implicitly means letting Adobe document services find and map the field to the URL value. See *To align objects to each other* for more information about the Launch URL custom object.

To bind the hidden field explicitly

1) In the Object Library palette, click the tab that contains the Launch URL object and drag it onto the form design.
2) Select the hidden field, URLField1.
3) In the Object Palette, click the Binding tab and bind URLField1 to the XML schema.
4) Save, preview, and test the form in the PDF Preview tab.

To bind the hidden field implicitly

1) In the Object Library palette, click the tab that contains the Launch URL object and drag it onto the form design.
2) Rename the hidden field, URLField1, to match the data node name and structure.
3) In the Object Palette, click the Binding tab and set the binding to Normal.
4) Click the Value tab and remove the URL value in the Default box.
5) Save, preview, and test the form in the PDF Preview tab.

NOTE: You may want to enable the JavaScript Debugger to catch any errors while testing the form. See "Debugging Calculations and Scripts" in *Scripting Using Designer* for more information about the JavaScript Debugger.

10.15. Setting a PDF form action upon submission of data

About setting a PDF form action

You can use the SubmitSuccessClose or SubmitSuccessReadOnly custom object to set an action when a user successfully submits form data. The object contains a script in the `form:ready` event that executes automatically when new data arrives from the server.

The script in both objects relies on the value of a specific field. The name of the data field is $record.CONTROL_PARAM.SUBMIT.RESULT. The value of the field is `true` upon a successful submit.

You can use the object when you have not bound the `pdfSource` attribute of InteractiveFormsElement in your Web Dynpro application:

- The application server processes the submit request from the form.
- The form data includes a special field that indicates the result of the submit request.
- The application server sets the value of the special field after it successfully processes the submit request and returns the XML data to the client application.
- The XML data merges into the existing data DOM.
- The merging of the XML data into the existing data DOM triggers the `form:ready` scripts of all fields on the form. All `form:ready` events execute when the XML form data is modified.
- The script inside the `form:ready` event inspects the value of the field and then takes the appropriate action. The action is to set all fields in the form to read-only or close the form, depending on the object you use.

RELATED LINKS:

Using buttons

To set a PDF form to close after data submission

The SubmitSuccessClose object is a subform that contains a script that closes a PDF form after a successful data submission by a user. The `form:ready` script invokes the closing of the PDF form.

See the object's script for more details.

1) In the Object Library palette, click the tab that contains the Successful Submit Close object and drag it onto the form design.
2) Create a Submit button and add the following script:

```
$record.CONTROL_PARAM.SUBMIT_RESULT.value = "true";
$form.remerge();
```

3) Save and test the form in the PDF Preview tab. You cannot test the closing of the form within Designer. You may want to enable the JavaScript Debugger to catch any errors while testing the form. See "Debugging Calculations and Scripts" in *Scripting Using Designer* for more information about the debugger.

To set PDF form fields to read-only

The SubmitSuccessReadOnly object is a subform that contains a script that makes PDF form fields read-only after a successful data submission by a user. The `form:ready` script changes the PDF form fields to read-only.

See the object's script for more details.

1) Add the SubmitSuccessReadOnly object to your form design.
2) Create a Submit button and add the following script to it:

```
$record.CONTROL_PARAM.SUBMIT_RESULT.value = "true";
$form.remerge();
```

3) Save, preview, and test the form in the PDF Preview tab. You may want to enable the JavaScript Debugger to catch any errors while testing the form. See "Debugging Calculations and Scripts" in *Scripting Using Designer*for more information about the debugger.

11. Object Properties

11.1. Layout properties in the Layout palette

Using the Layout palette, you can specify a number of characteristics that apply to the selected object, including its size and position, margins, and location of captions. You can specify layout properties for each object on the page.

X and Y

Sets the horizontal and vertical position of the object's anchor point. If the setting in the Anchor list is not Top Left, X and Y options change to AnchorX and AnchorY. See *To position objects*

Width, Height, and Expand to Fit/Auto-fit

Sets the minimum overall height and width of the object and enables expansion in that direction if required. See *To position objects* and *To make objects expand to fit*.

NOTE: To manually expand objects, you can also select the Show Text Overflow Indicators option on the Wizards and Tips panel in the Options dialog box. See *Wizards and Tips (Options dialog box)*.

The Auto-fit options appears for subform objects only. When selected, the subform changes size in Designer to fit the enclosed content.

Anchor

Sets the object's anchor point. See *To rotate objects*.

Rotate

Rotates the object around its anchor point. See *To rotate objects*.

Content Alignment in a Flowed Container

Sets the alignment of the selected object in a subform that flows content.

Margins

Sets the amount of white space on the left and right sides of the object and above and below the object. See *Formatting objects*.

Caption

Sets the position of a caption. See *Creating an insertion point*.

Reserve

Sets the amount of space reserved for a caption. See *Formatting objects*.

A reserve of 0 sets the caption area to auto-fit. It adjusts the size of the object to fit the caption.

11.2. Border properties in the Border palette

Using the Border palette, you can edit the border properties for objects that have borders. You can edit the borders individually or together.

Edges

Sets border properties for all or individual edges:

Edit Individually

Applies a border to each edge separately. This option does not apply to three-dimensional borders.

Edit Together

Applies the same border to all edges.

Side

Sets the style and width of the borders.

Color Picker

Sets the line color.

Corners

Applies a style to border corners.

Radius

Sets the radius of notched corners.

Omit Border Around Page Breaks

Omits the selected object's bottom border on the first page and top border on the second page.

Style

Sets the background fill color or pattern:

Color Picker

Sets the background fill colors.

11.3. Font properties in the Font palette

Using the Font palette, you can change the font family, size, style, and scale, as well as the baseline shift, letter spacing, and kerning of the text in one or more selected objects.

Font

Sets the typeface.

Font Size

Sets the typeface size.

Style

Sets the typeface style.

Baseline Shift

Sets the amount of space to move a character (or group of characters) up or down relative to the baseline. Positive numbers shift characters up, and negative numbers shift characters down.

Letter Spacing

Sets the amount of space to leave between the letters in a word or between a group of words.

Vertical Scale

Sets the degree (percentage) by which to increase or decrease the vertical size of text.

Horizontal Scale

Sets the degree (percentage) by which to increase or decrease the horizontal size of text.

Auto Kern

Reduces the amount of space between letters within a word or group of words.

NOTE: The Font Size and Style options are available with fixed size fonts (for example, Courier New). However, it is recommended that you do not modify fixed sized fonts. Fixed sized fonts are printed in their original size and style. (See XDC Editor Help.)

11.4. Paragraph properties in the Paragraph palette

Using the Paragraph palette, you can change the justification, indentation, and line spacing of the selected text.

Align

Sets horizontal alignment options:

Align Left

Left-aligns the caption or value.

Align Center

Center-aligns the caption or value.

Align Right

Right-aligns the caption or value.

Justify

Justifies the caption or value. Full justification is applied to all of the lines except the last line in a multiple-line caption or value (single-line paragraphs cannot be justified).

Radix

Sets the alignment for numeric fields based on the radix (decimal point). This option appears on the Paragraph palette only when a numeric field object is selected.

IMPORTANT: Numeric fields with a defined radix alignment setting are not supported in PDF forms for Acrobat 6.0.2 and Adobe Reader 6.0.2.

Vertical Alignment

Sets vertical alignment options:

Align Top

Aligns to the top of the area reserved for the caption and value.

Align Middle

Aligns to the middle of the area reserved for the caption and value.

Align Bottom

Aligns to the bottom of the area reserved for the caption and value.

Lists

Sets list options:

Bulleted List

Starts a bulleted list. Click the arrow to choose different bullet styles.

Numbered List

Starts a numbered list. Click the arrow to choose different number styles.

Decrease Indent

Decreases the indent level of the list.

Increase Indent

Increases the indent level of the list.

Start

Sets the start number for the selected item in the numbered list.

Compound Tags

Creates a multi-level numbered list.

First

Sets the first line indentation:

None

Indents all lines by the same amount.

First line

Indents the first line only.

Hanging

Indents all lines except the first.

By

Sets the amount of the first line or hanging indentation.

Spacing

Sets the amount of space above and below the paragraph:

Above

Sets the amount of vertical space above the paragraph. This value is added to any value specified in the Below box.

Below

Sets the amount of vertical space below the paragraph. This value is added to any value specified in the Above box.

Line Spacing

Sets the amount of space between lines:

Single

Makes the line height equivalent to the height of the tallest character in the line.

1.5 Lines

Makes the line height one and a half times the height of the tallest character in the line.

Double Line

Makes the line height twice the height of the tallest character in the line.

Exactly

Makes the line height equivalent to the setting in the At box.

Hyphenate

Adds or removes hyphenation in individual objects (text objects, the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, and the value area of text field

objects). Use the Hyphenate option to manually adjust the layout of text on an object-by-object basis. This option is available when you select Allow Hyphenation in Text and Field Captions, or Allow Hyphenation in Text Field Values, or both.

11.5. Accessibility properties in the Accessibility palette

Designer includes a number of options that support screen readers. For each field object in a form, you can specify one of several settings for screen reader text:

- Custom screen reader text, which you set in the Accessibility palette
- Tool tips for objects, which you set in the Accessibility palette
- Captions for fields
- Names of objects, as specified in the Name option in the Binding tab

  The settings determine the information that screen readers read for objects on PDF forms. Only one setting is spoken for each object.

  When the form is saved as tagged PDF, Designer searches the form for these settings. The default search order is Custom Text, Tool tip, Caption, and Name. You can override this default order by using the Screen Reader Precedence option in the Accessibility palette.

  TIP: You can dynamically populate a tool tip and custom screen reader text with values from a data source.

  NOTE: Tool tips appear at run time when the user hovers the pointer over an object. The settings on the Accessibility palette have no effect on objects when the form is rendered as a PDF form.

Role

Determines how screen readers interpret the subform, table, selected row in a table, list, heading, or heading level:

Table

Assigns the role of a table to the selected subform. When the user navigates to this subform, most screen readers identify it as a table and indicate the number of rows and columns.

Header Row

Assigns the role of a header row to the selected subform or table row. When speaking the contents of a body row cell, most screen readers first identify the content of the corresponding cell in the header row.

Body Row

Assigns the role of a body row to the selected subform or table row. If a cell contains a subform, screen readers typically speak the content of the corresponding cell in the header row, followed by the fields in the subform.

Footer Row

Assigns the role of a footer row to the selected subform or table row.

List

Assigns the role of a list to the selected subform.

List Item

Assigns the role of a list item to the selected subform. A list item role can only be assigned to a subform that is contained in a subform that has a List role specified. You cannot define a table or table row as a list or list item; however, a list item can contain a table.

Heading

Assigns the role of heading to the selected text object.

Heading Level 1 to Heading Level 6

Assigns the role of the heading level to the selected text object.

(None)

Specifies a row that conveys information about the table or its content. The row is not considered to be part of the table; however, the screen reader will read its contents.

Tool Tip

Defines a tool tip for the object. Tool tips appear at run time when the user hovers the pointer over the object. A screen reader can read the text entered in this box.

You cannot have both unique custom tool-tip text and unique custom screen-reader text for one object. You must choose one or the other. If you want to use the same text for the tool tip and the text read by the screen reader, type a tool tip and select Tool Tip from the Screen Reader Precedence list.

NOTE: Tool Tip is a dynamic property for most objects. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu.

Screen Reader Precedence

Indicates which setting the screen reader should read. Only one setting is spoken for each object:

Custom Text

Reads the text specified in the Custom Screen Reader Text box. This setting is the default.

Tool Tip

Reads the text specified in the Tool Tip box.

Caption

Reads the caption specified for the object. The position of the caption relative to the object does not change the order in which the screen reader reads the caption. By default, the screen reader reads the caption if nothing is specified in the Tool Tip box or the Custom Screen Reader Text box. This setting is the preferred choice for screen readers.

Name

Reads the name of the object, as specified in the Name field in the Binding tab of the Object palette.

None

Disables the custom tool tip or custom screen reader text for the field.

Custom Screen Reader Text

Defines custom text for the selected object. The screen reader reads the text entered in this box.

You cannot have both unique custom tool tip text and unique custom screen reader text for one object. You must choose one or the other. If you want custom screen reader text to be both the tool tip text and the screen reader text, type the custom text and select Custom Text from the Screen Reader Precedence list.

NOTE: Custom Screen Reader Text is a dynamic property for most objects. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu.

11.6. Master page properties in the Master Page tab

Use the Master Page tab of the Object palette to set the page name and various layout options for each master page. To display the Master Page tab, select the master page you want to work with in the Hierarchy palette.

Name

Sets the name of the master page.

Paper Type, Height, and Width

Sets the size of the master page. The dimensions are displayed in the Height and Width boxes and can be modified when the Paper Type option is set to Custom.

Orientation

Sets the orientation of the master page:

Portrait

The page height is greater than or equal to the page width.

Landscape

The page width is greater than or equal to the page height.

NOTE: Set the page orientation explicitly as portrait or landscape for the Custom Paper Type. If the form width is greater than the form height, the orientation does not change to landscape automatically when Custom Paper Type is selected.

Restrict Page Occurrence, Min Count, and Max

Select this option only for creating forms whose layout adjusts to accommodate data. It enables or disables the repetitive rendering of pages that are based on the selected master page. When the option is selected, you can enter a minimum number of repetitions in the Min Count box and a maximum number of repetitions in the Max box.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

One-sided Printing Only

The object is printed only when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

11.7. Master page properties in the Pagination tab

When you select a Master Page, the Pagination tab in the Object palette presents a number of pagination options.

Odd / Even

Specifies whether the master page appears on the front (odd) side, back (even) side, or both sides of the paper when the rendered form is printed:

Odd (Front) Pages

The master page appears on the odd-numbered printed pages.

Even (Back) Pages

The master page appears on the even-numbered printed pages.

Blank Pages

Inserts a blank page with no flowed content. This option is useful in double-sided printing when you want the back (even) side of the printed page to be blank so that the next section starts on the front (odd) side of the next printed page.

No Odd/Even Restrictions

There are no restrictions on the master page during double-sided printing. This option is the default for all master pages.

Placement

Specifies the page that the master page is applied to in the form design:

First Page (in Page Set)

The master page is applied to the first page within the page set.

Last Page (in Page Set)

The master page is applied to the last page within the page set.

Only Page (in Document)

The master page is applied to the only page within the page set. Choose the Only Page (in Document) option when the data merged into the form does not cause the content area to flow onto more than one page. Only one master page can have a placement set to Only Page (in Document).

Rest of Pages

The master page is applied to pages between, but not including, the first and last pages generated by the page set.

No Placement Restrictions

No restriction is applied to the master page's location. No Placement Restrictions is the default placement.

Include Page in Numbering

Specifies whether pages that have been rendered according to the selected master page should contribute to the total page count. By default, the option is selected and all pages are counted.

If First Page in Document

Sets the page numbering for pages that have been rendered, according to the selected master page:

Continue Numbering from Previous Document in Batch

The numbering increments from the last document processed by Forms.

Start At

The numbering starts at this specified value.

11.8. Page set properties in the Page Set tab

Use the Page Set tab of the Object palette to set the name and occurrence options for each page set. To display the Page Set tab, select the page set you want to work with in the Hierarchy palette.

You can set these options for each page set.

Name

Sets the name of the page set.

Printing

Controls whether the page set is printed on only the front side of each sheet of paper or on both sides of each sheet of paper.

Restrict Page Set Occurrence, Min Count, and Max

Select this option only for creating interactive forms. It enables or disables the repetitive rendering of pages that are based on master pages in the selected page set. When the option is selected, you can enter a minimum number of repetitions in the Min Count box and a maximum number of repetitions in the Max box.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

One-sided Printing Only

The object is printed only when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

12. Properties in the Object palette

12.1. Barcode

Barcode properties in the Field tab

When you select a barcode, the Field tab in the Object palette displays several options for formatting barcodes.

Location

Sets the location of the barcode text. The options available on the list change depending on the barcode selected. This list is unavailable with certain barcodes.

Property/Value

Sets barcode-specific properties.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using barcodes

Barcode properties in the Value tab

When you create a barcode, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets an initial value for the barcode. The length of the default value entered can affect the displayed width of certain barcodes.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input and be compatible with the data format selected on the Binding tab. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only, or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: This option is not available when the Type option is set to Calculated - Read Only or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using barcodes

To control how a barcode obtains data

To dynamically populate a validation pattern message

To dynamically populate a validation script message

Barcode properties in the Binding tab

The Binding tab displays several options for binding barcodes. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the barcode name. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using barcodes

To define custom data-binding properties for a barcode

Supported barcode formats

The following is a list of supported barcode formats:

Aztec

Aztec format (hardware-rendered)

AUS Post Custom2

AUSPOST Custom 2 format

AUS Post Custom3

AUSPOST Custom 3 format

AUS Post Reply Paid

AUSPOST Reply Paid format

AUS Post Standard

AUSPOST Standard format

Codabar

Codabar format, which offers USD-4, NW-7, and 2-of-7 Code compatibility.

Code 11

Code 11 format (hardware-rendered)

Code 128SSCC

UCC/EAN 128 Serial Shipping Container Code (SSCC) format

Code 128

Code128 format

Code 128A

Code128, set A format

Code 128B

Code128, set B format

Code 128C

Code128, set C format

Code 2 of 5 Industrial

2 of 5 Industrial format

Code 2 of 5 Interleaved

2 of 5 Interleaved format, which offers Code 25, I2of5, ITF, and I25 compatibility

Code 2 of 5 Matrix

2 of 5 Matrix format

Code 2 of 5 Standard

Code 2 of 5 standard format (hardware-rendered)

Code 3 of 9 - 3

3 of 9 format

Code 93

Code 93 format (hardware-rendered)

Code 49

Code 49 format (hardware-rendered)

Data Matrix

Data Matrix format

GS1 Data Matrix

GS1 Data Matrix format

EAN13

EAN13 format

EAN8

EAN8 format

Japanese Postal Barcode

Japanese Postal barcode format

Logmars

Logmars format (hardware-rendered)

MSI

MSI format (hardware-rendered)

PDF 417

A two-dimensional format that accommodates up to 1 800 ASCII characters and supports binary data of upto 1 Mb in size.

Paper Forms Barcode

Paper Forms Barcode format

Planet Code

Planet Code format (hardware-rendered)

Plessey

Plessey format (hardware-rendered)

QR Code

Quick Response (QR) 2-D Code format (hardware-rendered)

RFID

Radio frequency identification barcode (hardware-rendered)

RSS 14 Stacked

RSS 14 Stacked format (hardware-rendered)

RSS 14 Omni

RSS 14 Omni format (hardware-rendered)

RSS 14 Truncated

RSS 14 Truncated format (hardware-rendered)

RSS Expanded

(RSS Expanded format (hardware-rendered)

RSS Limited

RSS Limited format (hardware-rendered)

UK Post RM4SCC

UK/Royal Mail RM4SCC format

UPC-A

UPC-A format

UPC-E

UPC-E format (hardware-rendered)

UPS Maxicode

UPS Maxicode format (hardware-rendered)

UPC EAN2

UPC EAN2 format (hardware-rendered)

UPC EAN5

UPC EAN5 format (hardware-rendered)

US Postal Zip-5

US Postal 5-digit (ZIP) format

US Postal DPBC

Delivery Point ZIP +6 format

US Postal Standard

US Postal Standard format

US Postal Intelligent Mail

US Postal Intelligent Mail Barcode

NOTE: Additional hardware-specific barcode formats can be added to the default set of supported barcode formats.

RELATED LINKS:

Using barcodes

Valid barcode text characters

This table identifies the characters that you can use in barcode text.

Barcode	Valid Characters
Aztec AUS Post Custom2 AUS Post Custom3 AUS Post Reply Paid AUS Post Standard Code 2 of 5 Industrial Code 2 of 5 Standard Code 3 of 9 - 3 Code 2 of 5 Interleaved Code 2 of 5 Matrix EAN13 EAN8 MSI UPC-A UPC-E UPC-EAN2 UPC-EAN5 US Postal Zip-5 US Postal DPBC US Postal Standard UPS Maxicode	0 1 2 3 4 5 6 7 8 9
Codabar	0 1 2 3 4 5 6 7 8 9 - $: / . +
Code 11	0 1 2 3 4 5 6 7 8 9 -
Code 128SSCC Code 128 Code 128A Code 128B Code 128C	0 1 2 3 4 5 6 7 8 9 ! # & ' < > ` $ % ( ) * + , - . / : ; ? @ # = [ ] \ ^ _ { } | ~ SPACE A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z
Code 3 of 9	0 1 2 3 4 5 6 7 8 9 - . $ / + % * SPACE A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Code 49	0 1 2 3 4 5 6 7 8 9 < > - . $ / + % : ; ? = A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Barcode	Valid Characters
Code 93	0 1 2 3 4 5 6 7 8 9 & ' - . $ ( ) / + % , SPACE A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Logmars	0 1 2 3 4 5 6 7 8 9 - . $ / + % SPACE A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
RSS 14 RSS 14 Stacked RSS 14 Stacked Omni RSS 14 Truncated RSS Expanded RSS Limited	0 1 2 3 4 5 6 7 8 9
PDF 417	No restrictions
Plessey	0 1 2 3 4 5 6 7 8 9 A B C D E F
UK Post RM4SCC	0 1 2 3 4 5 6 7 8 9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Japanese Postal	0 1 2 3 4 5 6 7 8 9 - A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

RELATED LINKS:

Using barcodes

12.2. Button

Button properties in the Field tab

When you select a button, the Field tab in the Object palette displays several options for formatting buttons.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style:

No Border

Removes the line around the button.

Solid Border

Creates a thick line around the button.

Raised Border

Creates a shadow around the button so that the button looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the button.

Highlighting

Sets the button highlight when the button is clicked:

None

Removes the button highlight.

Inverted

Inverts the button highlight when the button is clicked.

Push

Creates a shadow around the button so that the button looks like it is recessed.

Outline

Creates a line around the button when the button is clicked.

Rollover Caption

Sets a rollover caption when the button highlight is set to Push. This caption appears when the mouse pointer moves over the button.

Down Caption

Sets a down caption when the button highlight is set to Push. This caption appears when the button is clicked.

Control Type

Determines what happens when a user clicks the button. These options are not displayed for the Print or Reset buttons:

Regular

The attached script or calculation will run. The script or calculation is provided by the user.

Submit

Data will be submitted according to the settings in the Submit tab. See *Button properties in the Submit tab*.

Execute

A web-service operation or database query will be executed according to the settings in the Execute tab. See *Button properties in the Execute tab*.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using buttons

Print button properties in the Field tab

When you select a button, the Field tab in the Object palette displays several options for formatting buttons.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style:

No Border

Removes the line around the button.

Solid Border

Creates a thick line around the button.

Raised Border

Creates a shadow around the button so that the button looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the button.

Highlighting

Sets the button highlight when the button is clicked:

None

Removes the button highlight.

Inverted

Inverts the button highlight when the button is clicked.

Push

Creates a shadow around the button so that the button looks like it is recessed.

Outline

Creates a line around the button when the button is clicked.

Rollover Caption

Sets a rollover caption when the button highlight is set to Push. This caption appears when the mouse pointer moves over the button.

Down Caption

Sets a down caption when the button highlight is set to Push. This caption appears when the button is clicked.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using buttons

Reset button properties in the Field tab

When you select a button, the Field tab in the Object palette displays several options for formatting buttons.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style:

No Border

Removes the line around the button.

Solid Border

Creates a thick line around the button.

Raised Border

Creates a shadow around the button so that the button looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the button.

Highlighting

Sets the button highlight when the button is clicked:

None

Removes the button highlight.

Inverted

Inverts the button highlight when the button is clicked.

Push

Creates a shadow around the button so that the button looks like it is recessed.

Outline

Creates a line around the button when the button is clicked.

Rollover Caption

Sets a rollover caption when the button highlight is set to Push. This caption appears when the mouse pointer moves over the button.

Down Caption

Sets a down caption when the button highlight is set to Push. This caption appears when the button is clicked.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using buttons

Button properties in the Submit tab

When you create a button and set its Control Type to Submit, the Submit tab appears in the Object palette. It displays several options specific to formatting buttons that submit data.

Submit to URL

Sets the location of a web-hosted server. You can specify the ftp, http, https, or mailto protocol.

Submit As

Sets the format of the data to submit.

XML Data Package (XDP)

Submits a package in the file format created by Designer. Choose this format if the form initiates server-side processing, or to submit the form design, the form data, annotations, and all other relevant information needed for Forms to subsequently render the form at run time.

PDF

Submits a package containing an embedded PDF file. Choose this format if the form contains a signature field, or if a copy of the form along with its data needs to be saved by Forms, or submitted to another type of target server. Do not choose this option if the form initiates server-side processing, if Forms will be used to render at run time HTML forms or forms whose layout adjusts, or if the form is a PDF form that will be filled in Adobe Reader without the use of Acrobat Reader DC extensions.

XML Data (XML)

Submits an XML data stream, which allows for the hierarchical representation of data and can be parsed by any generic XML parser. Choose this format if the server that communicates with the run-time user application program must receive an XML data steam.

URL-Encoded Data (HTTP Post)

Submits a text stream to the specified Uniform Resource Locator (URL) using the `POST` method. The text stream can be parsed by an FTP server, a mail server, or a web server. To use this method, users must open the form in Adobe Reader 6.0 or later or in a web browser unless the URL specifies the mailto protocol.

Sign Submission

Applies a data signature to the submitted data. When a form filler clicks the button, a digital signature is created to cover the submitted data and attachments. Data signatures secure the signed data and guarantee the data integrity during transmission. You apply data signatures to the form data or to the entire submission, including attachments. Click the Settings button to define optional security properties for the data signature, such as the signature handler, signing certificates, and certificate issuers.

Encrypt Submission

Encrypts form content. When a form filler clicks the button, the form content is encrypted before submission to secure the form content during transmission. You apply encryption to the form data or to the entire submission, including attachments. Click the Settings button to define optional encryption properties such as encryption algorithm, encryption certificate, certificate issuers, and key usage.

Include

Sets the items to include as attachments. Attachments may be included with XDP files only.

Annotations

Includes review comments, tool tips, and any other special tags needed to capture screen reader text.

PDF (Includes Signatures)

When selected, includes a PDF version of the form when it is submitted as an attachment; otherwise, a reference to an embedded PDF file is included.

Template

Includes a copy of the form design without merged data.

Other

Includes one or more <xdp> elements in the XDP source file. The specified elements must be separated by commas and white space is optional. For example:`xci, xslt, sourceset`.

Data Encoding

Sets the encoding format for data transfers.

UTF-8

Unicode Transformation Format 8.

UTF-16

Unicode Transformation Format 16.

Shift_JIS

Shifted encoding of the Japanese Industrial Standard.

Big5

Common standard for encoding traditional Chinese characters.

GBK

Simplified Chinese (GB 13000.1-93) character encoding (an extension of GB 2312-80).

KSC_5601

Korean encoding.

GB18030

People's Republic of China (PRC) official character set encoding (supersedes GB2312).

RELATED LINKS:

Using buttons

About submitting data using a button

Button properties in the Execute tab

When you create a button and set its Control Type to Execute, the Execute tab appears in the Object palette. It displays several options specific to formatting buttons with a data connection.

Connection

Sets the connection. See *To create a data connection using a WSDL file*.

Connection Info

Provides connection information and the name of the operation or query to run.

Run At

Sets the execution location.

Client

Processes the request on the client computer.

Server

Processes the request on the server.

Client And Server

Processes the request on the client computer and server.

Re-merge Form Data

Enables or disables the updating of the form structure after processing completes.

RELATED LINKS:

Using buttons

About processing options for a button

12.3. Check box

Check box properties in the Field tab

When you select a check box, the Field tab in the Object palette displays several options for formatting check boxes.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the check box:

None

Does not include a border around the box.

Solid Square

Uses a solid square to represent the box.

Sunken Square

Creates a square shadow for the box so that it looks three dimensional.

Solid Circle

Uses a solid circle to represent the box.

Sunken Circle

Creates a circular shadow for the box so that it looks three dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

States

Sets the number of supported states:

On/Off

The check box will have On (selected) and Off (clear) states.

On/Off/Neutral

The check box will have On (selected), Off (clear), and Neutral (not selected or clear) states.

Values for each state must be defined in the Binding tab of the Object palette.

Size

Sets the size of the box.

Check Style

Sets the check style:

Default

Uses an X for the check style.

Check

Uses a check mark for the check style.

Circle

Uses a circle for the check style.

Cross

Uses a cross for the check style.

Diamond

Uses a diamond for the check style.

Square

Uses a square for the check style.

Star

Uses a star as the check style.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using check boxes

Check box properties in the Value tab

When you create a check box, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts.

User Entered

Users may choose to enter data or not.

Calculated - Read Only

A data value will be calculated and displayed at run time through an attached script. Users will not be able to edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Read Only

A data value will be merged or calculated and displayed at run time. Users will not be able to edit the value.

Default

Sets the initial state of the check box.

On

The default state is On. The check box is selected initially.

Off

The default state is Off. The check box is deselected initially.

Neutral

The default state is Neutral. The box is filled with grey initially.

NOTE: The Default list is available only when the Type option is set to User Entered or Read Only. The Neutral option is available from the Default list only when the States area in the Field tab is set to On/Off/Neutral.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protect, Calculated - Read Only, or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using check boxes

Check box properties in the Binding tab

When you create a check box, the Binding tab presents data binding options that you can apply to the object. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the check box. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

On Value

Specifies the value of the check box's On state in the data source. See *To change the values assigned to check box states*.

Off Value

Specifies the value of the check box's Off state in the data source.

Neutral Value

Specifies the value of the check box's Neutral state in the data source.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using check boxes

12.4. Circle

Circle properties in the Draw tab

When you create a circle, the Draw tab in the Object palette presents options specific to formatting circles.

Type

Sets the type of object. Objects are the building blocks of every form.

Appearance

Sets the shape of the object.

Ellipse

Draws an elliptical shape.

Circle

Draws a circular shape.

Arc

Draws an arc.

Start

Sets the start point of an arc.

Sweep

Sets the end point of an arc.

Line Style

Sets the line style and thickness.

Color Picker

Sets the line color.

Fill

Sets a fill style.

Color Picker

Sets fill colors.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using circles, lines, and rectangles

12.5. Content area

Content area properties in the Content Area tab

When a content area is selected, the Content Area tab presents two options.

Name

Sets the name of the content area.

Flow Direction

Sets the flow direction for subforms within the content area.

Top to Bottom

Sets the default tabbing order and data-fill order starting from the top of the page and moving to the bottom of the page.

Western Text

Sets the default tabbing order and data-fill order starting from the top of the page and moving to the right until the last object on the right edge of the page has been reached. When the right edge of the page has been reached, continue the tabbing order and flow direction at the next object down on the left side of the page.

Right to Left

Sets the default tabbing order and data-fill order starting from the right of the page and moving to the left of the page.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

One-sided Printing Only

The object is printed only when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using content areas

Data flow between content areas

12.6. Date/time field

Date/time field properties in the Field tab

When you select a date/time field, the Field tab displays several options specific to formatting date/time fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Limit Length to Visible Area

Sets the maximum amount of characters and numbers allowed in the field according to the horizontal length of the date/time field.

Use Cells and Cells

Enables the comb format.

Patterns

Sets the pattern for displaying formatted values in a form, for syntax of user input, for validating user input, and for storing and retrieving bound data or saving data when the form is not bound to a data source. See *Patterns dialog box*.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using date/time fields

To specify a display pattern

To specify an edit pattern

Date and time patterns

Date/time field properties in the Value tab

When you create a date/time field, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets an initial value for the field. The value is formatted according to the option selected in the Select Type list in the Patterns dialog box, but must conform to the value in the short date or time format according to the locale specified for the field. See *Date formats* and *Time formats* to learn more about short dates and times. If the data is bound and a data pattern has been specified, the value must match the data pattern specified in the Binding tab. See *To specify a default value*.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input*.

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: This option is not available when the Type option is set to Calculated - Read Only or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Calculation Script

The value will be calculated by a script written for the object's `calculate` event.

NOTE: This option is available only when the Type option is set to Calculated - Read Only.

Runtime Property

Sets the following run-time property for the object. When you select a run-time property, the actual value corresponding to the option is inserted dynamically when the form is rendered.

Current Date/Time

Display the current date and time in the date/time field.

NOTE: This option is available only when the Type option is set to Calculated - Read Only.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using date/time fields

Date/time field properties in the Binding tab

When you create a date/time field, the Binding tab presents data binding options that you can apply to the field. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the date/time field. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Data Format

Sets the format in which data in the field is displayed.

Date

Displays the data in the field in date format.

Time

Displays the data in the field in time format.

Date and Time

Displays the data in the field in date/time format.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using date/time fields

To specify the data to display in date/time fields

Date and time patterns

Examples of date/time patterns

A formatted value can be displayed or processed. A raw value conforms to Designer defaults.

Predefined patterns

The following table shows the predefined date and time patterns that you can specify by using the Patterns option in the Field tab of the Object palette and the results of formatting some examples of input default values according to the locale specified. In these examples, the locale is English (USA).

Predefined pattern	Input value (using the short format, which is M/D/YY for date and h:MM A for time)	Formatted value
YYYY-MM-DD	8/23/08	2008-08-23
EEEE, MMMM D, YYYY	8/23/08	Saturday, August 23, 2008
HH:MM:SS	5:02 PM	17:02:00
date{YYYY-MM-DD} time{HH:MM:SS}	8/23/08 5:02 PM	2008-08-23 17:02:00

NOTE: To find the default short format for date and time, select View > XML Source and search for the following lines:

```
<datePatterns>
  <datePattern name="full">EEEE, MMMM D, YYYY</datePattern>
  <datePattern name="long">MMMM D, YYYY</datePattern>
  <datePattern name="med">MMM D, YYYY</datePattern>
  <datePattern name="short">M/D/YY</datePattern>
</datePatterns>
<timePatterns>
  <timePattern name="full">h:MM:SS A Z</timePattern>
  <timePattern name="long">h:MM:SS A Z</timePattern>
  <timePattern name="med">h:MM:SS A</timePattern>
  <timePattern name="short">h:MM A</timePattern>
</timePatterns>
```

Custom patterns

The following table shows some custom display patterns that you can define for formatting input default values according to the locale specified. In these examples, the locale is English (USA).

Custom pattern	Input value (using the short format, which is M/D/YY for date and h:MM A for time)	Formatted value
MMMM DD, YYYY	8/23/08	August 23, 2008

Custom pattern	Input value (using the short format, which is M/D/YY for date and h:MM A for time)	Formatted value
EEEE, 'the' D 'of' MMMM, YYYY	8/23/08	Saturday, the 23 of August, 2008
HH:MM	5:02 PM	17:02
h:MM:SS 'o''clock' A Z	5:02 PM	5:02:00 o'clock PM EDT
HH:MM:SS Z	5:02 PM	17:02:00 EDT

12.7. Decimal field

Decimal Field properties in the Field tab

When you select a decimal field, the Field tab in the Object palette presents options for decimal fields. All but two of the options are the same as the options for numeric fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Limit Length to Visible Area

Sets the maximum amount of numbers allowed in the field according to the horizontal length of the numeric field.

Comb of <x> characters

Enables the comb format.

Patterns

Sets the pattern for displaying formatted values in a form, for syntax of the user input, for validating user input, and for storing and retrieving bound data or saving data when the form is not bound to a data source. See *Patterns dialog box*.

Limit Leading Digits and Max

Sets the maximum number of digits that can appear before the decimal separator. If the maximum is exceeded, a zero displays.

Default: Unlimited

Maximum: 15 (including trailing digits). Designer displays a warning message if the leading digits exceed 15 digits.

Limit Trailing Digits and Max

Sets the maximum number of digits that can appear after the decimal separator. If the maximum is exceeded, the number is truncated.

If you deselect the Limit Trailing Digits option, the precision of decimal data is reflected exactly as it is entered.

Default: 2

Maximum: 15 (including leading digits). Designer displays a warning message if the trailing digits exceed 15 digits.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using decimal and numeric fields

Overview of numeric fields

Numeric field properties in the Field tab

Numeric field properties in the Value tab

Numeric field properties in the Binding tab

12.8. Drop-down list

Drop-down list properties in the Field tab

When you select a drop-down list, the Field tab displays several options specific to formatting drop-down lists.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

List Items

Sets the items in the list and their positions. You can add or remove list items, paste items, move them up or down, or sort the items in ascending or descending order using the buttons beside the label.

NOTE: List Items is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu.

Allow Custom Text Entry

Enables or disables custom user entries. User-entered values must match the Designer defaults. See *Defaults for value formatting*.

Commit On

Determines when the selected option is committed.

Select

Commits the option when it is selected.

Exit

Commits the selected option when the user exits the list box, moving the focus to another object.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using drop-down lists and list boxes

Drop-down list properties in the Value tab

When you create a drop-down list, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message

appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets the default selection. If Allow Custom Text Entry is selected on the Field tab, you can type a default selection that is not one of the list items.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input and be compatible with the data format selected on the Binding tab. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only, or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protect, Calculated - Read Only, or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *FormValidation (Form Properties dialog box)*.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using drop-down lists and list boxes

Drop-down list properties in the Binding tab

When you create a drop-down list, the Binding tab presents data binding options that you can apply to the list. Options that are not specifically related to creating a data connection apply both to a data source and to data saved to a file when the object is not bound to a data source.

Name

Sets the name of the list. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Specify Item Values

Lets you specify custom data values for each list item. If this option is not selected, the data values will match the text for the list items. If this option is selected, the default values will be sequential integers, starting at "1" for the first list item. See *To specify list item values for a drop-down list or list box*.

NOTE: Specify Item Values is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command from the Object palette menu.

Up, Down, Sort Ascending, Sort Descending

Reorders the items in the list (for example, if the data should be stored in a different order compared to the display order of options in the drop-down list). You can move list items up or down, or sort them ascending or descending using the buttons beside the label.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using drop-down lists and list boxes

12.9. Email submit button

Email submit button properties in the Field tab

When you select an email submit button, the Field tab in the Object palette displays several options for formatting email submit buttons.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style:

No Border

Removes the line around the button.

Solid Border

Creates a thick line around the button.

Raised Border

Creates a shadow around the button so that the button looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the button.

Highlighting

Sets the button highlight when the button is clicked:

None

Removes the button highlight.

Inverted

Inverts the button highlight when the button is clicked.

Push

Creates a shadow around the button so that the button looks like it is recessed.

Outline

Creates a line around the button when the button is clicked.

Rollover Caption

Sets a rollover caption when the button highlight is set to Push. This caption appears when the mouse pointer moves over the button.

Down Caption

Sets a down caption when the button highlight is set to Push. This caption appears when the button is clicked.

Email Address

Sets the recipient of the email message and the attached form data.

Email Subject

Sets the subject line for the email message.

Submit As

Sets the format of the data to submit.

PDF

Submits a package containing an embedded PDF file. Choose this format if the form contains a signature field, or if a copy of the form along with its data needs to be saved by Forms, or submitted to another type of target server. Do not choose this option if the form initiates server-side processing, if Forms will be used to render at run time HTML forms or forms whose layout adjusts, or if the form is a PDF form that will be filled in Adobe Reader without the use of Acrobat Reader DC extensions.

XML Data (XML)

Submits an XML data stream, which allows for the hierarchical representation of data and can be parsed by any generic XML parser. Choose this format if the server that communicates with the run-time user application program must receive an XML data steam.

URL-Encoded Data (HTTP Post)

Submits a text stream to the specified Uniform Resource Locator (URL) using the `POST` method. The text stream can be parsed by an FTP server, a mail server, or a web server. To use this method, users must open the form in Adobe Reader 6.0 or later or in a web browser unless the URL specifies the mailto protocol.

Sign Submission

Applies a data signature to the submitted data. When a form filler clicks the button, a digital signature is created to cover the submitted data and attachments. Data signatures secure the signed data and guarantee the data integrity during transmission. Data signatures can apply to the form data or to the entire submission, including attachments. Click the Settings button to define optional security properties for the data signature, such as the signature handler, signing certificates, and certificate issuers.

Encrypt Submission

Encrypts form content. When a form filler clicks the button, the form content is encrypted before submission to secure the form content during transmission. You apply encryption to the form data or to the entire submission, including attachments. Click the Settings button to define optional encryption properties such as encryption algorithm, encryption certificate, certificate issuers, and key usage.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using buttons

HTTP submit button

12.10. Extended Decimal Field

Extended Decimal Field properties in the Field tab

When you select an extended decimal field, the Field tab in the Object palette presents options for extended decimal fields. All but two of the options are the same as the options for numeric fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Limit Length to Visible Area

Sets the maximum amount of numbers allowed in the field according to the horizontal length of the numeric field.

Comb of <x> characters

Enables the comb format.

Patterns

Sets the pattern for displaying formatted values in a form, for syntax of the user input, for validating user input, and for storing and retrieving bound data or saving data when the form is not bound to a data source. See *Patterns dialog box*.

Limit Leading Digits and Max

Sets the maximum number of digits that can appear before the decimal separator. If the maximum is exceeded, a zero displays.

Default: Unlimited

Maximum: 31 (including trailing digits).

Limit Trailing Digits and Max

Sets the maximum number of digits that can appear after the decimal separator. If the maximum is exceeded, the number is truncated. The numbers are rounded using ROUND_HALF_UP rounding. For example, when a user specifies lead digits and trail digits as 2, value1.244xxxxx is rounded to 1.24 and value 12.999xxxxx is rounded to 13.0.

By default, this option is deselected and the precision of decimal data is reflected exactly as it is entered.

Maximum: 14

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

12.11. (Deprecated) Flash field

NOTE: The Flash field is not available in Designer 6.1. Your existing forms containing Flash field continue to work but you cannot use Designer 6.1 to add a Flash field or change the properties of an existing Flash field.

(Deprecated)Flash field properties in the Field tab

Caption

Sets a caption for the object.

URL

Specifies the URL of the Flash source file (SWF) to be displayed in the currently selected Flash field. Enter the location of the file or click the browse button to select the file.

Embed Flash Data

Stores flash data in the form.

Poster

Specifies the URL of the image file to display in the currently selected Flash field, when the Flash file (SWF) is not displayed.

Embed Poster Image

Stores the image file in the form.

Property/Value

Activation

Explicit

Flash content is activated with a user action or script.

Page Current

Flash content is activated when the page that the Flash field is located on is the current page.

Page Visible

Flash content is activated when the page that the Flash field is located on is visible.

Deactivation

Explicit

Flash content is deactivated with a user action or script.

Page Current

Flash content is deactivated when the page that the Flash field is located on is no longer the current page.

Page Invisible

Flash content is deactivated when the page that the Flash field is located on is no longer visible.

Pass Context Click

A flag indicating whether a context-click on the Flash content should be passed to the Flash player runtime or should be handled by the viewer application. A context-click is usually generated by a mouse right-click, but may be invoked by other means. This can include, but is not limited to, an explicit context-menu keyboard key or the combination of a mouse click and a keyboard modifier key.

Yes

Indicates that the context menu in the viewer application will not be visible, and the user will see the context menu and any custom items generated by the Flash player runtime.

No

the viewer application handles the context click.

Display Flash Content in Floating Window

enables floating window property and value settings.

Property/Value

Default Width

Specifies the default width of the floating window.

Max Width

Specifies the maximum width of the floating window.

Min Width

Specifies the minimum width of the floating window.

Default Height

Specifies the default height of the floating window.

Max Height

Specifies the maximum height of the floating window.

Min Height

Specifies the minimum height of the floating window.

Horizontal Alignment

Specifies the horizontal alignment of the Flash content window. Horizontal alignment determines how horizontal offset is applied. Options include Near, Center, or Far.

Vertical Alignment

Specifies the vertical alignment of the Flash content window. The vertical alignment determines how the vertical offset is applied. Options include Near, Center, or Far.

Horizontal Offset

Specifies the offset from the alignment point specified by the Horizontal offset. A positive value for Near and Center alignments produces an offset towards the Far direction. A positive value for Far alignment produces an offset towards the Near direction.

Vertical Offset

The offset from the alignment point specified by the Vertical Alignment. A positive value for Near and Center alignments produces an offset towards the Far direction. A positive value for Far alignment produces an offset towards the Near direction.

Edit

Opens the Additional Assets dialog box so you can add, remove, and embed assets like video, sound, image, text, XML, and SWC files.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

NOTE: If you select the Hidden (Exclude from Layout) option, and apply a script to make the Flash Field visible and enabled when a form filler clicks the field, form fillers must click the field twice to initiate both changes.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Flash field properties in the Binding tab

Using flash fields

Flash field properties in the Binding tab

Name

Sets the Flash field name. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name. (See *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard. (see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.)

No Data Binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings (Execute WSDL)

Sets an import/export binding for a WSDL data connection. (See *To create a data connection using a WSDL file*.)

RELATED LINKS:

Working with Data Sources

Flash field properties in the Field tab

Using flash fields

12.12. HTTP submit button

HTTP submit button properties in the Field tab

When you select an HTTP submit button, the Field tab in the Object palette displays several options for formatting HTTP submit buttons.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style:

No Border

Removes the line around the button.

Solid Border

Creates a thick line around the button.

Raised Border

Creates a shadow around the button so that the button looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the button.

Highlighting

Sets the button highlight when the button is clicked:

None

Removes the button highlight.

Inverted

Inverts the button highlight when the button is clicked.

Push

Creates a shadow around the button so that the button looks like it is recessed.

Outline

Creates a line around the button when the button is clicked.

Rollover Caption

Sets a rollover caption when the button highlight is set to Push. This caption appears when the mouse pointer moves over the button.

Down Caption

Sets a down caption when the button highlight is set to Push. This caption appears when the button is clicked.

URL

The URL that the form data will be posted to.

Sign Submission

Applies a data signature to the submitted data. When a form filler clicks the button, a digital signature is created to cover the submitted data and attachments. Data signatures secure the signed data and guarantee the data integrity during transmission. Data signatures can apply to the form

data or to the entire submission, including attachments. Click the Settings button to define optional security properties for the data signature, such as the signature handler, signing certificates, and certificate issuers.

Encrypt Submission

Encrypts form content. When a form filler clicks the button, the form content is encrypted before submission to secure the form content during transmission. You apply encryption to the form data or to the entire submission, including attachments. Click the Settings button to define optional encryption properties such as encryption algorithm, encryption certificate, certificate issuers, and key usage.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

HTTP submit buttons

Using buttons

12.13. Image field

Image field properties in the Field tab

When you select an image field, the Field tab displays several options specific to formatting image fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

URL

Sets the location of the source image file. To employ relative path names for retrieving linked images when the form is opened, the image files must be stored in a folder that is accessible to users. If Forms is available, the path must be relative to Forms.

Embed Image Data

When selected, a copy of the image information is stored in the form. When deselected, the image data is stored separately from the form and the image is resolved when the form is opened.

If the image field will be used to load images dynamically when the form is rendered, do not select the Embed Image Data option.

Sizing

Enables or disables image resizing when the image is loaded.

Scale Image Proportionally

Resizes the image in the object, ensuring that the aspect ratio of the image is preserved.

Scale Image to Fit Rectangle

Resizes the image to match the dimensions of the object. The aspect ratio of the image is not preserved.

Use Original Size

Does not resize the image. The object is resized to preserve the actual size of the image.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using image fields

Image field properties in the Binding tab

When you create an image field, the Binding tab presents data binding options that you can apply to the object. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the image field name. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using image fields

12.14. Image

Image properties in the Draw tab

When you create an image object, the Draw tab in the Object palette presents options specific to formatting images.

Type

Sets the type of object. Objects are the building blocks of every form.

URL

Sets the location of the source image file. To employ relative path names for retrieving linked images when the form is opened, the image files must be stored in a folder that is accessible to users. If Forms is available, the path must be relative to Forms.

Embed Image Data

When selected, a copy of the image information is stored in the form. When deselected, the image data is stored separately from the form and the image is resolved when the form is opened.

Sizing

Enables or disables image resizing when the image is loaded:

Scale Image Proportionally

Resizes the image in the object, ensuring that the aspect ratio of the image is preserved.

Scale Image to Fit Rectangle

Resizes the image to match the dimensions of the object. The aspect ratio of the image is not preserved.

Use Original Size

Does not resize the image. The object is resized to preserve the actual size of the image.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using images

12.15. Line

Line properties in the Draw tab

When you create a line, the Draw tab in the Object palette presents options specific to formatting lines.

Type

Sets the type of object. Objects are the building blocks of every form.

Appearance

Sets the slope of the line.

Line Style

Sets the line style and thickness.

Color Picker

Sets the line color.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using circles, lines, and rectangles

12.16. List box

List box properties in the Field tab

When you select a list box, the Field tab displays several options specific to formatting list boxes.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

List Items

Sets the items in the list and their positions. You can add or remove list items, paste items, move them up or down, or sort the items in ascending or descending order using the buttons beside the label.

NOTE: List Items is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu.

Allow Multiple Selection

Enables users to select more than one option in the list (run-time).

NOTE: If you add a List Box object with the Allow Multiple Selection option selected on the Field tab to a paper forms barcode collection, you must also select the Enforce Strict Scoping Rules in JavaScript option on the Defaults tab in the Form Properties dialog. Otherwise, any values that a form filler selects in the List Box object may not encode properly in the paper forms barcode.

Commit On

Commits the selected option when the user exits the List box, moving the focus to another object.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using drop-down lists and list boxes

List box properties in the Value tab

When you create a list box, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets a default selection.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input and be compatible with the data format selected on the Binding tab. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only, or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: This option is not available when the Type option is set to Protect, Calculated - Read Only, or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using drop-down lists and list boxes

List box properties in the Binding tab

When you create a list box, the Binding tab presents data binding options that you can apply to the list. Options that are not specifically related to creating a data connection apply both to a data source and to data saved to a file when the object is not bound to a data source.

Name

Sets the name of the list. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Specify Item Values

Lets you specify custom data values for each list item. If this option is not selected, the data values will match the text for the list items. If this option is selected, the default values will be sequential integers, starting at "1" for the first list item. See *To specify list item values for a drop-down list or list box*.

NOTE: Specify Item Values is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command from the Object palette menu.

Up, Down, Sort Ascending, Sort Descending

Reorders the items in the list (for example, if the data should be stored in a different order compared to the display order of options in the drop-down list). You can move list items up or down, or sort them ascending or descending using the buttons beside the label.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using drop-down lists and list boxes

12.17. Numeric field

Numeric field properties in the Field tab

When you select a numeric field, the Field tab in the Object palette displays several options for formatting fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Limit Length to Visible Area

Sets the maximum amount of numbers allowed in the field according to the horizontal length of the numeric field.

Comb of <x> characters

Enables the comb format.

Patterns

Sets the pattern for displaying formatted values in a form, for syntax of the user input, for validating user input, and for storing and retrieving bound data or saving data when the form is not bound to a data source. See *Patterns dialog box*.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

NOTE: If you select the Hidden (Exclude from Layout) option, and apply a script to the Flash Field object to make it visible and activated when a form filler clicks the field, form fillers must click the field twice, once for each event.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Overview of numeric fields

Using decimal and numeric fields

Numeric field properties in the Value tab

When you create a numeric field, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets an initial value for the field. The value is formatted according to the setting specified using the Patterns option on the Field tab, but must conform to the value according to the locale specified for the field. If the data is bound and a data pattern has been specified, the value must match the data pattern specified in the Binding tab. See *To specify a default value*.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input and be compatible with the data format selected on the Binding tab. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: This option is not available when the Type option is set to Protect, Calculated - Read Only, or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Opens the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Calculation Script

The value will be calculated by a script written for the object's `calculate` event.

NOTE: This option is available only when the Type option is set to Calculated - Read Only.

Runtime Property

Sets one of the following run-time properties for the object. When you select a run-time property, the actual value corresponding to the option is inserted dynamically when the form is rendered.

Current Page Number

Displays the number of the current page in the numeric field.

Number of Pages

Inserts the total number of pages making up the form.

Current Sheet Number

Inserts the number of the current piece of paper (sheet) in the numeric field.

Number of Sheets

Inserts the total number of pieces of paper (sheets) that make up the form.

NOTE: This option is available only when the Type option is set to Calculated - Read Only.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Overview of numeric fields

Using decimal and numeric fields

Numeric field properties in the Binding tab

When you create a numeric field, the Binding tab presents data binding options that you can apply to the object. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the numeric field. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Data Format

Sets the format of the data. See *To specify a data pattern*:

Float

A three-part representation of a number that contains a radix character. The number is represented by the following parts: a sign, a number, and a radix character. Some examples of floating point numbers are 4.23423412, 1234.1234234, or 4.00. Note that in some locales, the radix character is not a decimal point. For example, in a German (Germany) locale, the radix character is a comma (as in 1234,1234).

Integer

Any sequence of the digits 0 through 9 possibly preceded by a minus sign. For example, the number may be a positive (such as 1, 2, or 3), a zero, or a negative whole number (such as -1, -2, or -3).

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Overview of numeric fields

Using decimal and numeric fields

Numeric patterns

Specific symbols must be used to create numeric patterns for numeric fields. You can also use these symbols to create numeric patterns for decimal fields.

NOTE: The dash (-), colon (:), slash (/) and space ( ) are treated as literal values and can be included anywhere in a pattern. To include a phrase in a pattern, delimit the text string with single quotation marks ('). For example, `'Your balance is' z,zz9.99` can be specified as the Display Pattern.

Use these pattern symbols to create patterns. The pattern determines the format for users to enter or format data. The formatted results in the Example column are delimited by double quotation marks so that you can see where spaces would appear in the result. The quotation marks are not part of the result. In these examples, the locale is English (USA).

Numeric symbol	Description	Example
9	Single digit or the zero digit if the input is empty. When following the decimal radix, indicates the exact number of digits as contained in the data value.	Display Pattern: zzz,zz9.999 Input Value: 123456.1236 or 1234 Formatted Value: “123,456.124” or “1,234.000”
z	Single digit or no output if the input is the leading zero digit, or no output if the input is empty	Display Pattern: $zzz9 Input Value: 123 Formatted Value: “$123”
z	Single digit or a space if the input is the leading zero digit, or a space if the input is empty	Display Pattern: $ZZZ9 Input Value: 123 Formatted Value: “$ 123”
s	Minus sign (-) if the number is negative (when the number is positive, no symbol is required)	Display Pattern: $szzz9 Input Value: 123 or -123 Formatted Value: “$123” or “$-123”
S	Minus sign (-) if the number is negative, or a space if the number is positive	Display Pattern: $Szzz9 Input Value: 123 or -123 Formatted Value: “$ 123” or “$-123”

Numeric symbol	Description	Example
E	Exponential symbol (E+3 means exponent value is 3 and E-2 means exponent value is -2). The exponential symbol (E) may be used after a 9, z, or Z digit only.	Display Pattern: 99.999E Input Value: 12345 or 0.12345 Formatted Value: “12.345E+3” or “12.345E-2”
$	Currency symbol, whichever one corresponds to the object’s current Locale setting	Display Pattern: $zzz9 Input Value: 123 Formatted Value: “$123”
cr	Credit symbol (CR) if the number is negative (when the number is positive, no symbol is required)	Display Pattern: cr$zzz9 Input Value: 123 or -123 Formatted Value: “$123” or “CR$123”
CR	Credit symbol (CR) if the number is negative, or spaces if the number is positive	Display Pattern: CR$zzz9 Input Value: 123 or -123 Formatted Value: “ $123” or “CR$123”
db	Debit symbol (DB) if the number is negative (when the number is positive, no symbol is required)	Display Pattern: db$zzz9 Input Value: 123 or -123 Formatted Value: “$123” or “DB$123”
DB	Debit symbol (DB) if the number is negative, or spaces if the number is positive	Display Pattern: DB$zzz9 Input Value: 123 or -123 Formatted Value: “ $123” or “DB$123”
( )	Left and right parenthetic characters to show a negative number. If the number is a positive number, the parenthetic characters become spaces. ***NOTE:*** *The left and right parenthetic characters may enclose an 8, 9, z, or Z digit only.*	Display Pattern: ($zzz9) Input Value: 123 or -123 Formatted Value: “ $123 ” or “($123)”
.	Decimal radix, whichever one corresponds to the object’s current Locale setting	Display Pattern: zzz9.99 Input Value: 123.45 Formatted Value for a German (Germany) locale: “123,45”
v	Implied decimal radix, whichever one corresponds to the object’s current Locale setting (the decimal radix is stripped out before the output is written)	Display Pattern: zzz9v99 Input Value: 123.45 Formatted Value: “12345”

Numeric symbol	Description	Example
,	Grouping character, whichever one corresponds to the object's current Locale setting	Display Pattern: z,zz9.99 Input Value: 1234.56 Formatted Value for a German (Germany) locale: "1.234,56"
%	Percentage symbol, whichever one corresponds to the object's current Locale setting	Display Pattern: z9.99% Input Value: 0.123 Formatted Value: "12.30%"

Reserved symbols

The following symbols have special meanings and cannot be used as literal text.

Symbol	Description
?	When submitted, the symbol matches any one character. When merged for display, it becomes a space.
*	When submitted, the symbol matches 0 or Unicode white space characters. When merged for display, it becomes a space.
+	When submitted, the symbol matches one or more Unicode white space characters. When merged for display, it becomes a space.

RELATED LINKS:

Formatting field values and using patterns

To specify a data pattern

To specify an edit pattern

Examples of numeric patterns

Examples of numeric patterns

The following table shows some example display patterns and the results of formatting some example input values (user input or bound data) using the display pattern. The formatted results are delimited by double quotation marks so that you can see where spaces would appear in the result. The quotation marks are not part of the result. In these examples, the locale is English (USA).

Pattern	Input value	Formatted value
zzz,zz9.99	123456.1236 123456.4 123456	"123,456.12" "123,456.40" "123,456.00"

Pattern	Input value	Formatted value
$Z,ZZ9.99	123.4	“ 123.40”
$z,zz9.99CR	1234 -1234	“$1,234.00” “$1,234.00CR”
$Z,ZZ9.99DB	123 -123	“$ 123.00” “$ 123.00DB”
($z,zz9.99)	1234 -1234	“ $1,234.56 “ “($1,234.00)”
zzz9	12.0	“12”
Szzz,zz9.99	1234.56 -1234.56	“1,234.56” “-1,234.56”
$9,999.99	123.4	“$0,123.40”
$Z,999.99	123.4	“$ 123.40”
99999	1234	“01234”
S999v99	-1.23	"-00123"
S999V99	-1.23 123	"-001.23” " 123.00"
SZZ9.99	12.3 -12.3	" 12.30" "- 12.30"
szz9.99	12.3 -12.3	“12.3” “-12.3”
99.999E	12345 0.12345	"12.345E+3" "12.345E-2”

The table below shows the results of converting some example input values (user input or bound data) into raw values using an edit pattern or data pattern.

Custom pattern	Input value	Raw value
99V99	1050 3125	10.50 31.25
99.999E	12.345E3 12.345E-2	12345 0.12345
z999	150 0150	150 150

Custom pattern	Input value	Raw value
z,zz9.99	10.50 3,125.00	10.50 3125.00
$z,zz9.99DB	$1,234.00 $1,234.00DB	1234.00 -1234.00

12.18. Paper Forms Barcode

Paper forms barcode properties in the Field tab

When you select a paper forms barcode object on the form design, the Field tab in the Object palette displays several options specific to formatting paper forms barcodes.

Type

Sets the type of object. Objects are the building blocks of every form.

Label

A custom or automatically generated caption that appears below the barcode. Useful for distinguishing barcodes in distinct versions of barcoded forms.

Generate Label Automatically

When selected, the barcode label is automatically generated. Deselect the option to specify a custom label.

Symbology

Specifies the symbology for the barcode:

- PDF417
- QR Code
- Data Matrix

  NOTE: End users filling a form that contains a QR Code or DataMatrix barcode require Acrobat 7.0.5 or later, or Adobe Reader 7.0.5 or later. Decoding of QR Code and DataMatrix barcodes is supported by barcoded forms 7.0 but is not supported by barcoded forms 7.0 Standalone (ST).

Scanning Method

Specifies the conditions for decoding the information in the paper forms barcode after it is printed on a form. The decode condition determines the horizontal and vertical data cell size and the error correction level for the barcode. Designer contains the following preset and custom options:

Document Scanner (Default)

Selecting this option automatically sets the Error Correction Level, Module Height/Width, and Module Width to values that work well when scanning a paper forms barcode using a 300-dpi scanner. Designer specifies these values:

- Error Correction Level: 5
- Module Height/Width: 2 (PDF417), 1 (QR Code and Data Matrix)
- Module Width: 0.0133 inches

Hand-held Scanner

Selecting this option automatically sets the Error Correction Level, Module Height/Width, and Module Width to values that work well when scanning a paper forms barcode using a hand-held scanner. Designer specifies these values:

- Error Correction Level: 5
- Module Height/Width: 3 (PDF417), 1 (QR Code and Data Matrix)
- Module Width: 0.0133 inches

Fax Server

Selecting this option automatically sets the Error Correction Level, Module Height/Width, and Module Width to values that work well when scanning a paper forms barcode that was printed on a form received by a fax machine. Designer specifies these values:

- Error Correction Level: 6
- Module Height/Width: 2 (PDF417), 1 (QR Code and Data Matrix)
- Module Width: 0.0133 inches

Custom

Selecting this option activates the Property/Value list and lets you set the properties for the Error Correction Level, Module Height/Width, and Module Width to any values within the following ranges:

- Error Correction Level: 1 - 8
- Module Height/Width: 1 - 4 (whole numbers for PDF417), 1 (QR Code and Data Matrix)
- Module Width: 0.0133 - 0.04 inches
- Columns/Rows: Specify number of columns or rows (Data Matrix)

When Custom is selected, the Property list remains activated, regardless of the Error Correction Level, Module Height/Width, and Module Width property values, even if they are identical to those of a preset condition.

NOTE: The Error Correction Level setting is disabled for Data Matrix barcodes.

Property/Value

When the Custom option is selected, you must set the property values for the error correction level, module height-over-width ratio, and module width:

Error Correction Level

Specifies the level of data redundancy that is added to the barcode to help correct for possible scanning errors. Acceptable levels for Value range from 1 to 8. Higher levels allow for more redundancy and a more robust barcode. Higher levels will also require a larger barcode. A more robust barcode can recover from errors, such as pen marks or folds in the document, by including redundant information in the barcode. Designer will automatically set an Error Correction Level based on the type of scanning device you select.

An Error Correction Level of 5 tends to provide good performance and still allow the barcode to be a reasonable size. It is recommended that you keep the Error Correction Level above 5 unless you are very familiar with barcode decoding technology.

NOTE: The Error Correction Level setting is disabled for Data Matrix barcodes.

Module Height/Width

Sets the height of the barcode, as a ratio of the module height over width. Acceptable levels for Value are integers (whole numbers) from 1 to 4 (PDF417), 1 (QR Code and Data Matrix).

Module Width

Sets the width of the smallest bar in the barcode. Every other bar in the barcode will be a multiple of this dimension. Acceptable levels for Value range between 0.0133 inches and 0.04 inches.

The Module Width value will be rounded to the nearest 1/300 inch when the barcode is rendered in Acrobat or Adobe Reader. You should be very familiar with barcode decoding technology when selecting the module width.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using paper forms barcodes

Paper forms barcode properties in the Value tab

When you select a paper forms barcode object on the form design, the Value tab of the Object palette presents options specific to formatting paper forms barcodes.

Compress Data Before Encoding

Indicates whether the data is compressed before it is encoded by the paper forms barcode object. The default setting is to not compress the data.

The data compression uses the Flat method. When this option is selected, compression is performed inside Acrobat or Adobe Reader when the barcode is drawn.

Compressed data requires less storage space in the barcode, allowing more form data to be stored in the barcode.

NOTE: Typically, you will want data compressed unless the data will be decoded with a handheld scanner or your own decode solution that does not include decompression software. For more information about compressing data, see the Acrobat Help.

Automatic Scripting

Specifies whether the barcode data is obtained automatically or through custom scripting.

Format

Specifies the format of the barcode data that is automatically scripted. When an option is selected, the associated script appears in the Script Editor. If required, you can edit the predefined scripts. Select from the following formats:

XML

The value of the barcode is obtained through a script that encodes the data in XML format.

Delimited

The value of the barcode is obtained through a script that encodes the data in a delimited format (a flat ASCII representation of field values separated by the specified delimiter character).

Include Field Names

Separates the field names using the specified delimiter and encodes them. Field names are listed first, followed by the field values. This option is available when the Delimited option is selected in the Format list.

Include Label

Encodes the label (shown in Field tab) as the first delimited field in the barcode contents. This option is available when the Delimited option is selected in the Format list.

Use Legacy Format

Uses a legacy encoding format to produce XML barcode content. The legacy encoding format is used in paper forms barcode XML scripts in Designer 7.0 to 8.0. This option is selected automatically when you open a form that uses the legacy encoding format in Designer. By default, Designer uses a new encoding format.

The paper forms barcode object supports two XML encoding formats produced by two XFA JavaScript functions, `xfa.datasets.saveXML()` and `xfa.record.saveXML()`. The legacy encoding format uses the function `xfa.datasets.saveXML()` and the new encoding format uses the function `xfa.record.saveXML()` to produce XML encoded data. The legacy encoding format produces longer XML data output than the new encoding format. The new encoding format produces more compact XML data output that provides more barcode space and is better suited for XML encoding scripts.

Delimiter

Specifies the delimiter character that is used to separate field names, field values, and barcode labels in the encoded data. This option is available when the Delimited option is selected in the Format list.

NOTE: It is recommended that you do not select the Carriage Return option as a delimiter, because the carriage return character is already used in the paper forms barcode values as a delimiter to separate the header and value rows. See *To specify a delimiter*.

Character Encoding

Specifies the character set used for barcode content. Select one of these options.

UTF-8

Characters are encoded using Unicode code points as defined by Unicode-3.2, and UTF-8 serialization as defined by ISO/IEC 10646. There is no byte order mark. This option is selected by default.

UTF-16

Characters are encoded using Unicode code points as defined by Unicode-3.2, and UTF-16 serialization as defined by ISO/IEC 10646 and ISO-10646. There is no byte order mark.

UCS-2

Characters are encoded using Unicode code points as defined by Unicode 3.2, and UCS-2 serialization as defined by ISO/IEC 10646. There is no byte order mark.

ISO-8859-1

Characters are encoded using ISO-8859-1, also known as Latin-1.

ISO-8859-2

Characters are encoded using ISO-8859-2.

ISO-8859-7

Characters are encoded using ISO-8859-7.

Shift_JIS

Characters are encoded using JIS X 0208, also known as Shift_JIS.

KSC_5601

Characters are encoded using the Code for Information Interchange (Hangul and Hanja).

GB-2312

Characters are encoded using Simplified Chinese.

Big Five

Characters are encoded using Traditional Chinese (Big-Five). Note that there is no official standard for Big-Five and several variants are in use.

Font Specific

Characters are encoded in a font-specific way. Each character is represented by one 8-bit byte.

None

No special encoding is specified. Characters are encoded using the ambient encoding for the operating system.

NOTE: Keep in mind that some scanning devices do not support certain character sets. Also, Acrobat and Adobe Reader 7.0. and earlier do not support character encoding.

Apply To

Specifies what the selected format applies to. Select from the following list:

Entire Form Data

Encodes the data from all form fields in the selected format

Collection Data

Encodes the data from all fields specified in the selected collection.

Collection

Specifies the collection to use to encode the data. Select from the following list:

Collection name

The name of the collection that you created.

New/Manage Collection

Creates, deletes, duplicates, renames, and modifies collections in the Collection list.

NOTE: If you add a List Box object with the Allow Multiple Selection option selected on the Field tab to a paper forms barcode collection, you must also select the Enforce Strict Scoping Rules in JavaScript option on the Defaults tab in the Form Properties dialog. Otherwise, any values that a form filler selects in the List Box object may not encode properly in the paper forms barcode.

Modify

Displays the selected fields for the current collection. You can also edit the fields in the collection and rename the collection in this window.

RELATED LINKS:

Using paper forms barcodes

To control how a paper forms barcode encodes data

Design tips for using a paper forms barcode

12.19. Password field

Password field properties in the Field tab

When you select a password field, the Field tab in the Object palette displays several options for formatting password fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Password Display Character

Sets the display character.

Limit Length to Visible Area

Sets the maximum number of characters allowed in the field according to the horizontal length of the password field.

Patterns

Sets the pattern for the syntax of the user input, for validating user input, and for storing and retrieving bound data or for saving data when the form is not bound to a data source. See *Patterns dialog box*.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using password fields

Password field properties in the Value tab

When you create a password field, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input and be compatible with the data format selected on the Binding tab. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only, or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic

Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: This option is not available when the Type option is set to Protect, Calculated - Read Only, or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

RELATED LINKS:

Using password fields

Password field properties in the Binding tab

When you create a password field, the Binding tab presents data binding options that you can apply to the object. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the field. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using password fields

Password patterns

If you choose, you can define password patterns for fields. Note however, that defining a specific pattern may make passwords more vulnerable to attack as it reduces the possible character combinations that may be used. If you want to define a pattern, use these symbols.

NOTE: The comma (,), dash (-), colon (:), slash (/), period (.), and space ( ) are treated as literal values and can be included anywhere in a pattern.

Password symbols	Description
A	Single alphabetic character
X	Single character
O	Single alphanumeric character
0	Single alphanumeric character
9	Single digit

NOTE: The O (capital o) and 0 (zero) symbols are intentional synonyms in recognition that these two characters are often mistyped and misread.

Reserved symbols

The following symbols have special meanings and cannot be used in a password.

Symbol	Description
?	When submitted, the symbol matches any one character. When merged for display, it becomes a space.
*	When submitted, the symbol matches 0 or Unicode white space characters. When merged for display, it becomes a space.

Symbol	Description
+	When submitted, the symbol matches one or more Unicode white space characters. When merged for display, it becomes a space.

RELATED LINKS:

Formatting field values and using patterns

Examples of password patterns

Examples of password patterns

The following table shows some example edit patterns that you can define for processing user-entered passwords.

Custom pattern	Input value
AAA999	MAB123
XXX-9	MA2-4
99999	12345
9009AA9	1M23BE4 1m23be4 1MA3BE4

12.20. Radio button

Radio button properties in the Field tab

When you select a radio button or exclusion group, the Field tab in the Object palette displays several options for formatting radio buttons.

Type

Sets the type of object. Objects are the building blocks of every form.

Appearance

Sets a border style for the button.

If a single radio button is selected, these options are available:

None

Does not display a border around the button.

Solid Square

Uses a solid square to represent the button.

Sunken Square

Creates a square shadow for the button so that the button looks three -dimensional.

Solid Circle

Uses a solid circle to represent the button.

Sunken Circle

Creates a circular shadow for the button so that the button looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

If the exclusion group is selected, these options are available:

No Border

Does not display a line around the group.

Solid Border

Creates a thick line around the group.

Raised Border

Creates a shadow around the group so that it looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option to define a custom look for the group.

Size

Sets the size of the radio button.

Check Style

Sets the radio button style:

Default

Uses a filled circle for the button style.

Check

Uses a check mark for the button style.

Circle

Uses a circle for the button style.

Cross

Uses a cross for the button style.

Diamond

Uses a diamond for the button style.

Square

Uses a square for the button style.

Star

Uses a star as the button style.

Item

Lists all of the radio buttons in the current exclusion group. You can double-click a radio button name to edit its caption.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using radio buttons

Radio button properties in the Value tab

When you create an exclusion group of radio buttons, the Value tab displays several options that you can apply to the entire group.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user

never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets a default selection.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protect, Calculated - Read Only, or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using radio buttons

Radio button properties in the Binding tab

When you create a radio button exclusion group, the Binding tab presents data binding options that you can apply to the group. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the exclusion group.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Specify Item Values

Lets you specify custom On values for each radio button in the exclusion group. The value of each radio button in the group must be unique compared to the values of all other radio buttons in the group.

When this option is not selected, the On values will match the caption for the radio button. When this option is selected, the default values are sequential integers starting at “1” for the first radio button in the group. See *To assign On values to radio buttons*.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using radio buttons

12.21. Rectangle

Rectangle properties in the Draw tab

When you create a rectangle, the Draw tab in the Object palette presents options specific to formatting rectangles.

Type

Sets the type of object. Objects are the building blocks of every form.

Line Style

Sets the line style and thickness.

Color Picker

Sets the line color.

Corners

Sets the corner style.

Radius

Sets the radius of notched corners.

Fill

Sets a fill style.

Color Picker

Sets fill colors.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using circles, lines, and rectangles

12.22. Signature field

Signature field properties in the Field tab

When you select a signature field, the Field tab in the Object palette displays several options for formatting signature fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using signature fields

Using digital signatures

Signature field properties in the Signature tab

When you select a signature field object, the Signature tab in the Object palette displays various options that control which fields (form objects) you can lock after the signature field is signed. The object types you can lock are buttons, check boxes, date/time fields, decimal fields, drop-down lists, image fields, list boxes, numeric fields, paper forms barcodes, password fields, radio buttons, other signature fields, and text fields.

You can also click the Settings button to define optional security properties for the signature, such as the signature handler, signing certificates, and certificate issuers.

Lock Fields After Signing

Specifies which fields (objects) the signature applies to. The signature field can apply to all of the fields in the document, all of fields inside a collection of selected objects, or all of the fields outside of a collection of selected objects. After the signing party signs the signature field, the fields in the specified area are locked. This option is selected be default when you add a signature field to a form design. If you deselect this option, none of the fields in the form will be locked after the signing party signs the signature field.

You can select one of options in the drop-down list to indicate which fields to lock or to manage collections.

All Fields in Document

Locks all the fields in the document.

Collection - <name>

Locks all fields in the selected collection.

New/Manage Collection

Opens the Collection List dialog box, where you can create, modify, duplicate, delete, and sort object collections.

All Fields in Collection

Locks all of the fields that are included in the selected collection.

All Fields not in Collection

Locks all of the fields that are not included in the selected collection.

RELATED LINKS:

Using signature fields

Using digital signatures

Using form object collections

12.23. Subform

Subform properties in the Subform tab

When you select a subform, the Subform tab in the Object palette displays several formatting options.

Source File

Sets the source file for the fragment reference. This property is visible only when the selected object is a fragment reference.

Fragment Name

Sets the name of the fragment. You can click the Frag Info button to view the fragment information.

This property is visible when a fragment reference or a fragment that is defined in a source file is selected. When the selected object is a fragment reference, this property does not appear if the source file is not specified. The Fragment Name list includes all the fragments in the specified source file. The Custom option directly supports setting a SOM expression or an ID value as the fragment reference and supports the implementation in the XML Forms Architecture.

Content

Sets positioned or flowed subform content.

Positioned

Positions the objects in the subform according to their individual X and Y coordinates.

Flowed

Positions the objects in the subform during the data-merging process so that no objects are rendered on top of each other.

Flow Direction

Sets a flow direction for flowed objects:

Top to Bottom

Lays down the objects one beneath the other.

Western Text

Lays down the objects from left to right, and when the objects wrap at the right edge of the subform, continue laying down objects one object down from left to right.

Right to Left

Sets the default tabbing order and data-fill order starting from the right of the page and moving to the left of the page.

NOTE: This option is unavailable unless the Flowed option is selected in the Content list.

Allow Page Breaks Within Content

Enables or disables page breaks in the middle of a subform.

This Subform Is An Insertion Point

Defines the selected subform as an insertion point. See *Creating an insertionpoint*.

Name

Name of the insertion point subform.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

About subforms

To specify a subform to span multiple pages

Subform properties in the Pagination tab

When you select a subform, the Pagination tab in the Object palette displays several pagination options.

Place

Determines where the subform will be placed.

Following Previous

Places the subform after the previous object in the parent subform.

In Content Area > [name_of_content_area]

Places the subform in the specified content area.

Top of Next Content Area

Places the subform at the top of the next content area.

Top of Content Area > [name_of_content_area]

Places the subform at the top of the specified content area.

On Page > [name_of_page]

Places the subform on the specified page (introduces a page break if one does not occur naturally).

Top of Next Page

Places the subform at the top of the next page (introduces a page break if one does not occur naturally).

Top of Page > [name_of_page]

Places the subform at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Odd Page

Places the subform on pages that are set to print on odd pages.

Top of Next Odd Page

Places the subform at the top of the odd page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Even Page

Places the subform on pages that are set to print on even pages.

Top of Next Even Page

Places the subform at the top of the even page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

Keep With Previous

Keeps the subform within the same content area as the previous subform.

Keep With Next

Keeps the subform within the same content area as the next subform.

After

Determines which area to fill after the subform is placed.

Continue Filling Parent

Continues merging data for all objects remaining to be filled in the parent subform.

Go to Next Content Area

Starts filling the next content area.

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Next Page

Starts filling the next page.

Go To Page > [name_of _page]

Starts filling the specified page.

Go To Next Odd Page

Starts filling the next odd page.

Go to Next Even Page

Starts filling the next even page.

Conditional Breaks

Specifies the number of conditional breaks that have been defined for the subform.

Overflow

Sets an overflow for a subform that spans pages.

None

Starts filling the next page (introduces a page break if one does not occur naturally).

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Page > [name_of _page]

Starts filling the specified page.

Overflow Leader

Specifies an overflow leader for a subform that must be paginated.

[name_of_subform]

Uses this existing subform as the overflow leader.

None

Does not use an overflow leader when paginating the subform.

New

Creates a new overflow leader subform in Designer and set it as the overflow leader when paginating this subform.

Overflow Trailer

Specifies an overflow trailer for a subform that must be paginated.

[name_of_subform]

Uses this existing subform as the overflow trailer.

None

Does not use an overflow trailer when paginating the subform.

New

Creates a new overflow trailer subform in Designer and set it as the overflow trailer when paginating this subform.

RELATED LINKS:

About subforms

Subform properties in the Subform tab

Subform properties in the Binding tab

Subform properties in the Binding tab

When you select a subform, the Binding tab in the Object palette displays several data binding options. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the subform. By default, subforms have no name. See *To name a subform*.

Data Binding

Sets the default data-binding method.

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. The subform will not be involved in any data-merging operations, but its objects (including any nested subforms) may still capture or display data.

Import/Export Bindings (Execute WSDL)

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

Initial Count

Sets the number of occurrences of the currently selected object to initially display on the form, regardless of the quantity of data from any data connection. The value you specify must be equal to or larger than the value for Min Count, and less than or equal to the value of Max.

RELATED LINKS:

Working with Data Sources

About subforms

12.24. Subform set

Subform set properties in the Subform Set tab

When you select a subform set, the Subform tab in the Object palette displays several formatting options.

NOTE: Edit Alternatives is unavailable unless the Select One Subform from Alternatives option is selected in the Type list.

Type

Sets the order that subforms within the subform set are rendered on a form.

Use All Subforms in Order

All subforms within the subform set are rendered in the order in which they appear on the form design.

Select One Subform from Alternatives

Only one of the subforms from within the subform set is rendered based on conditions specified by the form author.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

One-sided Printing Only

The object is printed only when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

RELATED LINKS:

Using subform sets

Subform set properties in the Pagination tab

Subform set properties in the Binding tab

Edit Data Nominated Subforms dialog box

Subform set properties in the Pagination tab

When you select a subform set, the Pagination tab in the Object palette displays several pagination options.

Place

Determines where the subform set will be placed.

Following Previous

Lays down the subform set after the previous object in the parent subform.

In Content Area > [name_of _content_area]

Lays down the subform set in the specified content area.

Top of Next Content Area

Lays down the subform set at the top of the next content area.

Top of Content Area > [name_of _content_area]

Lays down the subform set at the top of the specified content area.

On Page > [name_of _page]

Lays down the subform set on the specified page (introduces a page break if one does not occur naturally).

Top of Next Page

Lays down the subform set at the top of the next page (introduces a page break if one does not occur naturally).

Top of Page > [name_of _page]

Lays down the subform set at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Odd Page

Places the subform set on pages that are set to print on odd pages.

Top of Next Odd Page

Places the subform set at the top of the odd page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Even Page

Places the subform set on pages that are set to print on even pages.

Top of Next Even Page

Places the subform set at the top of the even page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

After

Determines which area to fill after the subform set is placed.

Continue Filling Parent

Continues merging data for all objects remaining to be filled in the parent subform.

Go to Next Content Area

Starts filling the next content area.

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Next Page

Starts filling the next page.

Go To Page > [name_of _page]

Starts filling the specified page.

Go To Next Odd Page

Starts filling the next odd page.

Go to Next Even Page

Starts filling the next even page.

Conditional Breaks

Specifies the number of conditional breaks that have been defined for the subform set.

Overflow

Sets an overflow for a subform set that spans pages.

None

Starts filling the next page (introduces a page break if one does not occur naturally).

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Page > [name_of _page]

Starts filling the specified page.

Overflow Leader

Specifies an overflow leader for a subform set that must be paginated.

None

Does not use an overflow leader when paginating the subform set.

[name_of_subform]

Uses this existing subform set as the overflow leader.

New

Creates a new overflow leader subform set and sets it as the overflow leader when paginating this subform.

Overflow Trailer

Specifies an overflow trailer for a subform set that must be paginated.

None

Does not use an overflow trailer when paginating the subform set.

[name_of_subform]

Uses this existing subform set as the overflow trailer.

New

Creates a new overflow trailer subform and sets it as the overflow trailer when paginating this subform.

RELATED LINKS:

Using subform sets

Subform set properties in the Subform Set tab

Subform set properties in the Binding tab

Subform set properties in the Binding tab

When you select a subform set, the Binding tab in the Object palette displays several data binding options.

Name

Sets the name of the subform set. By default, subform sets have no name.

Repeat Subform For Each Data Item

Enables or disables subform repetition.

Min Count

Sets the minimum number of occurrences of the currently selected object to display on the form.

Max

Sets the maximum number of occurrences of the currently selected object to display on the form.

Import/Export Bindings (Execute WSDL)

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Using subform sets

12.25. Table

Table properties in the Table tab

When you select a table, the Table tab in the Object palette displays several formatting options.

Source File

Sets the source file for the fragment reference. This property is visible only when the selected object is a fragment reference.

Fragment Name

Sets the name of the fragment. You can click the Frag Info button to view the fragment information.

This property is visible when a fragment reference or a fragment that is defined in a source file is selected. When the selected object is a fragment reference, this property does not appear if the source file is not specified. The Fragment Name list includes all the fragments in the specified source file. The

Custom option directly supports setting a SOM expression or an ID value as the fragment reference and supports the implementation in the XML Forms Architecture.

Body Rows

Shows the number of body rows.

Header Rows

Shows the number of header rows.

Columns

Shows the number of columns.

Footer Rows

Shows the number of footer rows.

Allow Page Breaks Within Content

Enables or disables page breaks in the middle of a table.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using tables

Table properties in the Row Shading tab

Table properties in the Pagination tab

Table properties in the Binding tab

Table properties in the Row Shading tab

When you select a table, the Row Shading tab in the Object palette presents options for applying shading colors to alternating rows.

Apply Alternating Row Shading

Enables or disables shading of body rows in a table.

For a table in which data is merged when the form is rendered, shading is applied only to body rows in which merged data appears. Shading is not applied to body rows that are not part of the data-merging process. All rows to which shading is applied must have a name. See *Body row properties in the Binding tab*.

First Sets the number of body rows that the first shading color is applied to, starting from the first body row.

Color Picker

Sets the first fill color used for shading.

Next

Sets the number of body rows that the second shading color is applied to, following the body rows shaded using the first shading color.

Color Picker

Sets the second fill color used for shading.

RELATED LINKS:

Using tables

Applying borders and shading to a table

Table properties in the Table tab

Table properties in the Pagination tab

Table properties in the Binding tab

Table properties in the Pagination tab

When you select a table, the Pagination tab in the Object palette displays several pagination options.

Place

Determines where the table will be placed.

Following Previous

Lays down the table after the previous object in the parent subform.

In Content Area > [name_of _content_area]

Lays down the table in the specified content area.

Top of Next Content Area

Lays down the table at the top of the next content area.

Top of Content Area > [name_of _content_area]

Lays down the table at the top of the specified content area.

On Page > [name_of _page]

Lays down the table on the specified page (introduces a page break if one does not occur naturally).

Top of Next Page

Lays down the table at the top of the next page (introduces a page break if one does not occur naturally).

Top of Page > [name_of _page]

Lays down the table at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Odd Page

Places the subform on pages that are set to print on odd pages.

Top of Next Odd Page

Places the subform at the top of the odd page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Even Page

Places the subform on pages that are set to print on even pages.

Top of Next Even Page

Places the subform at the top of the even page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

Keep With Previous

Keeps the table within the same content area as the previous table.

Keep With Next

Keeps the table within the same content area as the next table.

After

Determines which area to fill after the table is placed.

Continue Filling Parent

Continues merging data for all objects remaining to be filled in the parent subform.

Go to Next Content Area

Starts filling the next content area.

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Next Page

Starts filling the next page.

Go To Page > [name_of _page]

Starts filling the specified page.

Go To Next Odd Page

Starts filling the next odd page.

Go to Next Even Page

Starts filling the next even page.

Conditional Breaks

Specifies the number of conditional breaks that have been defined for the table. See *To control table,header row, body row, footer row, and section breaks using conditional statements*.

Overflow

Sets an overflow for a table that spans pages. See *To set up a table to span multiple pages*.

None

Starts filling the next page (introduces a page break if one does not occur naturally).

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Page > [name_of _page]

Starts filling the specified page.

RELATED LINKS:

Using tables

To set up a table to span multiple pages

Table properties in the Table tab

Table properties in the Row Shading tab

Table properties in the Binding tab

Table properties in the Binding tab

When you select a table, the Binding tab in the Object palette displays several data binding options. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the table. If existing tables in the form use the same name, the number beside the name identifies the instance for the current table. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML schema* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

Repeat Table For Each Data Item

Enables or disables table repetition.

Min Count

Sets the minimum number of occurrences of the currently selected object to display on the form.

Max

Sets the maximum number of occurrences of the currently selected object to display on the form.

Initial Count

Sets the number of occurrences of the currently selected object to initially display on the form, regardless of the quantity of data from any data connection. The value you specify must be equal to or larger than the value for Min Count, and less than or equal to the value of Max.

RELATED LINKS:

Using tables

Table properties in the Table tab

Table properties in the Row Shading tab

Table properties in the Pagination tab

Header row properties in the Row tab

When you select a header row or footer row, the Row tab in the Object palette displays several formatting options.

Source File

Sets the source file for the fragment reference. This property is visible only when the selected object is a fragment reference.

Fragment Name

Sets the name of the fragment. You can click the Frag Info button to view the fragment information.

This property is visible when a fragment reference or a fragment that is defined in a source file is selected. When the selected object is a fragment reference, this property does not appear if the source file is not specified. The Fragment Name list includes all the fragments in the specified source file. The Custom option directly supports setting a SOM expression or an ID value as the fragment reference and supports the implementation in the XML Forms Architecture.

Type

Sets the type of row content:

Body Row

The row contains table content.

Header Row

The row contains header information and is usually the first row positioned for the table. For example, a header row may identify the information in each table column.

Footer Row

The row contains footer information and is usually the last row positioned for the table. For example, for a table that presents numeric data, a footer row may provide the total value of each column.

Layout

Sets the layout of the header row.

Left to Right Row

Sets the layout of the cells in the header row from left to right.

Right to Left Row

Sets the layout of the cells in the header row from right to left.

Allow Page Breaks Within Content

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using tables

To change a row to a header, body, or footer row

Header and footer row properties in the Pagination tab

Header row properties in the Binding tab

Footer row properties in the Row tab

When you select a header row or footer row, the Row tab in the Object palette displays several formatting options.

Source File

Sets the source file for the fragment reference. This property is visible only when the selected object is a fragment reference.

Fragment Name

Sets the name of the fragment. You can click the Frag Info button to view the fragment information.

This property is visible when a fragment reference or a fragment that is defined in a source file is selected. When the selected object is a fragment reference, this property does not appear if the source file is not specified. The Fragment Name list includes all the fragments in the specified source file. The Custom option directly supports setting a SOM expression or an ID value as the fragment reference and supports the implementation in the XML Forms Architecture.

Type

Sets the type of row content:

Body Row

The row contains table content.

Header Row

The row contains header information and is usually the first row positioned for the table. For example, a header row may identify the information in each table column.

Footer Row

The row contains footer information and is usually the last row positioned for the table. For example, for a table that presents numeric data, a footer row may provide the total value of each column.

Layout

Sets the layout of the footer row.

Left to Right Row

Sets the layout of the cells in the footer row from left to right.

Right to Left Row

Sets the layout of the cells in the footer row from right to left.

Allow Page Breaks Within Content

Enables or disables page breaks in the middle of a row.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using tables

To change a row to a header, body, or footer row

Header and footer row properties in the Pagination tab

Header row properties in the Binding tab

Header and footer row properties in the Pagination tab

When you select a header row or footer row, the Pagination tab in the Object palette displays several pagination options.

Place

Determines where the header or footer row will be placed.

Following Previous

Lays down the row after the previous row.

In Content Area > [name_of _content_area]

Lays down the row in the specified content area.

Top of Next Content Area

Lays down the row at the top of the next content area.

Top of Content Area > [name_of _content_area]

Lays down the row at the top of the specified content area.

On Page > [name_of _page]

Lays down the row on the specified page (introduces a page break if one does not occur naturally).

Top of Next Page

Lays down the row at the top of the next page (introduces a page break if one does not occur naturally).

Top of Page > [name_of _page]

Lays down the row at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Odd Page

Places the row on pages that are set to print on odd pages.

Top of Next Odd Page

Places the row at the top of the odd page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Even Page

Places the row on pages that are set to print on even pages.

Top of Next Even Page

Places the row at the top of the even page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

Keep With Previous

Keeps the row within the same content area as the previous row.

Keep With Next

Keeps the row within the same content area as the next row.

After

Determines which area to fill after the row is placed.

Continue Filling Parent

Continues merging data for all objects remaining to be filled in the parent table.

Go to Next Content Area

Starts filling the next content area.

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Next Page

Starts filling the next page.

Go To Page > [name_of _page]

Starts filling the specified page.

Go To Next Odd Page

Starts filling the next odd page.

Go to Next Even Page

Starts filling the next even page.

Conditional Breaks

Specifies the number of conditional breaks that have been defined for the header or footer row. See *To control table,header row, body row, footer row, and section breaks using conditional statements*.

Include Header Row At Beginning Of Table

For a header row, specifies that the row will appear in its current position in the table.

Include Header Row In Subsequent Pages

For a header row in tables that span pages, adds the row as the first row in the table on each page.

Include Footer Row In Subsequent Pages

For a footer row in tables that span pages, adds the row as the last row in the table on each page.

Include Footer Row At End Of Table

For a footer row, specifies that the row will appear in its current position in the table.

RELATED LINKS:

Using tables

To repeat a header or footer row on subsequent pages

Header row properties in the Row tab

Header row properties in the Binding tab

Body row properties in the Pagination tab

Header row properties in the Binding tab

When you select a header row or footer row, the Binding tab in the Object palette displays several data binding options. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the row. If existing rows in the table use the same name, the number beside the name identifies the instance for the current row. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML data file* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

Repeat Row For Each Data Item, Min Count, Max, and Initial Count

Enables or disables row repetition. See *To set the rows in a table to adjust to accommodate data*.

RELATED LINKS:

Using tables

Header row properties in the Row tab

Header and footer row properties in the Pagination tab

Body row properties in the Binding tab

Footer row properties in the Binding tab

When you select a header row or footer row, the Binding tab in the Object palette displays several data binding options. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the row. If existing rows in the table use the same name, the number beside the name identifies the instance for the current row. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML data file* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

Repeat Row For Each Data Item, Min Count, Max, and Initial Count

Enables or disables row repetition. See *To set the rows in a table to adjust to accommodate data*.

RELATED LINKS:

Using tables

Header row properties in the Row tab

Header and footer row properties in the Pagination tab

Body row properties in the Binding tab

Body row properties in the Row tab

When you select a body row, the Row tab in the Object palette displays several formatting options.

Source File

Sets the source file for the fragment reference. This property is visible only when the selected object is a fragment reference.

Fragment Name

Sets the name of the fragment. You can click the Frag Info button to view the fragment information.

This property is visible when a fragment reference or a fragment that is defined in a source file is selected. When the selected object is a fragment reference, this property does not appear if the source file is not specified. The Fragment Name list includes all the fragments in the specified source file. The Custom option directly supports setting a SOM expression or an ID value as the fragment reference and supports the implementation in the XML Forms Architecture.

Type

Sets the type of row content:

Body Row

The row contains table content.

Header Row

The row contains header information and is usually the first row positioned for the table. For example, a header row may identify the information in each table column.

Footer Row

The row contains footer information and is usually the last row positioned for the table. For example, for a table that presents numeric data, a footer row may provide the total value of each column.

See *Header row properties in the Row tab*.

Layout

Sets the layout of the row.

Left to Right Row

Sets the layout of the cells in the row from left to right.

Right to Left Row

Sets the layout of the cells in the row from right to left.

Allow Page Breaks Within Content

Enables or disables page breaks in the middle of a row.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using tables

To change a row to a header, body, or footer row

Body row properties in the Pagination tab

Body row properties in the Binding tab

Body row properties in the Pagination tab

When you select a body row, the Pagination tab in the Object palette displays several pagination options.

NOTE: The Include options are available only when a header or footer row is selected. See *Header and footer row properties in the Pagination tab*

Place

Determines where the row will be placed.

Following Previous

Lays down the row after the previous row.

In Content Area > [name_of _content_area]

Lays down the row in the specified content area.

Top of Next Content Area

Lays down the row at the top of the next content area.

Top of Content Area > [name_of _content_area]

Lays down the row at the top of the specified content area.

On Page > [name_of _page]

Lays down the row on the specified page (introduces a page break if one does not occur naturally).

Top of Next Page

Lays down the row at the top of the next page (introduces a page break if one does not occur naturally).

Top of Page > [name_of _page]

Lays down the row at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Odd Page

Places the row on pages that are set to print on odd pages.

Top of Next Odd Page

Places the row at the top of the odd page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Even Page

Places the row on pages that are set to print on even pages.

Top of Next Even Page

Places the row at the top of the even page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

Keep With Previous

Keeps the row within the same content area as the previous row.

Keep With Next

Keeps the row within the same content area as the next row.

After

Determines which area to fill after the row is placed.

Continue Filling Parent

Continues merging data for all objects remaining to be filled in the parent table.

Go to Next Content Area

Starts filling the next content area.

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Next Page

Starts filling the next page.

Go To Page > [name_of _page]

Starts filling the specified page.

Go To Next Odd Page

Starts filling the next odd page.

Go to Next Even Page

Starts filling the next even page.

Conditional Breaks

Specifies the number of conditional breaks that have been defined for the row. See *To control table, header row, body row, footer row, and section breaks using conditional statements*.

RELATED LINKS:

Using tables

Body row properties in the Row tab

Body row properties in the Binding tab

Body row properties in the Binding tab

When you select a body row, the Binding tab in the Object palette displays several data binding options. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the row. By default, body rows have no name. If existing rows in the form use the same name, the number beside the name identifies the instance for the current row. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML data file* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

Repeat Row For Each Data Item, Min Count, Max, and Initial Count

Enables or disables row repetition. See *To set the rows in a table to adjust to accommodate data*.

RELATED LINKS:

Using tables

Body row properties in the Row tab

Body row properties in the Pagination tab

Header row properties in the Binding tab

Section properties in the Section tab

When you select a section, the Section tab in the Object palette displays several formatting options.

Type

Determines the sections that are displayed when the form is rendered, based on the section content.

Use All Subforms In Order

All rows in the section are included when the form is rendered.

Select One Subform from Alternatives

One row of two possible rows are included when the form is rendered. The row that is included is determined based on the data that is merged. For example, for rows with a cell that displays a calculated value, you can specify that one row is displayed if the value is equal to or higher than zero and another row if the value is less than zero.

RELATED LINKS:

Using tables

Working with table sections

Creating choice sections in tables

Section properties in the Row Shading tab

Section properties in the Pagination tab

Section properties in the Binding tab

Section properties in the Row Shading tab

When you select a section, the Row Shading tab in the Object palette presents options for applying shading colors to alternating rows within the section.

Apply Alternating Row Shading

Enables or disables body row shading in a section.

For a section in which data is merged when the form is rendered, shading is applied only to body rows in which merged data appears. Shading is not applied to body rows that are not part of the data-merging process. All rows to which shading is applied must have a name. See *Body row properties in the Binding tab*.

First

Sets the number of body rows that the first shading color is applied to, starting from the first body row in the section.

Color Picker

Sets the first fill color used for shading.

Next

Sets the number of body rows in the section that the second shading color is applied to, following the body rows shaded using the first shading color.

Color Picker

Sets the second fill color used for shading.

RELATED LINKS:

Using tables

Section properties in the Section tab

Section properties in the Pagination tab

Section properties in the Binding tab

Section properties in the Pagination tab

When you select a section, the Pagination tab in the Object palette displays several pagination options.

Place

Determines where the section will be placed.

Following Previous

Lays down the section after the previous section.

In Content Area > [name_of _content_area]

Lays down the section in the specified content area.

Top of Next Content Area

Lays down the section at the top of the next content area.

Top of Content Area > [name_of _content_area]

Lays down the section at the top of the specified content area.

On Page > [name_of _page]

Lays down the section on the specified page (introduces a page break if one does not occur naturally).

Top of Next Page

Lays down the section at the top of the next page (introduces a page break if one does not occur naturally).

Top of Page > [name_of _page]

Lays down the section at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Odd Page

Places the row on pages that are set to print on odd pages.

Top of Next Odd Page

Places the row at the top of the odd page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

On Even Page

Places the row on pages that are set to print on even pages.

Top of Next Even Page

Places the row at the top of the even page whenever the specified page is rendered (introduces a page break if one does not occur naturally).

After

Determines which area to fill after the section is placed.

Continue Filling Parent

Continues merging data for all objects remaining to be filled in the parent table.

Go to Next Content Area

Starts filling the next content area.

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Next Page

Starts filling the next page.

Go To Page > [name_of _page]

Starts filling the specified page.

Go To Next Odd Page

Starts filling the next odd page.

Go to Next Even Page

Starts filling the next even page.

Conditional Breaks

Specifies the number of conditional breaks that have been defined for the section. See *To control table, header row, body row, footer row, and section breaks using conditional statements*.

Overflow

Sets an overflow for a section that spans pages. See *To set up a table to span multiple pages*.

None

Starts filling the next page (introduces a page break if one does not occur naturally).

Go To Content Area > [name_of _content_area]

Starts filling the specified content area.

Go To Page > [name_of _page]

Starts filling the specified page.

RELATED LINKS:

Using tables

Section properties in the Section tab

Section properties in the Row Shading tab

Section properties in the Binding tab

Section properties in the Binding tab

When you select a section, the Binding tab in the Object palette displays several data binding options. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the section. If existing sections in the table use the same name, the number beside the name identifies the instance for the current section. By default, sections have no name. See *To name and rename objects*.

Repeat Section For Each Data Item, Min Count, Max, and Initial Count

Enables or disables section repetition. See *Working with table sections*.

RELATED LINKS:

Using tables

Section properties in the Section tab

Section properties in the Row Shading tab

Section properties in the Pagination tab

Cell properties in the Cell tab

When you select a cell, the Cell tab in the Object palette presents formatting options.

By default, cells contain text, and the Cell tab presents formatting options for the Text object. If you change the content of a cell, the Cell tab presents the formatting options for the associated object type. For example, if you change the cell content to a text field, the Cell tab shows the formatting options for a Text Field object.

You can set the cell type to any default object type. The formatting options that appear in the Cell tab are the same as the formatting options that are available if you add the object directly to the form. For example, the options in the Cell tab for a Text Field cell type are the same options as the Field tab for a Text Field object. Similarly, if the cell type is a circle, then the Cell tab displays the same options as the Draw tab for a Circle object.

In addition to the formatting options for the object type, the Cell tab presents the following options:

Merge Cells

Merges two or more adjacent, selected cells.

Split Cells Horizontally

When a merged cell is selected, splits the cells based on their original position.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

One-sided Printing Only

The object is printed only when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

RELATED LINKS:

Using tables

To change a cell to another object type

To merge and split cells

Cell properties in the Value tab

Cell properties in the Binding tab

Cell properties in the Value tab

When you select a cell that is set to an object type to which Value options apply (such as text field and numeric field objects), the Value tab in the Object palette displays several options. The options presented on the Value tab depend on the type of cell that is selected. For example, if you select a Text Field cell, the Value tab shows the options for a Text Field object.

The options that appear in the Value tab are the same as the options that are available from the Value tab if you add the object directly to the form.

RELATED LINKS:

Using tables

Cell properties in the Cell tab

Cell properties in the Binding tab

Cell properties in the Binding tab

When you select a cell that contains an object to which Binding options apply (such as text field or subform objects), the Binding tab in the Object palette displays several data binding options. The options presented on the Binding tab depend on the type of cell that is selected.

For example, if you select a Text Field cell, the Binding tab shows the options for a Text Field object. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

The options that appear in the Binding tab are the same as the options that are available from the Binding tab if you add the object directly to the form.

RELATED LINKS:

Using tables

Cell properties in the Cell tab

Cell properties in the Value tab

12.26. Text

Text properties in the Draw tab

When you create a text object, the Draw tab in the Object palette presents options specific to formatting the object.

Type

Sets the type of object. Objects are the building blocks of every form.

Allow Page Breaks Within Content

Allows page breaks within the content of a text object.

Keep With Next

Keeps a text object with the next object in the document when a page break is introduced.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using text

To allow page breaks within a text object

To keep a text object with the next object in the form

12.27. Text field

Text field properties in the Field tab

When you select a text field, the Field tab in the Object palette displays several options for formatting fields.

Type

Sets the type of object. Objects are the building blocks of every form.

Caption

Sets a caption for the object.

NOTE: Caption is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on

and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a caption*.

Appearance

Sets the border style for the fillable area of the field:

None

Does not display a border around the field.

Underlined

Underlines the field.

Solid Box

Displays a solid border around the field.

Sunken Box

Creates a shadow around the field so that the field looks three-dimensional.

Custom

Opens the Custom Appearance dialog box. Select this option if you want to define a custom look for the fillable area.

Allow Multiple Lines

Enables or disables multiple lines.

Limit Length and Max Chars

Sets the field length according to the maximum number of characters that you specify.

Limit Length to Visible Area

Sets the maximum number of characters allowed in the field according to the horizontal length of the text field.

Allow Page Breaks Within Content

Allows page breaks within the content of a text field object.

Keep With Next

Keeps a text field object with the next object in the document when a page break is introduced

Comb Of Characters

Enables the comb format.

Field Format

Enables plain text or rich text formatting of the field value in the form:

Plain Text Only

Enables plain text formatting.

Rich Text

Enables rich text formatting.

Patterns

Sets the pattern for displaying formatted values in a form, for syntax of the user input, for validating user input, and for storing and retrieving bound data or saving data when the form is not bound to a data source. See *Patterns dialog box*.

Presence

Controls whether an object is displayed in a PDF form when users view the form in Acrobat or Adobe Reader, or when users print the form:

Visible

The object is visible on-screen, visible in the printed form, and occupies space in the form layout. Visible is the default presence setting for all objects.

Visible (Screen Only)

The object is visible on-screen, not visible in the printed form when printed from within Acrobat or Adobe Reader, and occupies space in the form layout.

Visible (Print Only)

The object is not visible on-screen, visible in the printed form (when printed from within Acrobat, Adobe Reader, or directly from the server), and occupies space in the form layout.

Invisible

The object is not visible on-screen, not visible in the printed form, and occupies space in the form layout.

Hidden (Exclude from Layout)

The object is not visible on-screen, not visible in the printed form, and does not occupy any space in the form layout.

One-sided Printing Only

The object is only printed when using single-sided printing.

Two-sided Printing Only

The object is printed when using double-sided printing. This option is useful when you want to place the page number in a different corner of the page when using double-sided printing than when using single-sided printing.

Locale

Renders the data according to the specified locale for language and country or region. You can select a specific language and country from the list, or you can specify one of these options:

Default Locale

Uses the Default Locale specified in the Defaults tab of the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

NOTE: Locales in the Locale list are organized first by language and then by country or region.

RELATED LINKS:

Using text fields

Text field properties in the Value tab

When you create a text field, the Value tab displays several options that you can apply to the object.

Type

Enables run-time calculations and prompts:

User Entered - Optional

Users can choose whether to enter data.

User Entered - Recommended

A user is recommended to enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears and the user can choose to ignore the message and submit the form.

User Entered - Required

A user must enter a value in the field. If the user enters a value, leaves the field and then clears it, a custom message box appears (if written in the Empty Message box). A standard empty message appears if you do not type a custom message. A message only appears if there was a value in the field, the value was deleted, and the user exited the field without re-entering a value. If the user never attempts to enter a value in the field and tries to submit the form, a field is required message appears.

Calculated - Read Only

A data value will be calculated and displayed through an attached script. Users cannot edit the calculated value.

Calculated - User Can Override

A data value will be calculated and displayed through an attached script. Users can edit the value if the calculation script has been written to accept the input. If a user does edit the calculated value, the custom message you specify in the Override Message box appears.

Protected

Prevents a user from making changes to the value in the field. Indirect changes such as calculations can occur. The protected field is not included in the tabbing sequence and it does not generate events.

Read Only

A data value will be merged or calculated and displayed at run time. Users cannot edit the value.

Default

Sets an initial value for the field. The value is formatted according to the setting specified using the Patterns option on the Field tab, but must conform to the value according to the locale specified for the field. If the data is bound and a data pattern has been specified, the value must match the data pattern specified in the Binding tab. See *To specify a default value*.

NOTE: This option is available only when the Type option is set to User Entered - Optional, User Entered - Recommended, User Entered - Required, or Read Only.

Empty Message

Sets a message for prompting users to enter a recommended or required value. See *To prompt users to enter data*.

NOTE: This option is available only when the Type option is set to User Entered - Recommended or User Entered - Required.

Validation Pattern

Sets a validation pattern for validating user input. The pattern must match the syntax of the user input and be compatible with the data format selected on the Binding tab. See *To validate user input*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only, or Read Only.

Validation Pattern Message and Error

Sets a custom message to display when a raw value does not match the validation pattern. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation pattern message*.

Validation Script Message and Error

Sets a custom message to display when an attached validation script detects an unacceptable value. By default, this situation causes a programming error to be generated at run time. To generate a warning instead, deselect the Error option. See *To validate user input* .

NOTE: This option is not available when the Type option is set to Calculated - Read Only or Read Only. Validation Script Message is a dynamic property. Dynamic properties are identified by active labels that have a green underline that you can click to dynamically bind the property to a data source. To turn active labels on and off, use the Show Dynamic Properties command in the Object palette menu. See *To dynamically populate a validation script message*.

Form Level Validation Settings

Open the Forms Properties dialog box on the Form Validation tab. See *Form Validation (Form Properties dialog box)*.

Calculation Script

The value will be calculated by a script written for the object's `calculate` event.

NOTE: This option is available only when the Type option is set to Calculated - Read Only.

Runtime Property

Sets one of the following run-time properties for the object. When you select a run-time property, the actual value of the corresponding option is inserted dynamically when the form is viewed:

Current Page Number

Inserts the current page number of the finished form.

Number of Pages

Inserts the total number of pages that make up the finished form.

Viewer Locale

Inserts the locale identifier of the locale specified for the application that is processing the form. For example, the locale identifier for English (USA) is en_US, where *en* is the language and *US* is the country.

Viewer Name

Inserts the name of the application that is processing the form.

Viewer Version

Inserts the version number of the application that is processing the form.

Current Sheet Number

Inserts the number of the current piece of paper (sheet) in the numeric field.

Number of Sheets

Inserts the total number of pieces of paper (sheets) that make up the form.

NOTE: This option is available only when the Type option is set to Calculated - Read Only.

Override Message

Sets a custom message to inform users that they are changing the value of a calculated field. The message appears when a user changes the calculated value.

NOTE: This option is available only when the Type option is set to Calculated - User Can Override.

RELATED LINKS:

Using text fields

Text field properties in the Binding tab

When you create a text field, the Binding tab displays data binding options that you can apply to the object. Options that are not specifically related to creating a data connection apply both to data bound to a data source and data saved to a file when the object is not bound to a data source.

Name

Sets the name of the text field. See *To name and rename objects*.

Data Binding

Sets the default data-binding method:

Use name

Enables data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

Use global data

Associates a single data value with all objects that have the same name (see *To define a global field*).

New Data Connection

Starts the New Data Connection wizard. To define a connection using the wizard, see *To create a data connection to an XML data file* or *To create a data connection to an OLE database*.

No data binding

Disables data binding. Because the object will not capture or display merged data, any information associated with the object is not written as output when the form data is saved/submitted.

Import/Export Bindings

Sets an import/export binding for a WSDL data connection. See *To create a data connection using a WSDL file*.

RELATED LINKS:

Working with Data Sources

Using text fields

Text field patterns

Text field patterns

The following symbols must be used to create text patterns for text fields.

NOTE: The comma (,), dash (-), colon (:), slash (/), period (.), and space ( ) are treated as literal values and can be included anywhere in a pattern. To include a phrase in a pattern, delimit the text string with single quotation marks ('). For example, `'Your phone number is ('999')' 999-9999` can be specified as the Display Pattern.

Text symbol	Description	Example
`A`	Single alphabetic character	Display Pattern: A9A 9A9 Input Value: K1S5K2 Formatted Value: K1S 5K2
`X`	Single character	Display Pattern: XXX XXX Input Value: !1$5&2 Formatted Value: !1$ 5&2
`O`	Single alphanumeric character	Display Pattern: OO OO OO Input Value: 123ABC Formatted Value: 12 3A BC
`0`	Single alphanumeric character	Display Pattern: 00 00 00 Input Value: 123ABC Formatted Value: 12 3A BC
`9`	Single digit	Display Pattern: 999-9999 Input Value: 9407000 Formatted Value: 940-7000

NOTE: The `O` (capital "o") and `0` (zero) symbols are intentional synonyms in recognition that these two characters are often mistyped and misread.

Reserved symbols

The following symbols have special meanings and cannot be used as literal text.

Symbol	Description
?	When submitted, the symbol matches any one character. When merged for display, it becomes a space.
*	When submitted, the symbol matches 0 or Unicode white space characters. When merged for display, it becomes a space.
+	When submitted, the symbol matches one or more Unicode white space characters. When merged for display, it becomes a space.

RELATED LINKS:

Formatting field values and using patterns

Complex field patterns

Examples of text patterns

Examples of text patterns

Predefined patterns

You can quickly create and modify display, edit, and validation patterns to format data input values (user input or bound data) by selecting predefined patterns from the Select Type list on the Display tab in the Patterns dialog box (Object palette > Patterns). For example, the Postal Code (United Kingdom) option provides the pattern A9A 9A9|AA9A 9A9, such that an input value of KB1C3R2 produces a formatted value of KB1C 3R2.

Custom patterns

The following table shows some custom patterns that you can define in the Pattern box for formatting input values (user input or bound data) using a display pattern.

Custom pattern	Input value	Formatted value
'+1 ('999') '999-9999	6135551212	+1 (613) 555-1212
999.999.9999	6135551212	613.555.1212

In the next example, a serial number requires input like this: three alphabetic characters, followed by four digits, followed by a single character of any type. A suitable text pattern would be as follows:

```
AAA-9999-X
```

The following table below shows the results of converting some example input values (user input or bound data) into raw values using an edit pattern or data pattern.

Custom pattern	Input value	Raw value
AAA-9999-X	ABC-1234-5 ABC-1234-D 123-4567-8	ABC12345 ABC1234D ""

In the last example, because the input value 123-4567-8 does not match the pattern, the resulting raw value is an empty string.

RELATED LINKS:

Formatting field values and using patterns

When to use patterns

Defaults for value formatting

Using paper forms barcodes

Design tips for using a paper forms barcode

13. Working with Data Sources

You can create form designs that go beyond simple data capture. Using Designer, you can integrate your form designs with corporate databases or web services to build and maintain data capture solutions that read from, validate against, and add to corporate data sources. To do this, you connect your form design to one or more data sources and bind the data to one or more fields in your form design.

You can connect and bind fields to these data sources:

XML schemas

XML schemas specify how to formally describe the elements in an XML document. By connecting to an XML schema, you can bind the elements and attributes defined in the schema to fields in the form design. You can also select the XML data root element to use for the connection.

OLEDB databases

OLEDB is a Microsoft integration standard, which provides low-level application program interface (API) to data across an organization. By connecting to an OLEDB database, you can create form designs that serve as a data entry or capture tool for enterprise database servers. You can also use a database to provide content for portions of a form at run time.

Sample XML data

By connecting to an XML file, you can bind fields to the elements and attributes defined in the XML file. This method is useful when you do not have an XML schema, but you have an XML file that you can use as a sample.

Adobe Data Model

Connecting to an Adobe Data Model enables you to bind elements, attributes, and relationships described in the model to fields in the form. You can then use server-side processing to merge the form design with data based on the model.

Web Service Definition Language (WSDL) files

WSDL files describe web services by using XML. By connecting to a WSDL file, you can bind fields in a form design to one or many web services. You can create a data connection to a WSDL document on a web server and use HTTP/HTTPS or message level authentication, or both, to control access.

RELATED LINKS:

Importing digital IDs

13.1. About data connections

When creating interactive form designs, you can use external data sources (databases, XML data files, XML schemas, and web services) to define field values, populate fields on a form, or collect data entered on a form. Designer provides the data connection tool that lets you define connections to data sources.

You can then bind fields that accept these types of data to the data sources. If a field supports scripting, you can write FormCalc or JavaScript solutions for expressing business logic and creating calculations and validations. In addition, you can also take advantage of two custom objects, the Data Drop-down List and the Data List Box, to connect to a database to populate a list box. You can find these two objects on the Custom category of the Object Library palette.

The types of data you choose depend on your form and how you plan to distribute it. You can connect to one or more data sources and bind the data to one or more fields in your form design.

Type	Description
XML Schema	An XML schema specifies how to formally describe elements in an XML document. You can connect one XML schema to a form, and bind specific elements and attributes defined in the schema to fields in the form. You can use this to map data into and out of form fields in a format that conforms to the schema. For more information, see *To create a data connection to an XML schema*.
Sample XML Data	If you do not have an XML schema, you can connect an XML data file to a form, and bind elements and attributes in the file to fields in your form. The data captured in the form will conform to the XML data file. You can use the data file to provide the mapping of data into and out of form fields in the required format. For more information, see *To create a data connection to an XML data file*.
Adobe Data Model	Connecting to an Adobe Data Model enables you to create forms that use a data structure captured in AEM forms. Designer saves forms that have an Adobe Data Model data connection in XDP format only. For more information, see *To create a data connection to an Adobe Data Model*.
WSDL File	WSDL file web services provide functions and data to other applications through the Internet by using HTTP/HTTPS and message level (SOAP) authentication standards. The web service can receive and process a command and return the results to your form. You can connect one or more web services to a form. For more information, see *To create a data connection using a WSDL file*.
OLEDB Database	You can connect one or more databases to a form to exchange data between the form and the database. You can populate fields with data from a database or integrate data captured in the form with the database. Data exchange is possible for interactive forms in Acrobat using client-side binding. For more information, see *To create a data connection to an OLE database*.

Designer provides built-in tools for creating, binding, importing, and exporting from and to data sources. However, form authors who are not experienced in working with data sources should seek the assistance of database administrators and other technical staff to provide support for implementing the data connections.

Consider these points when planning how to set up and execute the data exchange required by your form:

- Identify the data sources you will use to capture user-entered data and to import data into form fields.

- Decide if you will use field binding to an XML schema or data file to define the structure of the data.
- Obtain data mapping information, including the data types and structures used by your existing data source.

NOTE: In Acrobat, security concerns dictate that you cannot specify an ODBC connection string using the Driver=; syntax, so the client system needs a DSN configured for ODBC connection. Also, if you are using external data sources in a form that you want to make available using Adobe Reader, you must set usage rights for the form using Acrobat Reader DC extensions 10.

RELATED LINKS:

Connecting to a data source

Binding fields to a data source

Data View palette

13.2. Updating data connections and the Data View palette

When you create a data connection, Designer creates a data description and saves it in the form. In addition, the Data View palette displays a hierarchy view of the data source. If the data source is an embedded XML schema or Adobe data model, the hierarchy is generated from the description saved in the form. If the data source is a non-embedded XML schema, an XML data file, or WSDL file, the hierarchy is generated using the external file.

When you reopen the form, Designer determines if the source file for the data connection has changed. If it has changed, you are prompted to update the connection. When you update the connection, Designer recreates the data description that is saved with the form file. It also regenerates the hierarchy in the Data View palette.

To prevent the update warning, in the Data Binding panel of the Options dialog box, clear the Show Update Dialog When Source File Changes option.

Updating the Data View palette for non-embedded XML schema data connections

When the source for a data connection is a non-embedded XML schema, an XML data file, or WSDL file, the hierarchy view in the Data View palette is generated using an external file. If the external file changes, when you reopen the form, the hierarchy in the Data View is regenerated to reflect the changes. If the external file changes when you are editing the form, a dialog box is displayed. This dialog box enables you to reload the hierarchy in the Data View palette.

To prevent these warnings, when you open the form file, update the data connection when you are prompted. Alternatively, in the Data View palette, select the name of the data connection, right-click, and then click Connection Properties to re-create the data connection.

13.3. To work with the Data View palette

- To show node details, select palette menu > Show Info.

- To expand a node, click the plus sign (+).
- To collapse a node, click the minus sign (-).
- To show the data description of the node (up to a maximum of 25 characters), select Show Data Descriptions from the palette menu. If there is no caption for the node, the data name is displayed. You must be connected to a data source that describes the data.
- To show both the data name and description of the node, select Show Both from the palette menu. You must be connected to a data source that describes the data.

RELATED LINKS:

Data View palette menu

Binding fields to a data source

13.4. Schema filtering

Schema filtering lets you choose the node to display in the Data View palette. When working with large schemas, you can use schema filtering to focus on several nodes and descendants at a time.

Schema filtering applies to the Data View palette only; the data description that is generated from the schema remains the same. Any binding that you create within the filtered Data View palette is the same as without filtering. Also, nodes that are not visible in the Data View palette remain valid, and validations continue to work for the complete data schema.

NOTE: When you choose which nodes to display, you cannot choose referenced data nodes. Referenced data nodes are generated when you connect to a source that is based on a relational data model.

To apply a filter to the Data View palette

1) In the Data View palette, select the top-level node.
2) In the Data View palette menu, select Filter Data View.
3) Choose the nodes to display and click OK.

To remove a filter from the Data View palette

1) In the Data View palette, select the top-level node.
2) In the Data View palette menu, select Clear Data View Filter.

RELATED LINKS:

To create a data connection to an OLE database

To create a data connection using a WSDL file

13.5. Connecting to a data source

Connecting a form design to a data source is the first step towards binding data to fields in the form design. After you create one or more data source connections in a form design, the Data View palette

displays a hierarchy view of each data source. The top nodes in the hierarchy represent each data connection and display the name that you assigned when you created the connection.

After you create a data connection, you need to bind the data to the fields in the form design.

Create a data connection to an XML schema

You can import an XML schema definition into a form design and then bind the objects in the form design to elements and attributes in the XML schema definition.

NOTE: If you are importing an XML schema that contains an <xs:include> or <xs:import> schema element, the element's schema Location attribute must be set to a file reference on the local file system.

1) Do one of the following:
   - Select File > New Data Connection.
   -
   - Select an object on the page and, in the Object palette, click the Binding tab and select New Data Connection from the Data Binding menu.
2) Select XML Schema and click Next.
3) Do one of the following tasks:
   - Click Browse and search for and select the XML schema file.
   - Type or paste the URL location of the XML schema file into the Select XML Schema File field.
4) Specify one or more of these options:
   - In the Use XML Data Root Element Name menu, select the root data element that will be used to name the root subform of the form design. This menu is populated with the global element declarations in the XML schema.
   - (Optional) Select Embed XML Schema to embed the XML schema as a packet in the XDP file.
   - (Optional) Select Transform Incoming Data and click Browse to search for and select the XSLT file that will be used to transform incoming data for the form design.
   - (Optional) Select Transform Outgoing Data and click Browse to search for and select the XSLT file that will be used to transform outgoing data for the form design.
5) Click Finish. The data appears in the Data View palette.

   NOTE: The at symbol (@) before **name** in the Data View palette indicates that this node is an attribute of an **option** node, and **#data** indicates the actual value of the **option***node.*

RELATED LINKS:

Binding fields to a data source

To create a bound field or subform

To bind referenced data nodes to fields or subforms

Create a data connection to an XML data file

If you do not have an XML schema, you can create a data connection to an XML file to use as an example of your data.

1) Do one of the following tasks:
   - Select File > New Data Connection.
   -
   - Select an object on the page. In the Object palette, click the Binding tab, and select New Data Connection from the Data Binding menu.
2) Select Sample XML Data and click Next.
3) Click Browse and search for and select the XML file.
4) Select one or both of these options:
   - (Optional) Select Transform Incoming Data, and then click Browse to search for and select the XSLT file that will be used to transform incoming data for the form design.
   - (Optional) Select Transform Outgoing Data, and then click Browse to search for and select the XSLT file that will be used to transform outgoing data for the form design.
5) Click Finish. The data appears in the Data View palette.

RELATED LINKS:

Binding fields to a data source

To create a data connection to an XML schema

To bind referenced data nodes to fields or subforms

Create a data connection to a model

You can use a data structure captured in other Document Services products to create your form design. In addition to providing UML-like capabilities, the Adobe application modeling technology supports advanced functionality, including derived properties, variants, constraints, filters, and styles. You create an Adobe Data Model connection by selecting an FML file.

A model data connection enables you to bind elements and attributes based on the model to fields in the form. You can also use the data structure to create bound fields and subforms. You can then submit the form to the server to merge it with data that is based on the model.

NOTE: Forms that are connected to an Adobe Data Model are saved in XDP format and preview as static PDFs. When you add an Adobe Data Model connection to a form, all interactive form features are disabled.

Do one of the following tasks:

- Select File > New Data Connection.
-
- Select an object on the page. In the Object palette, click the Binding tab, and select New Data Connection from the Data Binding menu.
- In the Name New Connection box, type a name for the connection.

The name must be one word and no longer than 127 characters. The first character of the name must be a letter or underscore (_). The remaining characters can be letters, digits, hyphens (-), underscores, or periods (.).

- Select Adobe Data Model and click Next.
- Click Browse and search for and select the FML file.
- Click Finish. The data schema appears in the Data View palette.

RELATED LINKS:

Binding fields to a data source

To create a data connection to an XML schema

To bind referenced data nodes to fields or subforms

Create a data connection to an OLE database

You may want your form to have access to information in a database. For example, you may want to retrieve data to populate a form based on a value, such as an employee ID, that the user enters when filling the form. When the user enters the ID number, it triggers a database lookup to retrieve the matching record and display it in the form.

To display, insert, update, or delete data from a database, you make a connection to the database and then bind the fields on the form to the corresponding columns in the database table.

You can also use scripting to perform the Insert, Update, Delete, and Next/Previous navigation of records in the database.

Determining when to use a database versus, for example, an XML data file, is up to the application developer. You could have multiple data connections configured for a form (one to a database, one to an XML file) and bind individual fields to either of the data connections.

OLEDB is the only database connection that you can connect to from Designer. Through the use of the Microsoft OLEDB Provider for ODBC Drivers, you can also access databases identified by the ODBC data source name (DSN). You can use OLE drivers other than the Microsoft OLEDB Provider for ODBC, however the form must be certified in Acrobat before it will function at run time.

In Acrobat, security concerns dictate that you cannot specify an ODBC connection string by using the `Driver=;` syntax. Therefore, the client computer needs a DSN configured for ODBC connections.

If you are using external data sources in a form design that you want users to fill in Adobe Reader, you must set usage rights for the form by using Acrobat Reader DC extensions 10.

1) Do one of the following tasks:
   - Select File > New Data Connection.
   -
   - Select an object on the page. In the Object palette, click the Binding tab, and select New Data Connection from the Data Binding menu.

2) In the Name New Connection box, type a name for the connection.

The name must be one word and no longer than 127 characters. The first character of the name must be a letter or underscore (_). The remaining characters may be letters, digits, hyphens (-), underscores, or periods (.).

3) Select OLEDB Database and click Next.
4) In the OLEDB dialog box, set up your database connection string. If you need assistance with this setup, refer to your database provider documentation.
5) Select one of these options then click Next:
   – Select the Table option and select a table as the record source. The list is populated with a list of all the tables available from the database identified by the connection string.
   – Select the Stored Procedure option and select the record source from the list. The list is populated with all of the stored procedures available from the database.
   – Select the SQL Query option and type the SQL query string.
6) Type your user name and password into the User and Password boxes and set the appropriate ADO record set options.
7) Click Next. Designer validates that the connection to the database is successful and prompts you for credentials, if required.
8) Click Finish. The data appears in the Data View palette.

RELATED LINKS:
Field types

About creating a data connection using a WSDL file

A web service, as defined by a Web Service Definition Language (WSDL) document, exposes a number of operations. For each operation, the WSDL file can define an input message, or an output message, or both. Input messages are sent to a server. The server can then reply with an output message. You can create and bind fields in a form design to one or many web services. The WSDL file describes the data description. The data description is used to build the WSDL data connection.

The WSDL data connection is different from the other types of data connections. A web service operation resembles a function call with input or output parameters or both. A particular field or fields may be the source of the input message and the destination of the output message.

Using Designer and a WSDL data connections, you can perform the following tasks:

- Bind one or multiple operations within one or more web services.
- Attach fields, subforms, and exclusion groups to the execution of a web service operation by a button click event.
- Execute a web service operation from any event by way of script.
- Enable script access to all returned elements of a web service regardless of whether those elements are bound to fields.
- Use the SOAP 1.1 style communication (SOAP bindings and an HTTP/HTTPS transport).
- Exchange data with a web service by using doc/literal exchange format.
- Write client-side scripts by using the Acrobat SOAP JavaScript Object (which supports RPC/encoded).

These features are not supported:

- Some XML Schema features
- Protocols such as SMTP, FTP, and so on as the underlying transport for SOAP
- Extensibility type elements
- Web service discovery that uses UDDI
- Remote Procedure Call (RPC) encoded SOAP messages
- Use of HTTP POST and GET WSDL bindings

You can create a data connection using a WSDL document on a secure web server, which requires HTTP/HTTPS client authentication to control access or message level authentication to request a web service or both.

HTTP/HTTPS authentication happens at the transport layer where client verification is required to access the WSDL document and to connect to a secure web service. HTTP/HTTPS authentication accepts user name and password, user name and password digest, and client certificate as acceptable authentication:

- User name and password authentication provides basic access verification, where the user is prompted with a login dialog box. The user name and password combination are sent as clear text and are not hashed before they are sent to the server.
- User name and password digest authentication provides enhanced verification, where the user is prompted with a login dialog box. However, the user name and password are hashed before they are sent to the server.

  NOTE: The login dialog box indicates the name of the server and the realm to which the user is attempting to connect. Because there may be several realms set up on a particular server, knowing this information can help the user to determine which credentials they require to log in.
- Client certificate authentication provides digital ID verification, where the user is prompted with the Select Digital ID dialog box listing the available digital IDs, and selects the correct digital ID to send to the server for authentication.

  NOTE: To use client certificate authentication, you need to have a digital ID available in the Windows Certificate Store or in the Designer Digital ID Files store, ready to use when you create a WSDL data connection. See *Importing digital IDs*.

Message level authentication happens at the SOAP (Simple Object Access Protocol) message layer where client verification requires a security token (included in the SOAP message header) to connect to a secure web service. The message level layer accepts user name and password authentication, where the user is prompted with a login dialog box. Acrobat 9.0 or later, supports user name and password message level authentication.

When you create a secure WSDL data connection, you specify (optionally) the type of credentials to accept for HTTP/HTTPS and message level authentication. The credentials for HTTP/HTTPS authentication are sent as part of the HTTP/HTTPS protocol header. The credentials for the message layer (SOAP) are used to create a security token (which is included in the SOAP message header). Keep in mind that you must specify the type of credentials to accept, when you have a server workflow where the server must provide credentials without user interaction.

NOTE: If you try to connect to a WSDL file in the Workbench repository, you may get an error indicating that the file cannot load. You must first open the WSDL file in Workbench before you create the data connection.

Create a data connection using a WSDL file

To create a secure WSDL data connection, you must know the location of the WSDL file. You must know the correct authentication credentials (user name and password or name of digital ID) required to access the WSDL file and to create the security token that is embedded in the web service request. You also need to decide whether you will use HTTP/HTTPS authentication or message-level authentication or both, and which credentials to accept for each authentication method.

1) Select File > New Data Connection.
2)
3) In the Name New Connection box, type a name for the connection.

   The name must be one word and no longer than 127 characters. The first character of the name must be a letter or digit. The remaining characters can be letters, digits, hyphens (-), underscores, or periods (.).
4) Select WSDL File and click Next.
5) In the WSDL File box, type the URL to the WSDL document, or click Browse to locate and select the document on your computer's hard disk. If the WSDL document is stored on a secure server, either the Authentication Required dialog box or the Select A Digital ID dialog box is displayed.
6) Perform one of the following actions:
   – In the Authentication dialog box, type the correct user name and password, and click OK.
   – In the Select A Digital ID dialog box, select the correct digital ID to use as a client certificate, and click OK.
7) Select the operation to call, and click Next.

   The window on the right displays details for the selected operation. If the selected operation is not SOAP-encoded or is not doc-literal, a warning appears in this window and you cannot continue until a you select a supported operation.
8) (Optional) To set up client authentication for a web service request, perform one of the following actions:
   – To define HTTP or HTTPS authentication, select Requires HTTP/HTTPS Authentication, and then optionally select the credentials to accept for verification.
   – To define SOAP message authentication, select Requires Message Level Authentication, and then optionally select Accepts User Name And Password.
9) Click Finish.

RELATED LINKS:

Authentication (New Data Connection dialog box)

To execute the connection to the web service

Field types

Execute the connection to the web service

You can execute the connection to the web service in various ways.

To execute the connection to the web service using a button

For WSDL data connections, the Data View palette includes a node representing a button.

1) Drag the node that represents a button onto the form design. This button executes that connection's operation.

To execute the connection to the web service from another event

You can execute the connection to the web service from another event instead of a button. For example, based on the value a user chooses from a list, you want to pass that value to the service.

1) Enter the following script on the event you want to trigger the web service:

   In JavaScript:`xfa.connectionSet.MyWSDLDataConnection.execute(0);`

   In FormCalc:`$connectionSet.MyWSDLDataConnection.execute(0)`

   In both cases, the parameter being passed is a Boolean value. When the results are returned from the web service, this value indicates whether the entire data DOM should be merged with the form or whether the returned values will replace those in the current instance of the form. Zero (0) indicates that the data will not be remerged; that is, no new fields or subforms will be added or removed directly as a result of this execute operation. One (1) indicates that all data in the data DOM will be remerged.

To view a button's connection information

1) Select a button on the page.
2) In the Object palette, click the Execute tab and view the connection information under Connection Info.

13.6. Modifying and deleting a data connection

Delete a data connection

You can delete a data connection from the form design.

1) In the Data View palette, select the data connection that you want to delete.
2) Select Delete Data Connection from the palette menu.
3) Choose one of these options:
   – To delete the bindings along with the database connection, select Remove Bindings For Deleted Data Connection.

– To delete the database connection but retain the data bindings, deselect Remove Bindings For Deleted Data Connection. The bindings are references to the form design data and are still valid even in the absence of the data connection, assuming the data remains the same.

4) Click Delete.

Modify a data connection

You can modify most of the data connection properties, except the data description type.

1) In the Data View palette, select the data connection that you want to modify.
2) Select Connection Properties from the palette menu.
3) Make the appropriate changes. Click Next to move through the dialog boxes.
4) Click Finish.

RELATED LINKS:

Binding fields to a data source

13.7. Binding fields to a data source

When you bind the fields in a form design to a data source, you create an association between the form design and the data source. The association allows you to capture, present, move, process, output, and print information that is associated with the form design.

The data description controls data generation in a form. It ensures that exported or saved data is valid according to the form's default data connection (either from an XML schema or sample XML data). Fields and subforms that are bound to data that is not defined by the data description or data connection produce warning messages in Designer. As well, the data in those fields is not exported in Acrobat.

If some fields are bound to a data source, data in fields bound by name is lost when the data is saved from Acrobat. When using a default data connection, all the fields for which you want the data saved must be bound to the data connection.

Designer enables you to use client-side or server-side binding. By using client-side binding, you can make immediate changes to the data. Server-side binding requires that you submit back to the server to commit changes to the data.

To perform client-side binding, you must have Acrobat or Adobe Reader (with a rights-enabled form) installed. You need Forms on a Microsoft server to do server-side database binding.

Binding to data not defined by the default data connection

You can override the default data binding by enabling settings to allow binding to data that is not defined in the default data connection. The Allow Binding To Data Not Defined By The Default Data Connection setting saves data in fields when they are bound by name or by explicit binding reference.

Saving or exporting undefined data in a separate namespace

To save or export the data in a separate namespace, you can select the namespace setting. The namespace lets you bind fields or subforms to data that is outside the default data connection. When set, the data description is automatically updated when nodes are added. Optionally, you can specify a prefix for the namespace. If you do not specify a namespace, the namespace information in the added nodes is inherited from the parents of the added nodes.

You can also specify that the added nodes are bound relative to a filtered node in the Data View palette.

Identifying nodes not bound by the default data connection

In the Data View palette, nodes that are not bound by the default data connection are dimmed. Keep in mind the following points when working with nodes that are not bound by the default data connection:

- You cannot drag the nodes to the form layout to create new nodes, and you cannot bind them to existing nodes.
- The node is not displayed in the Object palette> Binding tab.

To allow binding to data that is not defined by the default data connection

1) Select File > Form Properties.
2) Click the Data Binding tab and select Allow Binding To Data Not Defined By The Default Data Connection.
3) (optional) Do any of the following actions:
   – To add a namespace, select XML Namespace and type the namespace.
   – To add a prefix to the namespace, select XML Namespace Prefix and type the prefix.
   – To add data nodes in the filtered hierarchy, select Add Nodes Inside The Filtered Data Hierarchy.

RELATED LINKS:

Data View palette

Data Binding tab (Form Properties dialog box)

Ways to create bound fields or subforms

You can create a bound field or subform in one of the following ways:

Creating new form objects from the Data View palette

You can drag a node or a node and its descendants from the Data View palette onto the page and automatically create a bound field or multiple fields or subforms. The form field or subform, and its descendants, are bound to the data source. In addition, form field properties are automatically set according to the properties of the data binding. For example, the form object's type, name, and caption all reflect those of the binding object.

Associating a node from the Data View palette with an existing form field

You can drag a node from the Data View palette onto an existing field on the page and bind the two. You also have the option of updating the form field's properties to match those of the binding object. The properties that can be updated depend on the combination of form object type and binding type. For information about the properties that can be updated, see *Binding Properties dialog box*.

Specifying the binding node using the Binding tab in the Object palette

Select an object on the page and then select the node to which you want to bind the object in the Binding tab of the Object palette. The form field's properties can be updated in the same way as if you dragged a node from the Data View palette onto an existing form field. For information about the properties that can be updated, see *Binding Properties dialog box*.

Letting Designer auto-generate bound fields into a subform

You can generate new bound fields in a subform from the Data View palette.

To create a bound field or subform

1) Choose one of the following methods:
   - To create bound fields from all of the nodes in the Data View palette, drag the top node of a data connection onto the page. Designer creates subforms and objects that are bound to each of the nodes in the hierarchy.
   - To create a bound field, drag a node onto the page.
   - To create a bound subform, drag the node and its descendants onto the page.
2) Position the objects and subforms on the page to suit your requirements.
3) When you bind a text field.

   NOTE: When you bind a caption to a data source, the caption is always shown in the form, even if you have set Caption Position in the Layout tab to None.

RELATED LINKS:

To create a data connection to an OLE database

To create a data connection using a WSDL file

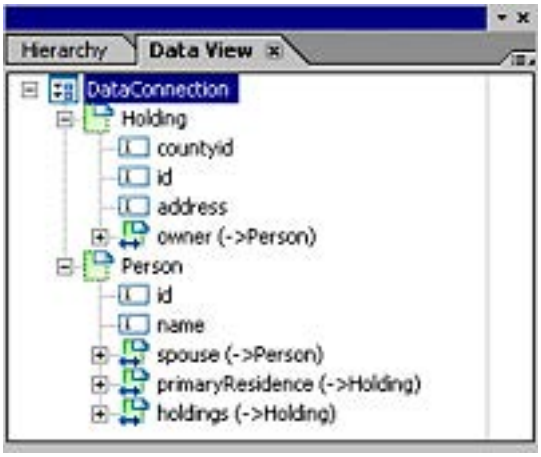
To bind referenced data nodes to fields or subforms

Field types

To bind referenced data nodes to fields or subforms

If the source of a data connection is based on a relational data model, associations among elements and attributes are displayed as referenced data nodes. Referenced data nodes have a unique icon and are named using both the association and the name of the target or referenced element.

This diagram shows referenced data nodes that are child nodes of the Holding and Person elements. The association (owner, spouse, primary Residence, holdings) and the referenced element (Person, Holding) are included in the node name.



You can use reference nodes to create subforms and fields by dragging the node or its parent to the page. You can also use the Data View palette menu Generate Fields command to create bound form objects. However, not all referenced data nodes are automatically available for binding to form objects. By default, when you create fields using a parent node, only the first level of descendants is included. Subsequent descendants (indicated by a red circle and bar in the icon) are not automatically included. Use the context (right-click) menu to include or exclude a referenced data node.

This diagram shows the default binding settings for the Person element. Referenced data nodes that are children of the element are included, but subsequent descendants are excluded.

- To include a referenced data node when its parent node is used to create bound objects, right-click the node, and then click Include Referenced Data.
- To exclude a referenced data node when its parent node is used to create bound objects, right-click the node, and then click Exclude Referenced Data.

RELATED LINKS:

Ways to create bound fields or subforms

To create a data connection to an XML schema

To bind a node to an existing field

You can bind a node to an existing field using the Data View palette or the Object palette.

To bind a node to an existing field using the Data View palette

1) In the Data View palette, select the appropriate node and drag it to the field on the page.
2) In the Binding Properties dialog box, select one of the following options:
   – Update All Related Properties
   – Update The Following Properties Only
   – Don't Update Any Related Properties
3) Click OK. Whether you update the object's properties or not, the form object is now bound to the data source.

To bind a node to an existing field using the Object palette

1) On the page, select the object that you want to bind to a node.
2) In the Object palette, click the Binding tab and select the node from the Data Binding menu.
3) In the Binding Properties dialog box, select one of the following options:
   – Update All Related Properties
   – Update The Following Properties Only
   – Don't Update Any Related Properties
4) Click OK. Whether you update the object's properties or not, the form object is now bound to the data source.

RELATED LINKS:

To create a data connection to an OLE database

To create a data connection using a WSDL file

Field types

To generate bound fields in a subform

1) Select the subform into which you want to generate bound fields.
2) In the Data View palette, select the node whose leaf nodes you want created in the subform.
3) Select Generate Fields from the palette menu. The new fields corresponding to the node are created in the subform.

RELATED LINKS:

To create a data connection to an OLE database

To create a data connection using a WSDL file

Field types

Create data bindings and conditional statements for choice subform sets

After you create a choice subform set, you can create data bindings with or without conditional statements for the various subform objects within the set. Using conditional statements provides better control over when subforms within the choice subform set are displayed on the form than if you specify only a data binding.

For each subform object listed in the Alternative Subforms list in the Edit Data Nominate Subforms dialog box, you can specify a data node from the data connection. If you specify a conditional statement, you can enter an expression that will evaluate to either `true` or `false` at run time. Only the first entry in the Alternative Subforms list with a conditional statement that evaluates to `true` will appear on your form.

NOTE: If you do not specify a conditional statement for an entry in the Alternative Subforms list, that entry is considered to have a value of `true`.

1) Select a subform set object.
2) In the Object palette, click the Subform Set tab and ensure that Select One Subform from Alternatives is selected in the Type list.
3) Select Edit Alternatives and choose how you want to specify a subform data binding from within the subform set by performing one of the following actions:
   - Select Choose Subform Whose Name Matches Data Element or Attribute to bind data nodes to subform objects by name. In this case, the names of the associated data nodes must match the names of the subform objects on your form design.
   - Select Choose Subform Using Expression to bind subform objects from your form design to data nodes from the data connection by manually specifying a binding.
4) In the Data Connection list, specify the data source you want to bind data from.
5) Click the Add button to insert a new subform into the choice subform set, or select an existing list item. Adding a new subform adds a duplicate of the currently selected subform. If no subform is selected, a new unnamed subform is added to the choice subform set.
6) In the Name field, type the name for a new subform object, if necessary. If you are binding the subform by data element or attribute name, ensure that the name in the field matches exactly with the name of the associated data node.

   If you selected either Choose Subform Whose Name Matches Data Element or Attribute in step 4, you can go directly to step 11. Otherwise, continue with step 8.
7) Click the triangle to the right of the Binding field and select a data node from the menu. Designer automatically populates the Binding field with a scripting reference to the data node you select. Alternatively, you can manually type a reference into the field.
8) Select a scripting language from the Language list.
9) In the Expression field, enter your script to perform the required actions or processing for the specified subform object and data node.
10) Repeat steps 6 to 10 for any additional alternative subforms.
11) Click OK when you finish adding entries to the list.

RELATED LINKS:

Using choice subform sets

Edit Data Nominated Subforms dialog box

Ways to show the same data in multiple places

You can display the same data in multiple places by using different methods:

- Set multiple fields to be global fields to display the same data that is entered interactively.
- Bind multiple fields to one node to display the same data from a data source in the fields.
- Use a script to display data that is entered into one field in another field.

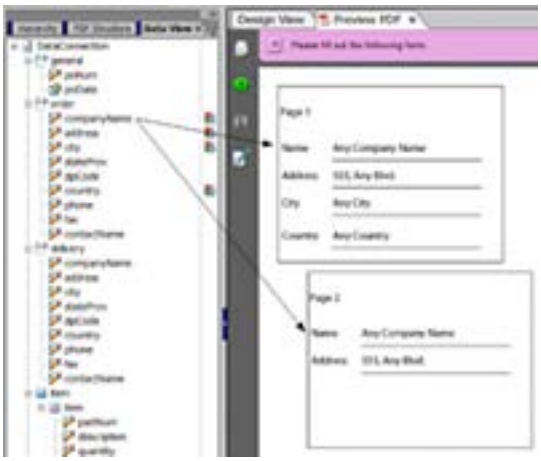
This table compares global fields, binding multiple fields to one node, and using a script to display the data that is entered into one field in another field.

Comparing...	Global fields	Binding	Script
What happens	Designer shows the same value in all fields that have the same name.	Designer shows the same value in all fields that have the same binding.	Designer displays the data that is entered into one field in another field.
Names of the fields	Names must be identical.	Names can be different.	Names can be different.
Run-time properties (such as the current page, number of pages, and current date/time)	You cannot add run-time properties to the field.	You cannot add run-time properties to the field.	You can add run-time properties to the field.
If you remove...	If you remove the global setting from one field, Designer removes the setting from all other fields with the same name.	If you remove the binding from one field, the other fields keep the binding.	Not applicable
Reference to nodes	Global fields can refer to nodes outside the current record.	Explicit data binding refers to nodes within the current record.	Not applicable

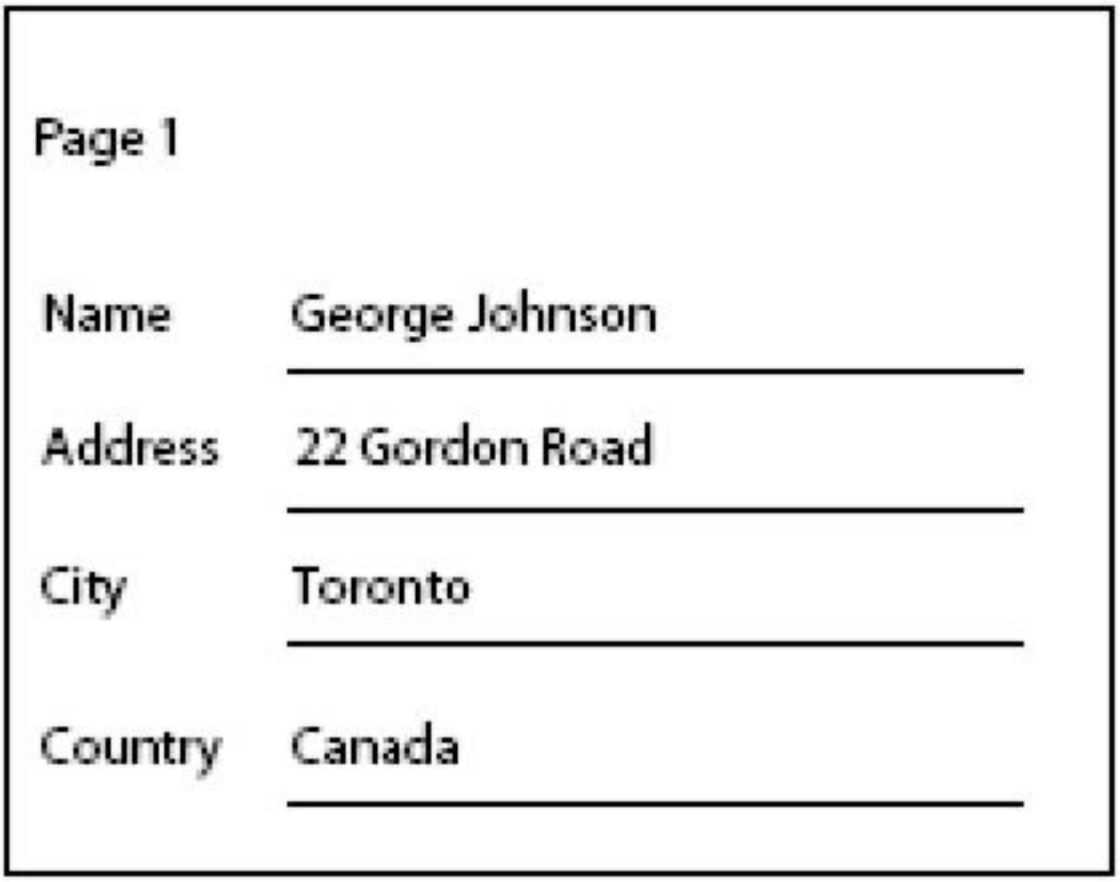
Comparing...	Global fields	Binding	Script
Example of when to use	Use for information that is repeated in the form, such as customer name or address. The user enters the information once and the data automatically fills the other fields that are set to global.	Use when you want a single instance of data from a data source displayed in multiple fields.	Use to copy the data entered from a Contact Name on page 1, and display it in a form letter on page 2.

This diagram shows what happens when you set multiple fields to be global fields. The name and address are filled in on Page 2 after the user enters the name and address on Page 1.

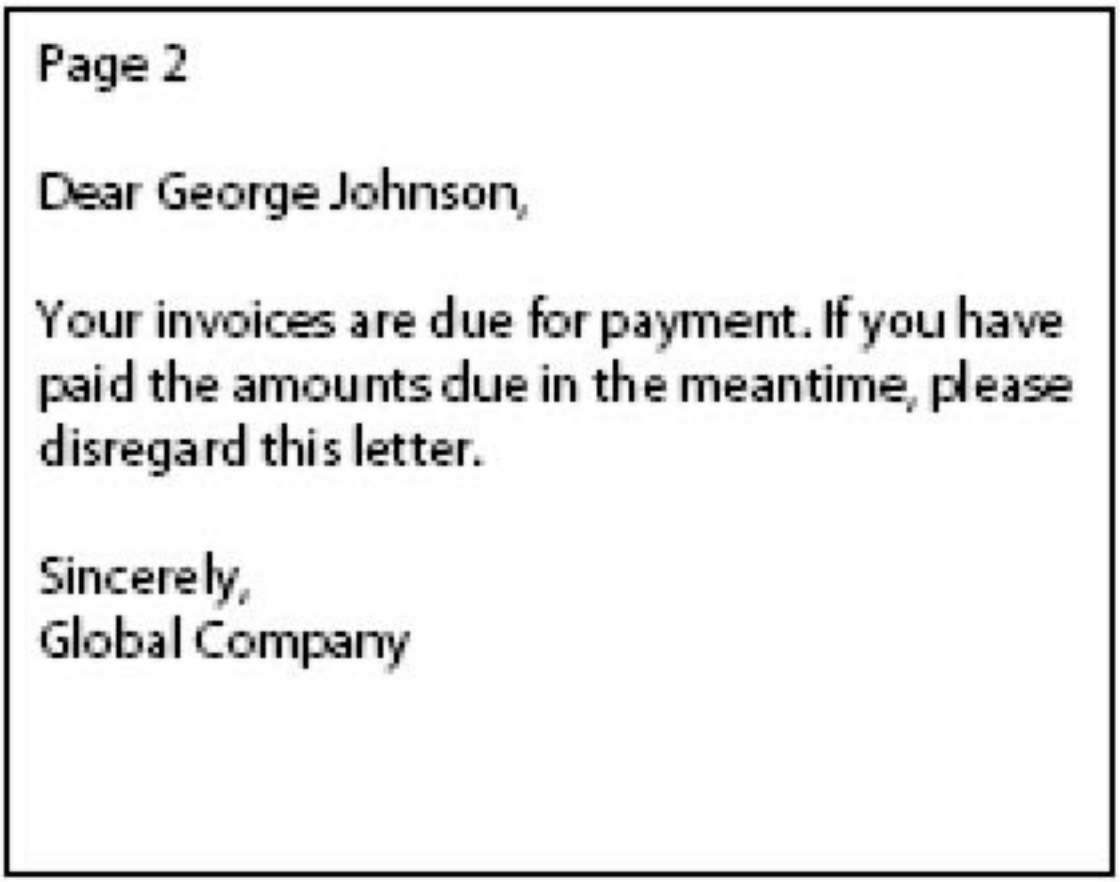
This diagram shows what happens when you bind multiple fields to one node. The name and address are filled in from the data source.



This diagram shows what happens when you use a script to display the data that is entered into one field in another field. The Contact Name in the form letter is the same as the name that is entered into the Name field on Page 1.



Dear {ContactName},



Bind multiple fields to one node

You can bind multiple fields to one node by doing one of the following actions:

- Associate a node in the Data View palette with existing form fields by using the drag-and-drop method.
- Explicitly specify the binding node by using the Binding tab in the Object palette for the existing form fields.

You should only bind a single instance of data to multiple fields. You should not bind repeating data to multiple fields.

1) In the Data View palette, select the appropriate node and drag it to the field on the page.
2) In the Binding Properties dialog box, select one of the following options:
   – Update All Related Properties
   – Update The Following Properties Only and select the options you want
   – Don't Update Any Related Properties
3) Click OK. Whether you update the object's properties or not, the form object is now bound to the data source.
4) Select the same node in step 1 and repeat steps 2 and 3 for the other fields that you want to bind to the same node.

Use a script to show the data that is entered into one field in another field

You can use a script to display the data that is entered into one field in another field. You can run this script on specific field-oriented events that trigger in response to user actions.

For example, an interactive form may prompt a user to enter the department name. This department name may be displayed on subsequent pages of the form (if it is a multipage interactive form). The field for the department name is bound to a data field, and this data field is where the department name is recorded. To display the department name on the other pages, the contents of the input field need to appear on other pages. The other instances do not need to be editable, and they may be part of another field (such as a floating field). In this case, you use a script to copy the department name to the other fields.

1) Select the first field.
2) If the Script Editor is not already showing, select Window > Script Editor.
3)
4) (Optional) Click the Expand button on the palette border until the palette is larger.
5) In the Show list, select a field-oriented event.
   For example, select Exit from the Show list.
6) In the Language list, select JavaScript and, in the Run At list, select Client.
7) In the Script Source field, type the script:
   For example, type the following script:
   ```
   ContactName.rawValue = this.rawValue;
   ```
   ContactName is the name of the second field from the Binding tab of the Object palette.
8) Select the second field.
9) In the Object palette, click the Value tab and, in the Type list, select Calculated - Read Only.

For more information about scripting, see Scripting Basics.

RELATED LINKS:
To define a global field

Set field generation options

You can specify the following options that will be applied when generating a field or group of fields:

- Generate validation scripts for new fields. The validation scripts generated are JavaScript scripts on the field. For example, a validation script can check that a field value is one from the allowed list.
- Generate descriptive information.
- Allow tables to be generated. Some data from a data source could be shown in a table. When you connect to a data source, the Data View palette identifies any items that could be tables. Designer also shows which items will become rows in the table if it finds a repeating data group.

1) From the Data View palette menu, select Options.
2) Deselect one of these options:
   – Generate Validation Scripts to disable generation of the validation scripts.

- Generate Descriptive Information to disable description information.
- Allow Tables To Be Generated

3) Click OK.

RELATED LINKS:

Field types

Remove a binding

You can remove binding from fields and subforms.

1) Select one or more objects from which you want to remove the binding.
2) In the Object palette, click the Binding tab and select No Data Binding from the Data Binding menu. Because the object will not capture or display data, any information associated with the object will not be written as output when the form data is saved.

   NOTE: When you disable binding from a subform, the subform will not be involved in any data-merging operations, but its objects (including any nested subforms) may still capture or display data.

RELATED LINKS:

To create a data connection to an OLE database

To create a data connection using a WSDL file

Field types

Refresh data connections

You can refresh all data connections in the form design. This command is useful when you have modified your files or database and want to update the form design.

1) From the Data View palette menu, select Refresh Data View.

RELATED LINKS:

To create a data connection to an OLE database

To create a data connection using a WSDL file

Field types

View a list of bound fields

You can generate a list of the following types of fields:

- Fields that have normal data binding
- Fields that are bound to a data reference
- Fields that have global binding
- Fields that have no data binding

- Unbound data connection node

1) In the Report palette, click the Bindings tab.
2) Select the appropriate option, and make a menu command from the Bindings menu:
   - To view a list of fields with normal data binding, select Fields With Normal Data Binding.
   - To view a list of fields with data binding by reference, select Fields With Data Binding By Reference.
   - To view a list of fields with global binding, select Fields With Global Data Binding.
   - To view a list of fields with no data binding, select Fields With No Data Binding.
   - To view a list of unbound fields, select Unbound Data Connection Nodes.

RELATED LINKS:

Field types

13.8. Field types

After you have created a data connection, you can create subforms and fields corresponding to all or some of the data. Often there is excess data that you do not need; therefore, selectively creating subforms and fields may be preferable to auto-creating them all. Another approach is to design the form design first and then use the Data View palette to bind specific fields in the form design to specific data nodes in the hierarchy.

Designer determines which field to create when you bind specific fields from an XML schema, XML file, or WSDL file, or when you generate fields from the Data View palette, based on the types of elements and attributes. For example, an image will be mapped to the Static Image object in the form design.

When you drag a node from the Data View palette onto the page to create a field or subform, the name of an XML schema element or attribute is used as the name of the generated field or subform. The element or attribute name is also used as the initial caption for the field. The binding value for the field is used to bind the field to the data node described by the element or attribute declaration in the schema.

The following See also section lists topics that provide more information about how Designer determines which field to create when you bind fields from an XML schema, XML file, or WSDL file.

RELATED LINKS:

To create a bound field or subform

To create a data connection to an OLE database

To create a data connection using a WSDL file

Simple types

XML schema element declarations can have simple or complex types. Elements with no element content and no attributes are declared with a simple type in the XML schema. Attribute declarations can have simple types. In general, elements with simple types and attributes are mapped to fields in Designer. The type of the field is determined from the simple type value.

Built-in simple types

The XML Schema defines a number of built-in simple types.

The following types are mapped to a Text Field object, which is any object with plain text content:

- string
- normalizedString
- token
- Name
- NCName
- QName
- language

The following types are mapped to the numeric field object:

- float
- double
- decimal
- integer
- long
- int
- short
- byte
- positiveInteger
- nonPositiveInteger
- negativeInteger
- nonNegativeInteger
- unsignedLong
- unsignedInt
- unsignedShort
- unsignedByte

The date and time types mappings are listed in the following table:

Date and time	Object type
duration	Text edit field
dateTime	Date/time field. The control type is Date and Time.
date	Date/time field. The control type is Date.
time	Date/time field. The control type is Time.
gYear	Text field with picture clause.

Date and time	Object type
gYearMonth	Text field with a picture clause
gMonth	Text field with a picture clause
gMonthDay	Text field with a picture clause
gDay	Text field with a picture clause

The following legacy types are mapped to the Text Field object:

ID

IDREF

IDREFS

ENTITY

ENTITIES

NMTOKEN

NMTOKENS

NOTATION

Other types are mapped in the following table:

Other	Object type
boolean	Check Box
hexBinary	Static Image
base64Binary	Static Image
anyURI	Text Field

Restricted simple types

New simple types can be derived from other simple types by using XML Schema restrictions. This section describes how these restrictions can be mapped to form designs.

Enumeration facets

An XML schema element or attribute declaration of any base type that declares one or more enumerated values will generate a drop-down list object rather than the default field for the declaration's base type, as shown by the following example:

```
<xsd:element name="Colors">
<xsd:complexType>
<xsd:simpleContent>
<xsd:extension base="xsd:string">
<xsd:enumeration value="red"/>
```

```
<xsd:enumeration value="green"/>
<xsd:enumeration value="blue"/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
<xsd:element>
```

The set of enumerated values will be stored as items for the drop-down list.

Bounds, length, and precision facets

Whenever possible, the restriction facets that control bounds (minInclusive, minExclusive, maxInclusive, and maxExclusive), length (length, minLength, and maxLength), and precision (totalDigits and fractionDigits) are mapped to properties on generated objects. Otherwise, they are used to generate validation scripts. Here are two examples:

- The maxLength facet is used to set the maxChars properties on the text value:
  <field><value><text maxChars="100"/></value></field>
- The fractionDigits facet is used to set the fracDigits property on the decimal value:
  <field><value><decimal fracDigits="3"/></value></field>

List and union types

List types allow definition of types whose values are whitespace-separated lists of single values. Elements and attributes that have list types will be mapped to Text Field objects.

Union types are types for which the set of allowed values are a union of two or more other sets of values. The various sets are not necessarily of the same base type. In general, union types will be mapped to Text Field objects. However, in the case where all the sets in the union are of the same type, the mapping will be the same as for any type by using one of those sets (that is, a union of two sets of enumerated values will generate a drop-down list with all values from the union).

RELATED LINKS:

Value ranges of XSD simple types mapped to numeric fields

Complex types

XML schema element declarations can have simple or complex types. Attribute declarations can have simple types. In general, elements with simple types and attributes are mapped to fields in Designer. Elements with complex types contain other elements and attributes, or attributes, and are generally mapped to subforms.

Complex types are used to define all other elements. Complex types can define element content, simple content, mixed content, and empty content.

Any complex type element may or may not declare attributes.

How a subform is generated from element content in a data source

An element that has only element content and optionally, attributes, will generate a subform when dragged into the form design.

This subform will contain objects for each attribute declared on the element and objects or subforms for each child element.

Simple content

Simple content is where an element is declared to contain simple content (text data) and may also declare attributes. In the case where the element declares no attributes, the mapping is the same as that for simple elements.

When attributes are declared, the mapping for the elements is to a subform that is named by using the element name. This subform will contain a field bound to the element and additional fields for each attribute. The name of the field bound to the element will be the name of the element with the string"_data"appended to it. An example of this is an element (for example,<phone>) that has numeric content and an attribute named`type`:

```
<phone type="work">5554567</phone>
```

This will generate a subform named `phone`containing objects named `type`and`phone_data`. The `type`object is bound to`$data...phone.type`, the `phone_data` field is bound to `$data...phone`.

Mixed content

Mixed content exists when an element is declared to contain a mixture of element children and text content.

Empty content

Complex types with empty content will be treated like simple content, where the object that is created, and bound to the element, has no content allowed.

Occurrence

XML Schema occurrence attributes (minOccurs and maxOccurs) for element declarations, which generate subforms, will be mapped to the repetition properties for the subform.

Content models

Where the content model for the children of an element is anything other than a sequence, the mappings will attempt to model choice and all groups by using subformSets.

Rich text objects

An element declaration that defines content from an XHTML schema will be mapped to a rich Text Field object rather that a plain Text Field object. That is, the<field> object that is created in the form design will have<value><exData contentType="text/html"/></value>.

Three situations arise in which an element will be identified as having rich text content:

- The schema imports the XHTML schema and declares an element containing a single child, which is the xhtml<body> element, as shown in the following example:

```
<xsd:schema xmlns:xsd="http://www.w3.ord/2001/XMLSchema"
xmlns:xhtml="http://www.w3.ord/1999/xhtml">
<xsd:import namespace="http://www.w3.org/1999/xhtml"/>
<xsd:element name="RichTextField">
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="xhtml:body"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

- A schema document declares a string type element with a default or fixed value that begins with the following content:

  <body xmlns="http://www.w3.org/1999/xhtml" ...

In this case, a rich text object is created with the value set to that of the default or fixed value.

- An element declaration includes an attribute of xfa:contentType with a fixed value of`text/html`, as shown in the following example:

  ```
  <xsd:schema xmlns:xsd="http://www.w3.ord/2001/XMLSchema"
  xmlns:xfa="http://www.adobe.com/2003/xfa">
  <xsd:import namespace="http://www.adobe.com/2003/xfa"/>
  <xsd:element name="RichTextField">
  <xsd:complexType>
  <xsd:simpleContent>
  <xsd:extension base="xsd:string">
  <xsd:attribute ref="xfa:contentType" fixed="text/html"/>
  </xsd:extension>
  </xsd:simpleContent>
  </xsd:complexType>
  </xsd:element>
  ```

  In this example, the xfa:contentType attribute will not be used to generate a field as in the general attribute case. Instead, it will be interpreted as a directive that the field generated should be a Rich Text Field. In terms of the namespace identifier, the form design has *xmlns="http://www.xfa.org/schema/xfa-template/2.1/**and the XDP file has**xmlns:xdp="http://ns.adobe.com/xdp/.*

Images

An image will be mapped to the static Image field object in the form design. Here are the attributes:

```
xfa:contentType
xfa:href
xfa:transferEncoding
```

Here is an example of such a declaration:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xfa="http://www.adobe.com/2003/xfa">
<xsd:import namespace="http//www.adobe.com/2003/xfa">
<xsd:element name="ImageField">
```

```
<xsd:complexType>
<xsd:extension base=”xsd:hexBinary”>
<xsd:attribute ref=”xfa:contentType” fixed=”image/jpg”/>
<xsd:attribute ref=”xfa:transferEncoding” fixed=”base64”/>
</xsd:extension>
</xsd:simpleContent>
</xsd:complexType>
</xsd:element>
```

In this case, the attributes will not be used to generate fields in the form design. They will be used to recognize the element as corresponding to an image field, and any default or fixed values for the attributes will be set on the generated<image>element.

Default/fixed values

Attributes' and elements' declarations that have a simple type or complex type with simple content may specify a fixed or a default value for the element or attribute.

Any element or attribute declaration that specifies a fixed or default value will generate a field where the initial value of that field is set to the fixed or default specified on the declaration.

Annotation information

Any annotation information that occurs as a child of an<xsd:element> or<xsd:attribute> in the XML Schema will be mapped to the field generated from that element or attribute declaration as<desc> content.

Namespaces

If targetNamespace is defined in the XML Schema, this information will be preserved and used to identify data elements. The namespace and namespace prefix information will not appear in the Data View palette or in the objects or subforms generated from the data connection. However, the namespace information will be preserved by way of the dataDescription generated from the schema that is used to map between the form design and the data described by the XML Schema.

Validation

Where the XML Schema provides information about the allowed values of an element or attribute (and therefore of the corresponding field in the template), the field generated will include a JavaScript validate script that verifies that the value is one allowed by the schema declaration. Such information can be provided by way of the data type of the element or attribute and also by restriction facets applied to the base type. Consider the following example:

```
<xsd:element name=”Number”>
<xsd:simpleType>
<xsd:restriction base=”xsd:integer”>
<xsd:minInclusive value=”4”/>
```

```
<xsd:maxInclusive value="9"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
```

The mapping will result in a Numeric field named "Number", which has a validation script that validates that any value is an integer between 4 and 9.

Many of the restriction facets provided by XML Schema provide information that might be used to generate a validation script. This information will only be used to generate validation scripts in cases where the information cannot be mapped directly to a property; for example, mapping the maxLength facet for a string type to <text maxChars="">.

Numeric type scripts

All numeric types in XML Schema are mapped to a single numeric field. However, the additional restrictions imposed by an XML Schema numeric type are maintained by generating validate scripts to ensure that a number value is the range allowed for the original XML Schema type. An "unsigned byte" type in the XML Schema will generate a numeric field with a script to validate that the value is an unsigned byte value; that is, this is the generated script:

```
this.rawValue >= 0 && this.rawValue <= 255;
```

Appropriate range-testing scripts will be generated for the following values:

- integer
- positive integer
- negative integer
- non-positive integer
- non-negative integer
- long, int
- short
- byte
- unsigned long
- unsigned int
- unsigned short
- unsigned byte datatypes

Inclusion/exclusion scripts

When a numeric type specifies min or max inclusion and/or exclusion values, a script will be generated to test that the value entered is within the range defined by these restriction facets. For a field that already has a numeric range script as defined in numeric type scripts in *Validation*, the script will test the range according to the most restrictive of the datatype range or the inclusion/exclusion range.

length, minLength, and maxLength scripts

An XML Schema string type may specify a length restriction facet or a minLength and/or a maxLength restriction facet. If the schema node is mapped to a text field, values specified for length and minLength restriction facets will be used to generate a validate script to test that the text field length is the value specified by length or is at least the value specified by minLength.

Note that any maxLength facet value will be mapped as maxChars for the text field.

When a string datatype maps to a pop-up list (because of the enumerated values), and length or min/maxLength, values will result in the validate script to check that the value selected in the list has a length that is valid according to the length, minLength, and maxLength facets.

totalDigits/fractionDigits scripts

The totalDigits facet specifies the maximum number of digits that can be used for a decimal or any integer type (that is, decimal, integer, positive integer, negative integer, non-positive integer, non-negative integer, long, int, short, byte, unsigned long, unsigned int, unsigned short, and unsigned byte datatypes). The fractionDigits facet is applicable for the decimal datatype only and specifies the maximum number of fractional digits that may be used.

If totalDigits or fractionDigits are specified for an element or attribute that is mapped to a numeric field, a validate script will be generated to test that the total number of digits and/or number of fraction digits are allowed by the restriction facets.

13.9. Dynamically populating form object properties from a data source

Dynamic properties are form object properties that are assigned values from a data source and are updated at run time. For example, the items in a drop-down list can be populated with a list of countries that are stored in a data source.

The following properties of form objects are dynamic and can be bound to values from a data source:

- List items in a drop-down list or list box
- A caption
- A validation pattern message
- A validation script message
- A tool tip
- Screen reader text

Dynamic properties allow you to modify form object properties outside of the form design and rely on a data source. This can be useful in deployment and maintenance scenarios.

In addition, the same data source can supply data to different form designs. For example, a long list of countries can be stored in one data file and used in many forms. You can use a wide variety of data sources, from complex databases to simple XML files.

NOTE: If you expect that a dynamically inserted text, such as caption or tool tip, might contain a hyperlink or a dot leader, select Acrobat and Acrobat Reader 9.0 or later as the target version for the form.

To identify which form object properties you can set dynamic properties for, Designer uses active labels. Active labels can be turned on and off using the Tools > Options > Data Binding dialog box and selecting the Show Dynamic Properties option. For example, here is the active label for the caption for a drop-down list before and after it is dynamically bound to a data source.

Before

After

In the Data View palette, this icon indicates that the node is dynamically bound to an object.

NOTE: If a node is bound to both a dynamic property and to a field, the default icon displays.

To dynamically populate form object properties from a data source, first you need to bind the form object properties to the data file and point to a data file that contains data to be used to populate the form object properties.

NOTE: You can bind the form object properties to the data file manually if you know the expected data file structure. In this way, you do not need a data connection.

The following table lists the objects and which active labels they each have.

Active Label/ Object	List Items	Specify Item Values	Caption	Validation Pattern Message	Validation Script Message	Tool Tip	Custom Screen Reader Text
Barcodes				Y	Y	Y	Y
Button			Y			Y	Y
Check Box			Y		Y	Y	Y
Date/Time Field			Y	Y	Y	Y	Y
Decimal Field			Y	Y	Y	Y	Y
Signature Field			Y			Y	Y
Drop-Down List	Y	Y	Y	Y	Y	Y	Y

Active Label/ Object	List Items	Specify Item Values	Caption	Validation Pattern Message	Validation Script Message	Tool Tip	Custom Screen Reader Text
Email Submit Button			Y			Y	Y
HTTP Submit Button			Y			Y	Y
Image						Y	Y
Image field			Y			Y	Y
List Box	Y	Y	Y	Y	Y	Y	Y
Numeric Field			Y	Y	Y	Y	Y
Password Field			Y	Y	Y	Y	Y
Print Button			Y			Y	Y
Radio Button					Y	Y	Y
Text						Y	Y
Text Field			Y	Y	Y	Y	Y

Set up dynamic properties

To identify which form object properties are dynamic, first you must turn active labels on.

Active labels identify which properties can be dynamic. Active labels are identified by a green underline that you can click to dynamically bind the property to a data source. You can set the default color for active labels.

For example, here is the active label for the caption for a drop-down list.

By default, dynamic properties are turned off.

To turn active labels on and off

1) Do one of the following actions:
   – From the Object palette menu, select Show Dynamic Properties.

– Select Tools > Options > and select Data Binding from the list on the left. Then select the Show Dynamic Properties option and click OK.

NOTE: If you turn active labels off, any form object properties that are already dynamically bound to a data source will remain bound and active.

To set the default color for active labels

1) Select Tools > Options and select Data Binding from the list on the left.
2) Select the Use Custom Label Color option.
3) Click the color picker and select a color.
4) Click OK.

Dynamically populate a drop-down list or list box

You can dynamically populate a drop-down list or list box with values from a data source.

If you have a predetermined and unchanging set of possible values, you can populate a drop-down list or list box when designing the form. See *Using drop-down lists and list boxes*. If you have a changing set of values, you can dynamically populate a drop-down list or list box from a data source at run time. If you have a long set of values, having it stored in a data source gives the form more flexibility.

For example, you can populate a drop-down list with a list of countries from a data source. Another example is to populate a list box with French product names but have English data values submitted to the data source.

For example, the schema you connect to could have the following defined for a list:

```
<form>
  <lists>
     <item uiname="item1" token="value1"/>
     <item uiname="item2" token="value2"/>
  </lists>
</form>
```

And the data file you point to could have the following defined for a list:

```
<form>
  <lists>
     <item uiname="MasterCard" token="MC"/>
     <item uiname="Visa" token="VS"/>
     <item uiname="Eurocard" token="EC"/>
     <item uiname="Special Card" token="SC"/>
  </lists>
</form>
```

When you merge the data file with the form design, you would see *MasterCard* as one of the options in the list, but when the form gets submitted, the token *MC* is sent to the data source.

TIP: If you have an OLEDB data connection, you can use the Data Drop-down List object or the Data List Box object from the Custom category of the Object Library palette to populate a drop-down list or list

box with data from a column in a table. See *Dynamically populate a drop-down list or list box from an OLEDB data connection*.

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *Set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

1) In the Object Library palette, click the Standard category and drag a Drop-Down List or a List Box onto the form design.

   For example, insert a Drop-Down List.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Credit Card`.

3) Do one of the following actions:
   - Click the List Items active label in the Field tab of the Object palette.
   - Click the Specify Item Values active label in the Binding tab of the Object palette.

   NOTE: For ease of use, both the List Items active label and the Specify Item Values active label display the same Dynamic Properties dialog box.

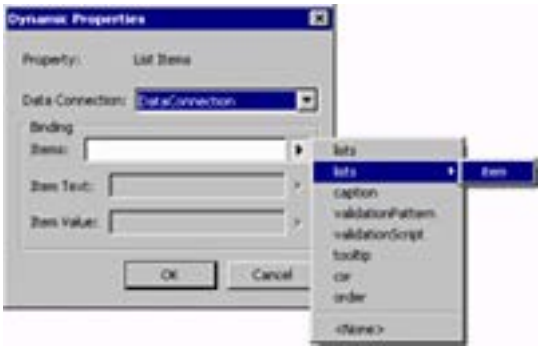
4) Select the data connection.

   NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

5) Click the triangle beside the Items box and select a binding.

   The binding you select should be a repeating data value or group so that the drop-down list or list box shows more than one item.

   For example, select lists > item.



   The following string appears in the Items box:

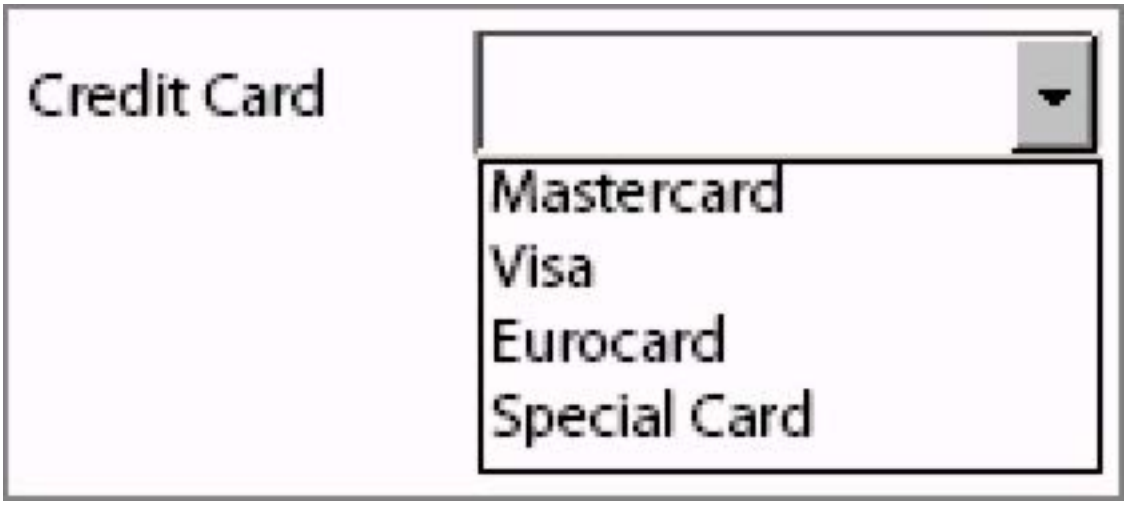
   `$record.lists.item[*]`

   TIP: You could type this string into the Items box instead of selecting it.

NOTE: For OLEDB data connections, the Items box is not available. Use the Item Text and Item Value boxes to select the database columns to use to populate the list.

6) To show a user-friendly item name at run time, click the triangle beside the Item Text box and select a binding.

For example, select @uiname to show MasterCard.



The data file you point to could have a repeating data value:

```
<form>
  <pymt>
     <cc type="MC">MasterCard</cc>
     <cc type="VS">Visa</cc>
  </pymt>
</form>
```

NOTE: If you select #data, the drop-down list shows *MasterCard.*

7) (Optional) To save a different value than the Item Text label to the data source, click the triangle beside the Item Value box and select a binding.

For example, select @token to save MC to the data file.

The data file you point to could have a repeating data value:

```
<form>
  <pymt>
     <cc type="MC">MasterCard</cc>
     <cc type="VS">Visa</cc>
  </pymt>
</form>
```

NOTE: If you select #data, the drop-down list shows *MC.*

8) Click OK.

After a binding has been set, a small link icon appears with the active label.

9) View the form in the Preview PDF tab.

Dynamically populate a drop-down list or list box from an OLEDB data connection

Two custom objects are available in the Object Library palette's Custom category that you can use to populate two columns with data from an OLEDB data connection. Both the Data Drop-down List object and the Data List Box object will populate two columns with data from an OLEDB data connection.

NOTE: This is a script-based alternative. You can use dynamic properties to accomplish the same thing. See *To dynamically populate a drop-down list or list box*.

1) In the Object Library palette, click the Custom category and drag the appropriate list object onto the page:
   - To create a drop-down list, drag the Data Drop-down List object.
   - To create a list box, drag the Data List Box object.
2) Select the object.
3) Select Window > Script Editor.
4) In `sDataConnectionName = "<value>"`, replace value with the data connection name.
5) In `sColHiddenValue = "<value>"`, replace value with the hidden value name.
6) In `sColDisplayText = "<value>"`, replace value with the display text column name.

RELATED LINKS:

To create a bound field or subform

To create a data connection to an OLE database

To create a data connection using a WSDL file

Field types

Dynamically populate a drop-down list in the Web DynPro environment

Two custom objects are available in the Object Library palette's Web Dynpro Native category that you can use to populate a drop-down list in the Web DynPro environment: Enumerated Drop-down list and Enumerated Drop-down list (no-select).

The Enumerated Drop-down list object contains a script for populating the list with data values. It uses data binding values to locate a list of elements in the data source. The script adds these items to the list. This object will select the first item in the list if the default value is not found.

The Enumerated Drop-down list (no-select) object is similar to the Enumerated Drop-down list object; however, it does not select the first item in the list if the default value is not found.

You can also customize the default value for the drop-down list.

Both objects contain a script that is invoked during the `preOpen` and `form:ready` events. The `preOpen` script executes when the form filler performs an action that causes the drop-down list to appear and the auto binding to the data source to take place. This event is initiated before the contents of the drop-down list is displayed. The `form:ready` event script is executed to display the selected item from the list. See `selectedIndex`.

A `complete` value, which is part of the node name is automatically supplied by Web DynPro. Here is an example:

```
<sap-vhlist>
<test complete="1">
...
</sap-vhlist>
```

If the value is "`0`", the data is partially populated and therefore the value selection is located by returning to the server. If the value is " `1`", the data is fully populated and no round trip to the server is required.

These two objects are designed for interactive forms built for Acrobat 8.0 and for use in the Web Dynpro environment.

To customize the default value in the drop-down list

1) In the Object Library palette, click the Custom category and drag the appropriate drop-down list object onto the page:
   - To create a drop-down list that selects the first item in the list if the default value is not found, drag the Enumerated Drop-down list object.
   - To create a drop-down list that does not select the first item in the list if the default value is not found, drag the Enumerated Drop-down list (no-select) object.
2) Select the object and then select Window > Script Editor.
3) Select the `form:ready` event.
4) To select a default value based on the index, refer to the following script example:

```
if (this.selectedIndex == -1) {
this.selectedIndex = 3; // Selects the fourth item in the list
Event.target.dirty = true;
}
```

5) To traverse through the list to select an item by value, refer to the following script example:

   JavaScript:

```
this.selectedIndex = -1;       // clear any existing selection
for (var i = 0; i < this.length; i++)
```

```
{
if (this.getSaveItem(i) == "FedEx")
this.setItemState(i, true);
 }
```

FormCalc:

```
this.selectedIndex = -1      // clear any existing selection
for var i = 0 upto DropDownList.length do
if (this.getSaveItem(i) eq "FedEx") then
// a non-zero value represents "true" in FormCalc
this.setItemState(i, 1)
endif
endfor
```

RELATED LINKS:

To set up dynamic properties

To create a data connection to an XML schema

Field types

Dynamically populate a caption

You can dynamically populate a caption with values from a data source.

For example, you can populate a caption with a product name in French. The schema you connect to could have the following defined for a caption:

```
<form>
  <caption product=”ProductName1”></caption>
</form>
```

The data file you point to could have the following defined for a caption:

```
<form>
  <caption product=”Desk Lamp”></caption>
</form>
```

When you merge the data file with the form design, the caption *Desk Lamp* is displayed for product.

Another example could be to provide the product name in another language. The data file you point to could have the following defined for caption:

```
<form>
  <caption product=”Lampe de Bureau”></caption>
</form>
```

When you merge the data file with the form design, the caption *Lampe de Bureau*is displayed for product.

NOTE: To use non-ASCII text (such as accents) in your data file, ensure the file is suitably encoded and the encoding is correctly identified in the XML processing instruction. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.

- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

1) In the Object Library palette, click the Standard category and drag an object onto the form design.

   For example, insert a Text Field object.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Product`*.*

3) Click the Caption active label.
4) Select the data connection.

   NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

5) Click the triangle beside the Binding box and select a binding.

   For example, caption > @product.

   NOTE: When you bind a caption to a data source, the caption is always shown in the form, even if you have set Caption Position in the Layout tab to None.

6) Click OK.
7) View the form in the Preview PDF tab.

Dynamically populate a validation pattern message

You can dynamically populate a validation pattern message with a value from a data source. This allows you to ensure users enter the correct value in the field.

For example, you can display a custom error message when users enter the wrong pattern in a field.

The schema you connect to could have the following defined for a validation pattern:

```
<form>
  <validationPattern dp_patternMessage="message1"/>
</form>
```

The data file you point to could have the following defined for a validation pattern:

```
<form>
  <validationPattern dp_patternMessage="DP Pattern Message - The order
```

```
number pattern is: A9A9A9"/>
</form>
```

When you merge the data file with the form design, and you type a number in the Order Number field that does not match the pattern expected, an error dialog box appears.

Another example could be to provide custom error messages for patterns in another language. The data file you point to could have the following defined for pattern validation:

```
<form>
  <validationPattern dp_patternMessage="Le numéro de commande doit suivre
le modèle suivant : A9A9A9"/>
</form>
```

When you merge the data file with the form design, and you type a number in the Order Number field that does not match the pattern expected, an error dialog box in French appears.

NOTE: To use non-ASCII text (such as accents) in your data file, ensure the file is suitably encoded and the encoding is correctly identified in the XML processing instruction. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

1) In the Object Library palette, click the Standard category and drag an object onto the form design.

   For example, insert a Text Field object.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Order Number`*.*

3) Click the Value tab and click Validation Patterns.
4) In the Select Type list, select a validation pattern.

   For example, select Postal Code (Canada) to display the pattern `A9A 9A9` *in the Pattern box.*

5) Click the Validation Pattern Message active label.
6) Select the data connection.

NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

7) Click the triangle beside the Binding box and select a binding.
For example, select validation > @dp_patternMessage.

The following string appears in the Binding box:
`$record.validation.dp_patternMessage`
TIP: You could type this string into the Items box instead of selecting it.

8) Click OK.

9) View the error message by using the Preview PDF tab.
For example, in the Preview PDF tab, enter an order number that does not match the pattern.

Dynamically populate a validation script message

You can dynamically populate a validation script message with a value from a data source. This allows you to ensure users enter the correct value in the field.

For example, you can display a custom error message when users enter something in a field that does not match what the script for a field expects. For example, a numeric field, such as Quantity, can have a script attached to it that says the number in this field cannot be greater than 100. If the user enters a number greater than 100, a custom error message appears.

The schema you connect to could have the following defined for a validation script:

```
<form>
  <validationScript dp_scriptMessage="message1"/>
</form>
```

The data file you point to could have the following defined for script validation:

```
<form>
  <validationScript dp_scriptMessage="DP Script Message - The quantity
must be less than or equal to 100."/>
</form>
```

When you merge the data file with the form design, and you type a number in the Quantity field that is greater than 100, an error dialog box appears.

Another example could be to provide custom error messages attached to a script in another language. The data file you point to could have the following defined for script validation:

```
<form>
  <validationScript dp_scriptMessage="Entrez un nombre inférieur ou égal
à 100."/>
</form>
```

When you merge the data file with the form design, and you type a number in the Quantity field that is greater than 100, an error dialog box in French appears.

NOTE: To use non-ASCII text (such as accents) in your data file, ensure the file is suitably encoded and the encoding is correctly identified in the XML processing instruction. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

1) In the Object Library palette, click the Standard category and drag an object onto the form design.

   For example, insert a Numeric Field object.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Quantity`.

3) In the Script Editor, select Validate from the Show list.
4) In the Script Editor, select a language from the Language box.

   For example, select JavaScript from the Language box.

5) Type a script for the object.

   For example, type this script for the Quantity field:

   ```
   this.rawValue <= 100
   ```

6) In the Object palette, click the Value tab and then click the Validation Script Message active label.
7) Select the data connection.

NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

8) Click the triangle beside the Binding box and select a binding.

For example, select validation > @dp_scriptMessage.

The following string appears in the Binding box:

`$record.validation.dp_scriptMessage`

TIP: You could type this string into the Items box instead of selecting it.

9) Click OK.

10) View the error message by using the Preview PDF tab.

For example, in the Preview PDF tab, enter a quantity greater than 100.

Dynamically populate a tool tip

You can dynamically populate a tool tip with a value from a data source. For example, you can display a custom tool tip for an object that is in French.

The schema you connect to could have the following defined for a tool tip:

```
<form>
  <tooltip dp_tt="tooltip1"/>
</form>
```

The data file you point to could have the following defined for a tool tip:

```
<form>
  <tooltip dp_tt="Quantité - Entrez un nombre inférieur ou égal à 100."/>
</form>
```

When you merge the data file with the form design, and you position the pointer over the Quantity field, you will see the following tool tip:

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.

- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

1) In the Object Library palette, click the Standard category and drag an object onto the form design.

   For example, insert a Text Field object.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Quantité`*.*

3) In the Accessibility palette, click the Tool Tip active label.

4) Select the data connection.

   NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

5) Click the triangle beside the Binding box and select a binding.

   For example, select tooltip > @dp_tt.

   The following string appears in the Binding box:

   `$record.tooltip.dp_tt`

   TIP: You could type this string into the Items box instead of selecting it.

6) Click OK.

7) View the form in the Preview PDF tab.

   For example, in the Preview PDF tab, try positioning the pointer over the Quantity field to see the custom tool tip.

Dynamically populate screen reader text

You can dynamically populate screen reader text with strings from a data source. For example, you can display custom screen reader text for an object that is in French.

The schema you connect to could have the following defined for custom screen reader text:

```
<form>
  <csr dp_csr="csr1"/>
</form>
```

The data file you point to could have the following defined for custom screen reader text:

```
<form>
  <csr dp_csr="The 7-digit customer identification field."/>
</form>
```

When you merge the data file with the form design, and you start the screen reader, it will read the type of field and then read “The 7-digit customer identification field.”

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.
- Ensure Acrobat is set up to Read Form Fields in the Preferences (Reading category) dialog box in the Edit menu.

1) In the Object Library palette, click the Standard category and drag an object onto the form design.

   For example, insert a Numeric Field object.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Customer ID`.

3) In the Accessibility palette, click the Custom Screen Reader Text active label.
4) Select the data connection.

   NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

5) Click the triangle beside the Binding box and select a binding.

   For example, select csr > @dp_csr.

   The following string appears in the Binding box:

   `$record.csr.dp_csr`

   TIP: You could type this string into the Items box instead of selecting it.

6) Click OK.
7) View the form in the Preview PDF tab.

   For example, in the Preview PDF tab, try entering data in the Customer ID field to hear the custom screen reader text.

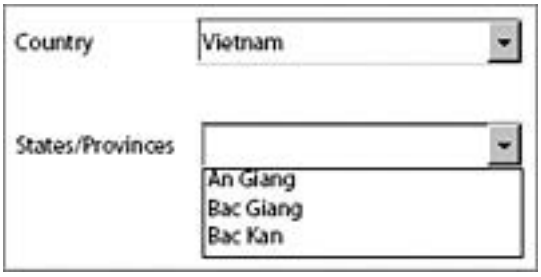
Dynamically populate a second field after populating the first

You can dynamically populate a second field after populating the first using scripting. For example, you can have one drop-down list that shows users a list of countries. After a user selects a country, the second list shows the states or provinces for that country.

For example, the sample XML file you connect to could have the following defined:

```
<MyData>
  <country/>
  <countries>
     <item uiname="United States" token="US"/>
     <item uiname="Vietnam" token="SRV"/>
  </countries>
  <state/>
  <US>
     <item>California</item>
     <item>New York</item>
     <item>Texas</item>
  </US>
  <SRV>
     <item>An Giang</item>
     <item>Bac Giang</item>
     <item>Bac Kan</item>
  </SRV>
</MyData>
```

For example, after you select Vietnam from the first drop-down list, the items listed in the States/Provinces list show only the ones that apply to Vietnam.



Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

To set up the first field

1) In the Object Library palette, click the Standard category and drag a Drop-Down List or a List Box onto the form design.

   For example, insert a Drop-Down List.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `Country`*.*

3) Do one of the following actions:
   – Click the List Items active label in the Field tab of the Object palette.
   – Click the Specify Item Values active label in the Binding tab of the Object palette.

4) Select the data connection.

   NOTE: If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

5) Click the triangle beside the Items box and select a binding.

   The binding you select should be a repeating data value or group so that the drop-down list or list box shows more than one item.

   For example, select countries > item.

   The following string appears in the Items box:

   `$record.countries.item[*]`

   TIP: You could type this string into the Items box instead of selecting it.

   NOTE: For OLEDB data connections, the Items box is not available. Use the Item Text and Item Value boxes to select the database columns to use to populate the list.

6) To show a user-friendly item name at run time, click the triangle beside the Item Text box and select a binding.

   For example, select @uiname to show United States.

7) (Optional) To save a different value than the Item Text label to the data source, click the triangle beside the Item Value box and select a binding.

   For example, select @token to save US to the data file.

8) Click OK.

   After a binding has been set, a small link icon appears with the active label.

To set up the script that dynamically populates a second field after populating the first

1) With the first drop-down list selected, in the Script Editor, select Change from the Show list.
2) Select JavaScript from the Language list.
3) Type the following script:

```
var tempString = "xfa.record." + this.boundItem(xfa.event.newText);
var oItems = xfa.resolveNode(tempString);
var nItemsLength = oItems.nodes.length;

DropDownList2.clearItems();

for (var nItemCount = 0; nItemCount < nItemsLength; nItemCount++) {
  DropDownList2.addItem(oItems.nodes.item(nItemCount).value);
}
```

Script	Description
`tempString`	Specifies the string for xfa.record. For example, if United States is selected, xfa.record becomes xfa.record.US.
`this.boundItem`	A list object method that translates from the user-friendly name to the value or token string. For example, when United States is selected, US is submitted.
`xfa.event.newText`	Returns the new text content of the list Change event. For example, when United States is selected, United States is submitted.
`var nItemsLength = oItems.nodes.length;`	Queries the number of child elements of <US> (or <SRV>), that is the number of <item> children.
`DropDownList2.clearItems();`	Removes any existing items in the second drop-down list.
`for (var nItemCount = 0; nItemCount < nItemsLength; nItemCount++)`	Queries for each <item> child of <US> (or <SRV>).
`DropDownList2.addItem(oItems.nodes.item(nItemCount).value);`	Gets the text value of the <item> child indicated by the nItemCount index and adds that value as a new item to the second drop-down list

To set up the second field

1) In the Object Library palette, click the Standard category and drag a Drop-Down List or a List Box onto the form design.

   For example, insert a Drop-Down List.

2) (Optional) In the Object palette, click the Field tab and type a caption for the object in the Caption box.

   For example, type `States/Provinces`*.*

3) Bind the field to the data source.

   For example, bind the state node to the States/Provinces drop-down list.

4) View the form in the Preview PDF tab.

Dynamically populate form object properties using the drag and drop method

You can quickly set up dynamic data binding for form object properties using the drag and drop method.

NOTE: You cannot use the drag and drop method for list items in a drop-down list or list box.

Before you perform this task, you must ensure the following settings are in effect:

- Ensure active labels are on. See *To set up dynamic properties*.
- Ensure you are connected to a data source that describes the data to which form object properties will be bound. See *Connecting to a data source*.
- To test the form with sample data, ensure you are pointing to a data file that has data defined for the form object property. See *To preview a form using sample data*.

1) Locate the active label for which you want to set up the dynamic data binding.
2) In the Data View palette, select the node that you want to link.
3) Click and drag the node to the active label.

   A small link icon appears with the active label for any dynamic property that is bound. For example:

   If the property already had a data binding set for it for the same data connection, it will be replaced.

Edit the dynamic data binding

1) Select the object that has dynamic data binding attached to it.

   A small link icon appears with the active label for any dynamic property that is bound. For example:

2) Locate the active label of the dynamic property for which you want to edit the data binding.
3) Do one of the following actions:
   – Click the active label.

– Right-click the active label and select Edit [property] Bindings.

4) Make any changes and click OK.

Remove the dynamic data binding

1) Select the object that has dynamic data binding attached to it.
   A small link icon appears with the active label for any dynamic property that is bound. For example:

2) Locate the active label for which you want to remove the dynamic data binding.
3) Right-click the active label and select Clear [property] Bindings.

   TIP: Another way to remove the dynamic data binding is to select <None> from the list in the Dynamic Properties dialog box. In addition, if you delete a data connection, both the default data bindings and the dynamic data bindings are deleted if you select the Remove Bindings For Deleted Data Connection option.

14. Creating Accessible Forms

14.1. About accessible forms

An accessible form is one that a wide range of people can use, including those who may have disabilities that affect how they are able to interact with the form on the computer screen. For example, the form user may have a visual impairment or reduced mobility.

Designer includes a number of features and capabilities that enhance the usability of forms for users with a range of disabilities, and that assist form authors in creating PDF forms that are more accessible to people with disabilities.

Designer provides support for creating accessible HTML (AHTML) files using Forms and tagged PDF forms. Tagged PDF forms include a complete logical structure plus additional information about a document's contents that substantially increases accessibility.

When creating accessible forms, you need to consider whether the form is intended for interactive filling by end users or whether the form is intended for viewing and printing purposes only.

You can also import existing accessible PDF files as the basis for your form. For more information, see *Importing tagged PDF files*.

14.2. Reading order versus tabbing order in accessible forms

Reading order is the sequence in which a screen reader reads all static objects (such as text and images) and field objects in a form. In PDF files, reading order is provided to the screen reader as a tree structure. Reading order always moves in a linear fashion and presents each piece of information on the form once. It also enables the user to read in the same order whether they are reading forward or backward.

Tabbing order determines how a user navigates through fields in a form using the keyboard. By default, Designer automatically sets the tabbing order for each form using geographical order (left-to-right, top-to-bottom). You can customize it using the Tab Order palette. The tabbing order is not necessarily linear.

Reading order restrictions

Designer uses the default or custom tabbing order to determine the reading order tree structure. However, because the reading order has different restrictions (for example, a linear structure) Designer generates the reading order and tabbing order separately. If you want both types of ordering to use the same sequence, ensure that the tabbing order conforms to the restrictions for the reading order.

Reading order and subforms

Subforms organize content on a form and improve performance when the user tabs between fields. The reading order treats subform objects as a group. All objects in a subform must be read before any objects in a subsequent subform are read.

Although you can customize the tabbing order to navigate between objects in different subforms, this type of navigation is not possible in the reading order. To maintain consistency between the reading and tabbing order, always tab through all the objects in a subform before navigating to another subform.

Static text objects and tabbing order

Static text objects are included in the tabbing order to enable you to modify their placement in the reading order. Headers and section titles are examples of static text objects. However, text objects are excluded from keyboard navigation. Changing the place of text objects in the tabbing order does not affect a user who is navigating using the keyboard.

RELATED LINKS:

Tabbing order

How the default tabbing order works

Changing the tabbing order

To position objects

To use the drawing aids

14.3. Designing accessible forms

When you design forms for a wide audience, you should consider that some users may have disabilities that affect how they interact with the form. Some users may have vision impairment, reduced mobility, or other disabilities. To accommodate the needs of all users, you may find it worthwhile to include certain practical design features in your forms and to test forms using various assistive technologies.

Tips for designing accessible forms

Here are some tips for designing accessible forms:

- A form is not accessible if it is not easy to use. You should try to design forms that are simple and usable. A simple layout of controls and fields with clear, meaningful captions and tool tips will make the form much easier for all users to fill.
- Designing forms that are uncluttered and logically arranged, and that provide clear and simple instructions, will help all users to fill forms as easily as possible. Navigation features, such as the tabbing order and keyboard shortcuts, should support the logical order of objects on the form. The caption for a field should be on the same line as the field's fillable area. Place captions consistently on the same side of the fields.

- Users work better with forms that reduce the possibility of making mistakes. Clear and concise error messages will help users fix any mistakes that they do make.
- Choose objects that make the form easy to use. For example, when used properly, tables are an effective way to organize and present tabular information. Avoid overly complex tables, such as those with nested tables and sections.
- Users should be able to fill the form completely using only the keyboard or an equivalent input device. Users with reduced mobility or impaired vision may have no choice but to use the keyboard, and many users who can use a mouse simply prefer keyboard input. By allowing various input methods, you not only create accessible forms, you also create forms that are better suited to the preferences of all users.
- Design your form to interact normally with other applications and system standards. For example, you could support standard Windows Control Panel settings for colors and use standard keyboard behavior.
- A well-designed form will be compatible with various assistive technologies. You should familiarize yourself with how these technologies work and use design techniques and user interface elements that are compatible with a wide range of assistive technologies.
- Color can greatly enhance a user experience with your form but only if it used properly. Colors can emphasize and enhance certain parts of your form, but you should not convey information by color alone. Large amounts of color may cause eye pain. Too much color can obscure foreground text. High-contrast colors, such as the default settings of black on a white background, are recommended.
- Images may help improve comprehension for users with some types of disabilities, However, many screen readers do not read graphics, which may decrease the accessibility of your form for users with vision impairments. If you choose to use images, provide text descriptions that describe the object and its purpose on the form.
- Be aware that client-side scripts can interfere with screen readers and keyboards if the script changes the focus of the client application. For example, the change and mouseEnter events, when used with drop-down lists or list boxes, have the potential to cause inappropriate actions. Client-side scripting should be written to avoid problems with screen readers and keyboards. Similarly, avoid scripting events that cause visual effects, such as blinking text, which may increase readability issues for users.
- Radio buttons are often misinterpreted, or difficult for users with disabilities to access on HTML forms because web browsers and screen readers treat them inconsistently. Avoid using radio buttons for HTML forms if you can use a list box instead.

RELATED LINKS:

Design tips for authoring forms for screen readers

Creating accessible PDF forms

Make simple tables accessible

Make complex tables accessible

Make images accessible

Make scripts accessible

14.4. Forms for users with vision impairment or disabilities

Users with vision impairment or disabilities typically use an assistive technology called a screen reader, which speaks to the user. A screen reader can relay information about where the cursor is displayed on the computer. The screen reader can also read the tool tips, so it is very important to include tool tips in your design.

In addition to tool tips, you can use captions, object names, or custom screen reader text to clearly identify each field. You can also use text objects to indicate what kind of input is required. When indicating which fields require input, be explicit. For example, tool tip text can contain the words “Required input”.

To ensure that a screen reader will work with a PDF form, each object should provide spoken information, and you must generate a tagged PDF form.

If you will be creating an HTML form from the same form design, screen readers can read and speak tool tips from AHTML transformations.

Keep the following design considerations in mind to assist users with vision impairment or disabilities:

- Users with vision impairment or disabilities cannot use a mouse to interact with a computer. Every function must be available using the keyboard. For most forms, the Tab and Shift+Tab keys move the cursor forward and backward through the form. Be sure to set a logical tabbing order that includes all fields and buttons.
- Many forms include administrative information that provides additional instructions to the user on how to complete the form. Use tool tips to ensure that all important information is read to the user at an appropriate time. For example, if keystrokes are required to perform a function, such as pressing the space bar to select a button or the Down Arrow key to select an item from a list box, the keystrokes should be announced to the user.
- Include enough spoken information for users to know what input is expected and how to complete the field correctly, but do not overwhelm users with redundant information.
- Add navigational buttons to the top and bottom of the form. For example, at the top of a form, you could include buttons such as Open Data File, Previous Page, and Next Page. At the bottom of the form you could include buttons such as Save Data, Email Data, Go to Top of Page, and Print.
- Smart fields can be an effective way to make some forms easier to fill. For example, a travel request form may have several rows and columns of fields. The last field in each row could check if any data was entered in the row. If the row is empty, the Tab key could jump to the next section of the form rather than continuing to tab through a number of fields that will remain empty.
- Users with vision impairment or those who are color blind may have difficulty seeing the cursor. Use appropriate colors to display objects using a color scheme selected by the user.
- Users with vision impairment typically use an assistive technology such as a magnification program. This product enlarges the cursor. Test your forms with a number of the more popular magnification programs to ensure that your forms are accessible.

14.5. Forms for users with reduced mobility

For users with reduced mobility and those with limited dexterity, completing a form can be very difficult. If a user is unable to use a mouse, every function in a form must be available using the keyboard.

A variety of assistive technologies are available for users with reduced mobility and those with limited dexterity, including special keyboards, speech-to-text software where the user speaks into a microphone to complete the form, or even more sophisticated technologies that can operate a computer by breath or eye movements.

Test your forms with a number of these assistive technologies to ensure that your forms are accessible.

14.6. Color in accessible forms

Color can greatly enhance a user's experience with your form if it is used properly. Large amounts of color may cause eye pain. Too much color can obscure foreground text.

Designing forms for accessibility involves considering some additional guidelines for using color. Colors can emphasize and enhance certain parts of your form, but you should not convey information by color alone.

For example, a red asterisk often indicates that a text field requires user input. Because users with vision impairment may have difficulty seeing the color red, a better choice for accessible forms is to set the text field type to User Entered - Required and define a message that indicates that the field requires input.

Many users with vision impairment rely on high contrast between text and the background to read the form. We strongly recommend that you use the default font and background colors, Black on a white background provides a high-contrast that improves the readability of the form. If you must change these default colors, ensure you choose an appropriate combination of high-contrast colors.

Design your form to interact normally with other applications and system standards, including support for standard Windows Control Panel settings for colors. Users with vision impairment or those who are color blind may have difficulty seeing the cursor. Use standard settings for color may help eliminate this issue.

While designing your form, test it frequently using a color scheme setting similar to what many users with vision impairment will be using to complete your form. This practice helps you discover and correct issues early in the design process.

RELATED LINKS:

Font properties in the Font palette

Paragraph properties in the Paragraph palette

Border properties in the Border palette

Techniques for testing form accessibility

14.7. Support for screen readers

To access spoken information about a form, users with vision impairment need to use a screen reader. Screen readers speak information about where the cursor is located on the form. As the form author, you must specify the text for the screen reader to read. If the form will be saved in PDF you must also generate a tagged PDF form.

Designer supports the Microsoft Active Accessibility (MSAA) compliant screen reader, JAWS for Windows from Freedom Scientific.

RELATED LINKS:

Design tips for authoring forms for screen readers

Accessibility properties in the Accessibility palette

14.8. Design tips for authoring forms for screen readers

To provide an accessible form, you need to understand how screen readers work. You also need to know how to use the Designer Accessibility palette to specify information that the screen reader will speak for the objects on forms. Keep in mind these considerations related to implementing accessible forms that can support screen readers:

- Introduce the form to tell the users what form is open and how to proceed with its completion. When a form opens, the screen reader will generally read through an entire page, including any text and image objects. When the user begins to tab through the form, the screen reader reads text for each object as it becomes active.
- Establish a logical tabbing order that includes all text, fields, buttons, and images to make it easier for users with visual impairment to navigate. Setting the tabbing order is important because PDF screen reader tags are ordered by default from top-to-bottom, left-to-right.
- Announce to users any special keystrokes they need to make in order to perform a function. Such keystrokes include actions such as pressing the spacebar to select a button, or the Down Arrow key to select an item from a list box.
- Announce the names of buttons and fields, as well as their purpose, when the user tabs into them.
- Announce the state of check boxes and radio buttons.
- In list boxes and drop-down lists, announce the default item selected in the list. Be sure that the user knows to use the Up Arrow and Down Arrow keys to move through the list items. Pressing the Tab key or the Enter or Return key will select the item in the list. Using scripting, you can set the object's Change event to announce which item is selected from the list.

RELATED LINKS:

Accessibility properties in the Accessibility palette

Specify custom screen reader text for an object

Specify a custom tool tip for an object

Specify a different search order for screen reader text

Turn off screen reader text for an object

14.9. Create a PDF form with accessibility tags

In addition to specifying text for screen readers, you must also create a tagged PDF form so that the screen reader can read the text. You do this by generating accessibility tags when saving the form design as a PDF file.

In Designer, the default behavior is to create tagged PDF forms.

NOTE: To display accessibility tags in forms with a flowable layout in Acrobat, you must run the screen reader before opening the form in Acrobat.

1) Select File > Form Properties.
2) To generate accessibility tags, click the Save Options tab and, in the PDF area, ensure that Generate Accessibility Information (Tags) For Acrobat is selected.
3) Click OK.

RELATED LINKS:

Color in accessible forms

Creating accessible PDF forms

14.10. Create a PDF form with accessibility tags

In addition to specifying text for screen readers, you must also create a tagged PDF form so that the screen reader can read the text. You do this by generating accessibility tags when saving the form design as a PDF file.

In Designer, the default behavior is to create tagged PDF forms.

14.11. Creating accessible PDF forms

For a PDF form to be accessible, it be created as a tagged document. The goal is to ensure that all fields are placed in the logical structure within the tag tree and that they have appropriate text descriptors using tool tips or captions.

Design tips for creating accessible PDF forms

These tips will help you create accessible forms for Acrobat and Adobe Reader:

- Circle, line, and rectangle objects are not tagged in PDF forms. These objects add no useful information for a user with vision impairment or disabilities. Screen readers will not read any information for these objects.
- Do not set the Speak Order Precedence to None for field objects. If you do, Acrobat will cause the screen reader to say “MSAA data prompt not available” when the user tabs into the field, even if the field is read-only.
- Subforms are useful for organizing related objects and provide a logical tabbing structure. The default tabbing order is in geographic order, left-to-right, top-to-bottom. If two subforms exist

side-by-side, and each subform contains a number of field objects, the tabbing sequence will go through the fields in the first subform before moving on to the next.

- When tabbing into a radio button, the screen reader first reads the text for the group object that contains it, and then the On value for the radio button. You should set custom screen reader text for each radio button exclusion group and ensure that the On value for radio buttons matches the caption or is some other meaningful value.
- Images may help improve comprehension for users with some types of disabilities, However, many screen readers do not read graphics, which may decrease the accessibility of your form for users with vision impairments. If you choose to use images, provide text descriptions that describe the object and its purpose on the form.
- The tabbing order of objects on pages and their master pages is determined by the vertical coordinates of objects. Test your forms to ensure that the screen reader reads objects in the order you want.
- Be aware that client-side scripts can interfere with screen readers and keyboards if the script changes the focus of the client application. For example, the change and mouseEnter events, when used with drop-down lists or list boxes, have the potential to cause inappropriate actions. Client-side scripting should be written to avoid problems with screen readers and keyboards. Similarly, avoid scripting events that cause visual effects, such as blinking text, which may increase readability issues for users.
- If your form has a large number of objects, tabbing in Acrobat 6.0.2 can be slow. If you are creating an Acrobat 6.0.2-compatible form, adding unnamed subforms around smaller groups of form objects will add levels to the logical structure and fix this problem.

  Form authors and users should be aware of these known issues between Acrobat and screen readers:

- When users type into any field, including password fields, screen readers read back each keystroke.
- Whenever a message box opens, for example to report a validation error, the form loses focus. Pressing Tab again moves the user back to the first field in tabbing order.
- The screen reader will read all text as if it were in the language of the form's Default Locale setting.
- To display accessibility tags in forms with a flowable layout in Acrobat, you must run the screen reader before opening the form in Acrobat.

RELATED LINKS:

Design tips for authoring forms for screen readers

Create a PDF form with accessibility tags

Accessibility properties in the Accessibility palette

Turn off screen reader text for an object

Techniques for testing form accessibility

To ensure that your forms are accessible to a wide variety of users, you should test them with a variety of assistive technologies. You can test your forms simply and inexpensively using the techniques described in this section.

Ensure that the form can be filled using only the keyboard. Be sure to fill the entire form and test all fields and buttons. As you complete the form, determine whether improvements are required based on your answers to the following questions:

- Are there any operations that cannot be performed?
- Are any operations awkward or difficult to perform?
- Are keyboard mechanisms well-documented?
- Do all controls and menu items have underlined access keys?
- Demo versions of screen reader software can be downloaded free from the Internet. To test screen reader results, turn your monitor off and use only the screen reader to navigate and fill the form. Because you are the form author, your familiarity with the form may make it difficult to determine if the information read by the screen reader is sufficient and makes sense. If possible, have someone else test your form in this way.
- Demo versions of screen magnification software are also available for testing from the Internet.
- Speech-to-text software is available at a nominal cost from local computer stores. Test the form by using voice input only.

  Many users with vision impairment rely on high contrast between text and the background to read the form. Microsoft Windows has a high contrast color scheme that provides a display similar to what many users with vision impairment will be using to complete your form. To set your display to high contrast mode, enable the feature through Accessibility Options in the Windows Control Panel. If you enable the high contrast mode, large icons appear in the Object palette. In addition, if you drag an object to the canvass, the label and value of the field appear in high contrast mode. In high contrast mode, you can use the short-cut keys Ctrl + and Ctrl - to adjust the pagination and to switch between sub types.

  As you complete the form in this mode, determine whether improvements are required based on your answers to the following questions:
- Do parts of the form become invisible, unrecognizable, or difficult to use?
- Do any areas continue to appear black on a white background?
- Are any elements improperly sized or truncated?

14.12. Making objects accessible

Specify custom screen reader text for an object

Use the Accessibility palette to define custom screen reader text for an object.

To test your form, ensure Acrobat is set up to Read Form Fields in the Preferences (Reading category) dialog box in the Edit menu.

1) Select the object on the form design.
2) Click the Accessibility palette and, in the Custom Screen Reader Text box, type the custom text.

Specify a custom tool tip for an object

Use the Accessibility palette to define a custom tool tip for an object. For most objects, tool tips appear at run time when the user hovers the pointer over the object. Tool tips appear for some read-only objects, such as a paper forms barcode object, only when a screen reader is in use.

You cannot have unique custom tool tip text and unique custom screen reader text for one object. You must choose one or the other. If you want a custom tool tip, type the custom text and either select Tool Tip from the Screen Reader Precedence list or do not include custom screen reader text. If you want custom screen reader text to be both the tool tip and the screen reader text, type the custom text and select Custom Text from the Screen Reader Precedence list.

If nothing is specified in either the Tool Tip box or the Custom Screen Reader Text box, then the screen reader uses the caption for the tool tip and for reading.

1) Select the object on the form design.
2) Click the Accessibility palette and, in the Tool Tip box, type the text you want. To create a multiline tool tip, type the first line and then use the Ctrl + Enter command to go to the next line.
3) If there is also custom screen reader text defined for this object, and you want the screen reader to read the tool tip instead of the custom screen reader text, from the Screen Reader Precedence list, select Tool Tip.

Specify a different search order for screen reader text

By default, the screen reader searches the form settings for screen reader text in the following order: Custom Text, Tool Tip, Caption, and Name. You can override this default order using the Screen Reader Precedence option.

1) Select the object on the form design.
2) Click the Accessibility palette.
3) In the Screen Reader Precedence list, select the preferred option.

   For example, if you want the screen reader to read the name instead of the caption, select Name. This also changes the search order to Name, Custom Text, Tool Tip, and then Caption.

Turn off screen reader text for an object

In certain situations, you may not want the screen to read any text for an object, not even the type of object. You can turn off text for a screen reader on an object-by-object basis.

To test your form, ensure Acrobat is set up to Read Form Fields in the Preferences (Reading category) dialog box in the Edit menu.

1) Select the object on the form design.
2) Click the Accessibility palette.

3) In the Screen Reader Precedence list, select None.

Make radio buttons accessible

When tabbing into an exclusion group, the screen reader first reads the text for the exclusion group and then the caption for the active radio button.

When a user with vision impairment tabs into a radio button, the screen reader needs to read two things:

- A general description of the purpose of the group of buttons
- A meaningful description for the purpose of each radio button

  For example, a form contains a group of radio buttons that pertain to payment. For the exclusion group as a whole, you need speak text to say "Select the method of payment". For each of the buttons, you need a value for "Cash", "Credit Card," and "Check". When the user tabs into the exclusion group and the first radio button, Cash, becomes active, the screen reader says the type of object and then says "Select the method of payment. Cash" and then it says the state (whether it is checked or unchecked).

  Designer includes options that support screen readers for defining speak text. You can use the Accessibility palette to define custom screen reader text and tool tips. You can also use object names, as specified in the Name option of the Binding tab, and captions. In most cases, the captions for the radio buttons should provide meaningful text for the screen reader.

To make radio buttons accessible using the button captions

1) In the Hierarchy palette, select the exclusion group.
2) Click the Accessibility palette and, in the Custom Screen Reader Text box, type the speak text for the group. For example, type `Select a method of payment`.
3) If the captions for each radio button provide meaningful speak text, in the Object palette, select the Binding tab and deselect Specify Item Value.

To make radio buttons accessible using a specified item value

1) In the Hierarchy palette, select the exclusion group.
2) Click the Accessibility palette and in the Custom Screen Reader Text box, type the speak text for the group. For example, type `Select a method of payment`.
3) In the Hierarchy palette, select the first radio button in the group.
4) In the Object palette, click the Field tab. In the Item area, double-click the item and type a meaningful value for the selected radio button. For example, type `Cash`.
5) Repeat steps 3 and 4 for each radio button in the exclusion group.

Make Table of Content accessible

The table of contents is the index page of the document which provides a list of topics that are discussed in the document. Each content of the table may link to the reference within the document where the topic is discussed in detail.

To make table accessible, the following tags: <TOC> and <TOCI> are used. <TOC> for the Table of Content and <TOCI> for the Table of Content Items, so that the screen reader can associate these objects properly. <TOC> and <TOCI> is defined for any type of table of contents such as a table of figures or illustrations.

To make Table of Content accessible using Accessible Palette

1) Insert Subform from the Object Library into the Design View.
2) Insert another Subform from the Object Library and position it within the first Subform.
3) Open the Accessibility Tab and select the role as TOC from the Role drop-down list. In the Role drop-down list under the Accessibility tab, the TOCI option gets enabled.
4) Select the role as TOCI from the Role drop-down list.

If the form design contains a Table of Contents created by nesting subforms, you can use the Accessibility palette to set the appropriate role for each subform. This ensures that the screen reader can announce the Table of Content items.

For example, assign the role of TOC to the top-level subform and assign the role of TOCI to the nested subforms that make up the Table of Content items.

You can select a role as Reference or Span for the nested subforms. Select a role as Reference or Span from the Role Drop-down list under the Accessibility Tab for a Subform added from the object library under TOCI role.

Make lists accessible

If the form design contains lists created by nesting subforms in a form design, you can use the Accessibility palette to set the appropriate subform role for each subform that makes up the list so that the screen reader can announce the list, the number of list items, the nesting level, and the list end.

For example, assign the role of List to the top-level subform and assign the role of List Item to the nested subforms that make up the list items.

Make images accessible

Images may help improve comprehension for users with some types of disabilities, However, many screen readers do not read graphics. For users with vision impairments, images may decrease the accessibility of your form.

If you choose to use images, provide text descriptions for all image and image field objects. When users navigate to the object, the screen reader speaks the text. Ensure that the text describes the object and its purpose on the form.

You provide text descriptions using tool tips or custom screen reader text in the Accessibility palette or text fields, captions, and object names, as specified in the Name option of the Binding tab.

In addition to providing descriptive text for images, consider these general guidelines to enhance the accessibility of images in your forms:

- Ensure that images enhance the form content without providing unnecessary detail.
- For images that use color, choose high-contrast colors for maximum readability.
- Ensure that the image size is not so small that it is difficult to read.

Make scripts accessible

As part of the form design process, a form developer can use scripts to provide a richer user experience. You can add scripts to most form fields and objects. For example, you can create simple scripts to dynamically update values on an interactive form in response to user input.

When designing scripts for accessibility, consider these general guidelines:

- Keep the form content free of visual interruptions. For example, avoid features that cause content to flicker, blink, or move.
- Ensure that pop-up windows appear only as a result of user-initiated actions. Similarly, do not allow the current focus of the form to change or content to redisplay unless initiated by the user. The focus of the form refers to the user's current view. For example, if the user is completing fields in the lower half of the form, do not allow the focus to change to the upper-left corner of the form unless the user chooses to navigate to this location.
- Users with disabilities may require more time to provide input in fields. Do not specify time-based responses for input fields.
- Be aware that client-side scripts can interfere with screen readers and keyboards if the script changes the focus of the client application. For example, the change and mouseEnter events, when used with drop-down lists or list boxes, have the potential to cause inappropriate actions. Client-side scripting should be written to avoid problems with screen readers and keyboards.

For more information about scripting, see Scripting Basics.

RELATED LINKS:

About accessible forms

Accessibility properties in the Accessibility palette

Working with the Keyboard

Color in accessible forms

Using radio buttons

Using images

Using image fields

14.13. Making tables accessible

Make simple tables accessible

Tables are an effective way to organize and present content in accessible forms. When used appropriately, a table's rows and columns provide a predictable and consistent structure for form content.

For example, when a user navigates into a body row cell, the screen reader specifies the cell location and then reads the cell content. The screen reader specifies the cell location using a combination of row and column headers or row and column numbers.

In addition to providing the location of a cell, screen readers may also specify header information, such as the content of the cell at the top of the column. Because screen readers provide information that orients the user to the location of content in the table, its layout directly affects the table's accessibility.

Tables with simple layouts are recommended. Simple tables begin with a single header row followed by the body rows.

When designing simple tables for accessibility, remember the following tips:

- The tabbing order for a table is geographic order, which is the same as for the form itself. Ensure that the table content is organized such that it makes sense when read from left to right and top to bottom.
- Most screen readers interpret the first row in a table as the header row. When reading the content of a body row cell, these screen readers first read the content of the associated header row cell. Ensure that the content in each header row cell meaningfully describes the column content.
- Avoid cells that span two or more columns, nested tables, or table sections. Screen readers have difficulty interpreting these features correctly or may not use them. For example, if a cell in a body row spans two columns, screen readers may not reference the correct cell content in the header row when reading the next cell in the row.

Navigate tables in accessible forms

In accessible forms, screen readers speak information about where the cursor is located in the table. As a user moves between the table cells in a form, the screen reader speaks the contents of the cell. When a user moves to a cell containing a nested table, the screen reader speaks the contents of the cell and the nested table.

The editable fields in a nested table are part of the tabbing order in the form. Therefore, users move between editable fields in a nested table by using the arrow keys. As the user moves in and out of a nested table, the beginning and end of the table is spoken.

NOTE: When using JAWS, the beginning and end of the nested table is not spoken in virtual cursor mode.

Result	Action
The screen reader speaks the current table cell.	Ctrl+Alt+5 (number pad)
Move horizontally between cells.	Ctrl+Alt+Left or Right Arrow
Move vertically between cells.	Ctrl+Alt+Up or Down Arrow

Make complex tables accessible

When designing tables for accessibility, we recommend that you keep the table layout simple, with one header row followed by body rows. However, some content may be best presented in a table format rather than other presentation options, but requires a more complex layout. For example, you may need to use cell spanning or more than one header to effectively convey the content.

You can create complex tables by using the table object or by combining subform objects. The table object lets you use features that are intended to help the design process, such as options for inserting and resizing columns and rows.

Depending on your design experience and preferences, you may choose to create complex tables by combining subform objects. For example, you can create one subform that includes two rows and specify this subform as the header for the table and specify another subform for the table body rows.

When using subform objects instead of table objects to create tables, the following additional steps are required:

- Set the type for each subform to Positioned. See *Subform properties in the Subform tab*.
- In the Accessibility palette, set the appropriate subform role for each subform that makes up the table. For example, assign the role of Header Row to the subform that is used as the table header. See *Accessibility properties in the Accessibility palette*.
- For rows that convey information about the table or its content but that are not considered to be part of the table, assign the subform role of None. The screen reader will read the row content.

  The features supported by the screen reader determine the information read for a complex table. For example, assume that a table includes a header row and a section with a header row. When the user navigates into a body row cell in the table section, the screen reader should read the following content, in order:

- Content from the appropriate cell in the header row for the table
- Content from the appropriate cell in the header row for the section
- Content from the selected cell

  Some screen readers may not read cells in both header rows.

14.14. Add structural navigation in forms

You can provide users with the ability to have the screen reader navigate a form by the headings. You provide this ability by placing text objects at the beginning of different sections in a form and then using the Accessibility palette to specify the Heading or Heading Level role for each text object.

When the user presses the H key in the JAWS screen reader, the screen reader moves from the current object to the next text object that is marked with the Heading role in the form, and that text object is then spoken. When the user presses keys 1 through 6, the screen reader moves from the current object to the next text object that is marked with the Heading Level role. For example, pressing 2 moves the screen reader to the next text object that is marked with the Heading Level 2 role.

14.15. Checking the accessibility of a form

The Accessibility Checker validates the accessibility of a form in Designer. The checker validates the accessibility of a form for a specific set of rules.

To run the Accessibility checker

1) Click Tools - > Macros -> Accessibility Check.
2) The rules dialog displays the set of checks that the checker performs.
   A checkbox against each check enables you to specify which checks you want to perform.
3) Click OK to run the checker.

As soon as the checker completes the validation, the PDF of the results, is displayed. The PDF provides a two-column output with each line item describing a form object and the accessibility error or warning that was found.

You can also view the results of the checker in the Designer Log palette for the form.

Accessibility checker rules

Rule	Description	Resolution
Fields without captions	Identifies fields for which captions are not specified	Select the field. Go to the **Layout** tab and specify a valid caption in the **Caption** group
Fields with no assist text	Identifies text fields for which alternate text or tooltip is not specified	Select the field. Go to the **Accessibility** tab and enter an appropriate description in the **Custom Screen Reader Text** or the Tool tip text field.

Rule	Description	Resolution
Tables with no header row	Identifies tables for which a header row in not defined	When you add a table to a form, select **Include Header Row in Table** in the **Insert Table** dialog
Images with no alternate text	Identifies images for which alternate text is not specified	Select the image. Go to the **Accessibility** tab and enter an appropriate description in the **Custom Screen Reader Text** field.
Read order and tab order diverge	Identifies if the read order and the tab order of a form are not identical. This happens if a user goes to the Tab Order tab and changes the tab order of the form	Go the **Tab Order**tab and ensure the tab order of form is identical to the read order
Primary Locale	Identifies if the Primary Locale of the form is not specified. By default, Designer uses the locale of the system where the form is created	Go to File -> Properties. In the **Defaults** tab, select a locale from the **Form Locale** drop-down list
Form Title	Identifies if a title is not specified for the form	Go to File -> Properties. In the **Info** tab, enter a valid title in the Title field
Check for presence of scripts	Warns the form author that scripts are present in the form. The author needs to check for accessibility violations	Verify that if the scripts perform any updates to the form, these updates comply with the accessibility rules
Check for repetitive hyperlinks	Warns the form author that if link text of a hyperlink is repeated more than two times in a form	-
Check for roles	Identifies if one or more roles are specified for static text objects Identifies if subforms for which the role List is specified contains list items	Select the static text object. Go to the **Accessibility** tab and specify a valid role From the Object Library add a nested Subform to the Subform. Go to the **Accessibility** tab and specify the role as List Item
Tables with no summary	Identifies tables for which no summary has been provided	Add tool-tip text or custom text for the warned table. The tool-tip text or custom text are treated as summary.

For more information about scripting, see Scripting Basics

15. Setting Security

A very important consideration for designing interactive PDF forms is ensuring that your forms and the data you gather is secure. The following functions and features in Designer provide security options for your forms:

PDF Security settings

Let you define the access capabilities users have to the PDF forms. In the Form Properties dialog box, you can specify a password that the users must enter to open and fill the form. You can also restrict various levels of access to the form. For example, you may want to allow users to fill fields and signatures but not add comments or pages to the form.

Digital signatures

Let you authenticate a signing party and ensure that a form is not altered after it is signed. To add a digital signature field to a form design, you can use a signature field or a data signature associated with a submit button. Use a signature field to protect the appearance of the form objects and the values they hold. Use a data signature to protect the integrity of the form data.

Password fields

Let you control access to a form. You can set character masking as an option to the field so that the data is not displayed on the screen. You can define whether the password is optional, recommended, or required, and you can set up messages to prompt users appropriately. All user input may be validated through scripting. You can also bind password fields to data sources.

Secure WSDL data connections

Let you use HTTP/HTTPS and message-level (SOAP) client authentication to access WSDL documents and to connect to secure web services by using a security token (embedded in the SOAP message header).

XML Encryption

Lets you apply XML encryption to form content. You can apply encryption options like specifying the content to encrypt, the encryption algorithm to use, the public encryption certificates to encrypt for, as well as the key usage criteria to use when filtering for public encryption certificates.

15.1. Importing digital IDs

When you create a secure WSDL data connection with HTTP/HTTPS client certificate authentication, you need to use a digital ID to access the WSDL document on the secure web server. The digital ID is the credential that the secure web server requires to carry out the client authentication.

A *digital ID* is a document that usually contains the owner's name and public key, the expiration date of the public key, the serial number of the ID, and the name and digital signature of the organization that

issued the ID. The digital ID binds together the owner's name and a pair of electronic keys (a public key and a private key) that can be used to encrypt and sign documents.

Before you can create a secure WSDL data connection with HTTP/HTTPS client certificate authentication, you must have a digital ID available in the Windows Certificate Store or in the Designer Digital ID Files store, ready to select when you create the connection. Client certificate authentication uses a digital ID for HTTP/HTTPS authentication, where you are prompted with the Select Digital ID dialog box. This dialog box contains a list of the digital IDs that are available in the Windows Certificate Store or in the Designer Digital ID Files store. You must select the correct digital ID to log in to the secure server. The digital ID is embedded in HTTP/HTTPS protocol header, and acts as a credential for the HTTP/HTTPS client authentication.

NOTE: By default, Microsoft® Windows® Vista™ provides increased HTTPS security in Internet Explorer 7 by adding certificate revocation checking. In addition to checking for a certificate, Windows Vista also checks whether the certificate is revoked. This additional check can cause the HTTPS client certificate authentication for Designer to fail. To disable certificate revocation checking, on the Advanced tab of the Internet Control Panel, in the Security section, deselect Check For Server Certificate Revocation.

When the HTTPS connection is first established, the HTTPS server sends its certificate to the client for verification. At this point, you must have the certificate that is needed to identify the HTTPS server's certificate installed on the client computer. If the HTTPS server certificate cannot be identified (trusted), Designer displays a warning dialog box that lets you indicate whether you want to continue and establish the connection.

You can use the two tabs in the Digital IDs dialog box (Tools > Options > Security > Digital IDs) to import digital IDs into the Windows Certificate Store or Digital ID Files store or both. The digital IDs in the Windows Certificate Store are available to all Windows applications. The digital IDs in the Digital ID Files store are available in Designer only. The Digital ID Files tab lists the digital IDs that are installed in the current user's private certificate store.

You can also add digital IDs to the Designer Digital ID Files store by using the Select Digital ID dialog box when you create a secure WSDL data connection with client certificate authentication.

Digital IDs are stored in a password-protected PKCS#12 file with the file name extension *.p12 or *.pfx.

To import a digital ID into the Windows Certificate Store

1) Select Tools > Options > Security.
2) Click Digital IDs.
3) Click the Windows Certificate Store tab and click Import Digital ID.
4) Click Next.
5) In the File Name box, type the path to the digital ID file to import, or click Browse to locate and select the digital ID file, and then click Next. Be sure to select the Personal Information Exchange (*.pfx* .p12) option in the File Of Type list.
6) In the Password box, type the password that is required to access the digital ID file.
7) (Optional) Select Enable Strong Private Key Protection.
8) (Optional) Select Mark This Key As Exportable.

9) Click Next.
10) Select Automatically Select The Certificate Store Based On The Type Of Certificate.
11) Click Next, and then click Finish. The name of the digital ID appears on the list of IDs that is available in the Windows Certificate Store.

To import a digital ID into the Digital ID Files store

1) Select Tools > Options > Security.
2) Click Digital IDs.
3) Click the Digital ID Files tab and click Add ID File.
4) In the File Name box, type the path to the digital ID file, or click Browse to locate and select the digital ID file.
5) In the Password box, type the password that is required to access the digital ID file, and then click OK. The name of the digital ID appears on the list of IDs available in the Digital ID Files store.
6) Click OK.

RELATED LINKS:

Digital IDs dialog box

To create a data connection using a WSDL file

15.2. Using XML encryption

Designer provides W3C XML standard encryption to secure the contents of forms during electronic transmission.

Applying XML encryption to form content ensures data integrity and allows information to travel securely. For example, you can use XML encryption to secure the information in financial or health care forms where the data being transmitted is private or sensitive.

You apply XML encryption to either the form data or the entire submission, including attachments. You can select encryption options like the encryption algorithm to use, the public encryption certificates to encrypt for, and the key usage criteria to use when filtering for public encryption certificates

You can use these three library objects to submit encrypted form content: Button, Email Submit Button, and HTTP Submit Button. For example, you can add an Email Submit Button to a form, and configure it to encrypt the form data when the user clicks the button to e-mail the form.

NOTE: When you add a button to a form to submit encrypted form content, ensure that the target version for the form is set to Acrobat and Adobe Reader 10.0 or later. Also, to work with forms that contain encrypted data, form fillers require Acrobat or Adobe Reader 10 or later. (See *Defaults (Form Properties dialog box)*.

To apply XML encryption to form content using a button

1) Add a Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Control Type area, select Submit. Data is submitted according to the settings in the Submit tab.
4) Click the Submit tab, and in the Submit to URL box, type the URL protocol.

To	Use this URL protocol	Example
Submit the package to an ftp site	ftp	ftp://ftp.gnu.org/gnu/GPL
Submit the package to a web server	Http	http://myserver/cgi-bin/
Submit the package to a secure web server	Https	https://myserver/cgi-bin/

NOTE: When submitting data to a URL, it is recommended that you specify an absolute target. Relative targets are interpreted relative to the user environment, which can vary from one user to the next at runtime.

5) In the Submit As list, select XML Data Package (XDP), PDF, or XML Data (XML).

   These formats initiate server-side processing. The XML Data Package (XDP) option submits the form data and optionally includes other information, such as the form design, annotations, and signatures, that is needed for Forms to later render the form at runtime. Attachments can be included with XDP files only.
6) (Optional) To apply a signature to the submitted data, select Sign Submission and then click Settings to select optional signature settings.
7) (Optional) To apply XML encryption to the form content, select Encrypt Submission, and then click Settings to select optional encryption settings.
8) (XDP only) In the Include area, select the types of attachments to include:
   - To include review comments, tool tips, and any other special tags that are required to capture screen reader text, select Annotations.
   - To include a PDF version of the form (including signatures) when it is submitted as an attachment, select PDF (Includes Signatures). Otherwise, a reference to an embedded PDF file is included.
   - To include a copy of the form design, select Template.
   - To include one or more <xdp> elements in the XDP source file, select Other. Separate the specified elements with commas. White space is optional. For example: `xci, xslt, sourceset`.
9) (XDP and XML Data) In the Data Encoding list, select one of the data encoding schemes:
   - UTF-8
   - UTF-16

- Shift_JIS
- Big5
- GBK
- KSC_5601

To apply XML encryption to form content using an email submit button

1) Add an Email Submit Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the Email Address box, type the e-mail address you want to submit the PDF to.
4) (Optional) In the Email Subject box, type a subject line for the e-mail message.
5) In the Submit As list, select PDF or XML Data (XML).
6) (Optional) To apply a signature to the submitted data, select Sign Submission in the Field tab and then click Settings to select optional signature settings.
7) (Optional) To apply XML encryption to form content, select Encrypt Submission, and then click Settings to select optional encryption settings.

To apply XML encryption to form content using an HTTP submit button object

1) Add an HTTP Submit Button object to the form design. See *To add objects to a form design*.
2) In the Object palette, click the Field tab.
3) In the URL box, type the URL that the form data will be posted to.
4) Select Encrypt Submission, and click Settings.
5) In the Encrypt Data And Submit Settings dialog box, select options as needed. See *Encrypt Data and Submit Settings dialog box*.

RELATED LINKS:

Submitting data using a button

Button properties in the Submit tab

Email submit button properties in the Field tab

HTTP submit button properties in the Field tab

Encrypt Data and Submit Settings dialog box

15.3. Using digital signatures

You can protect the integrity of forms by allowing people to use certificates to digitally sign forms. After the form is signed, the signed portion cannot be altered without invalidating the signature. The digital signature invokes a third-party signature handler that provides the required digital signature functionality.

Verifying the signature guarantees that no one tampered with the data after it was submitted. When someone signs a form, a message digest of the data to be signed is created, and a mathematical computation combines the person's private key with the specified form data and encrypts them together. The output is a *digital signature*. This digital signature contains the signed data and the certificate information associated with the person who signed the form.

When the signature is verified, the individual's public key is used to decrypt the signed data and to obtain the digest value. The new digest value is calculated against the received document by using the same algorithm of the signing process. If the two digest values do not match, this means that the data has been tampered with since the form was signed, and the verification fails.

Digital signatures also bind certificates to the signed data. The certificate included in the signature can be authenticated to validate the identity of the person who signed the data.

The individual must have a digital certificate from an appropriate certificate issuer to sign the form. Adobe signatures support the Public Key Cryptography Standard (PKCS) #7, using the RSA MD5, RSA SHA-256, DSA SHA-256, DSA SHA-384 hash algorithm.

Designer contains the following two types of digital signatures.

Document signatures

Document signatures protect the appearance of form objects and the values they hold. To create a document signature, add a signature field to the form design. You can specify whether a document signature applies to an entire form or to a collection of objects on a form. By default, it applies to the entire form. If you want the document signature to apply to a collection of form objects, the signing party must use Acrobat or Adobe Reader 8.0 or later.

If the document signature applies to a collection of objects, it ignores static objects, such as circles and rectangles. It applies only to buttons, check boxes, date/time fields, decimal fields, drop-down lists, image fields, list boxes, numeric fields, paper forms barcodes, password fields, radio buttons, other signature fields, and text fields.

To sign the form, the user clicks the signature field. If the document signature applies to a collection of form objects, the objects specified in the signature are locked and set to read-only. Locking the objects prevents form recipients from changing the object values after the document is signed.

Data signatures

Data signatures secure the form data and guarantee the data integrity during transmission. To create a data signature, add a submit button, email submit button, or HTTP submit button, and select the Sign Submission option in the Object palette. To sign the data, the signing party must use Acrobat or Adobe Reader 8.0 or later.

Data signatures can apply to the form data only or to the entire submission, including any attachments such as annotations, document signatures, or PDF documents.

To sign the form data, the user clicks the submit button. When a user clicks the button, they must provide their signing certificate so that their digital signature is applied to the form data before it is submitted.

To add a signature field

Designer provides a built-in signature field that you can add to a form so that users can sign the data they enter. By default, the signature applies to all the objects on the form. If you want the signature to apply to a collection of objects on the form, you need to create the collection and then specify the collection in the Signature tab of the Object palette. For more information see *Using signature fields*.

If the signature applies to a collection of objects, signing the document also locks the fields in the collection. If you add more than one signature field to cover different parts of the form, the first collection of objects is locked after the first signature field is signed and only the remaining fields can be filled in the form. For more information about object collections, see *Using form object collections*.

When a signature applies to a collection of objects, it protects the value in the objects. The signature can also protect the appearance of the objects in the collection so that their original appearance is restored when the form is opened. The ability to protect the appearance of the objects depends on the Preserve Scripting Changes To Form When Saved option in the Defaults tab of the Form Properties dialog box. If this option is set to the default, Automatically, the appearance of the objects is automatically protected. However, if the option is set to Manually, you will need to write a script to restore the appearance of the objects.

A signature field should not be repeatable and should not be inserted in a repeatable subform object. A form with repeatable signature fields is not valid and causes problems in Acrobat where the signature field may not appear at all.

If signature covers a collection of objects, the objects in the collection should not be repeatable, because the number of objects can change and invalidate the signature. You can either remove the repeatable objects from the collection or use a data signature instead.

To sign the form, the user clicks the signature field. To sign a collection of form objects, the signing party must use Acrobat or Adobe Reader 8.0 or later.

1) In the Object Library palette, click the Standard category and drag the Signature Field onto the form design.
2) (Optional) To apply the signature to a collection of fields on the form, click the Signature tab in the Object palette and select Lock Fields After Signing.
3) (Optional) Create and select the collection:
   – Select New/Manage Collection.
   – In the Collection List dialog box, click New, type a collection name, press Enter, and click Modify.
   – In the Collection Editor dialog box, select the objects that you want to include in the collection and click OK.
   – In the Collection List dialog box, click Close.
   – Select the collection from the list.
4) (Optional) Apply the signature to the fields inside or outside the collection:
   – To apply the signature to the fields inside the collection, select All Fields In Collection.
   – To apply the signature to the fields outside the collection, select All Fields Not In Collection.

5) (Optional) Click Settings to define optional security settings, such as a signature handler or certificate revocation list.
6) Use the palettes to format the appearance of the signature field.
7) Save the form design.

Considerations for adding multiple signature fields

When you place a signature field on a form design, the Lock All Fields After Signing option is selected by default. This means all that all form objects will be locked after the field is signed.

If you place more than one signature on the form design, and the Lock All Fields After Signing option is selected for at least one of the signature fields, all form objects, including all signature fields, will be locked after the form is signed.

To ensure that all signature fields can be signed, create a collection of form objects that includes only one of the signature field objects, and make sure that the Lock All Fields After Signing option is selected. After this signature field is signed, all of the objects specified in the collection will be locked. Keep any remaining signature field objects outside of the collection, and deselect the Lock All Fields After Signing option for each one, so that these signature fields can still be signed.

For more information about object collections, see *Using form object collections*.

To test a signature field

Use the Preview PDF tab to verify that the signature field behaves as expected. Before testing the form design, ensure that, in the Form Properties dialog box, you have set the default options for previewing an interactive form.

To sign a collection of form objects, the signing party must use Acrobat or Adobe Reader 8.0 or later.

1) Click the Preview PDF tab to view the form design as an interactive PDF form.
2) Click in the signature field.
3) In the Alert dialog box, select Continue Signing.
4) In the Data Exchange File dialog box, select a digital signature or click Add Digital ID to create a new ID.

For more information about testing forms, see *Test and troubleshoot*.

To add a data signature to a submit button

You can configure a submit button to create a data signature before submitting data so that users can sign the form data when they submit the form. Data signatures provide data integrity and authenticate the signing party during transmission. For example, you can use data signatures to secure the form data during automated transactions.

To sign the data, the signing party must use Acrobat or Adobe Reader 8.0 or later.

1) In the Object Library palette, click the Standard category and drag the Button, Email Submit Button, or HTTP Submit Button onto the form design.
2) In the Object palette, associate a data signature with the button:
   – If you are using a standard button, in the Control Type area of the Field tab, select Submit, and then select Sign Submission in the Submit tab.
   – If you are using an email or HTTP submit button, select Sign Submission in the Field tab.
3) (Optional) Click Settings to define optional security settings, such as whether the signature applies to the data only or to the entire submission.
4) Use the palettes to format the appearance of the submit button.
5) Save the form design.

For more information, see *Submitting data using a button*.

To use digital signatures in dynamic PDF forms

If you add a digital signature to a dynamic PDF form, you need to preserve the state of the form when it is signed. Preserving the form state ensures that scripting changes in a form are preserved when a form is saved. For example, if some fields are hidden when a form is signed, those fields should remain hidden when the form is saved and reopened.

The form state can be preserved automatically or manually. If it is automatically preserved, scripting changes are automatically preserved when the form is saved. If it is manually preserved, scripting changes are preserved through scripting when the form is saved. Use the `delta` script object and the `preSave` and `initialize` events to manually preserve the form state.

For more information about the `preSave` and `initialize` events, see “Events” in *Scripting Basics*.

If the form is not certified, you can preserve the form state automatically or manually. If the form is certified, you must manually preserve the form state. Automatically preserving scripting changes and restoring the form state invalidates the digital signature in a certified form.

NOTE: If the form is certified, you can use data binding instead of scripts to determine the form state.

You specify how the form state is preserved on the Form Properties dialog box.

1) To display the Form Properties dialog box, select File > Form Properties and click Defaults.
2) Select a Preserve Scripting Changes To Form When Saved option:
   – If the form is not certified, select Automatically or Manually.
   – If the form is certified, select Manually.

RELATED LINKS:

Using signature fields

Signature field properties in the Field tab

Signature field properties in the Signature tab

Defaults tab (Form Properties dialog box)

Using buttons

Collection Editor dialog box

Collection List dialog box

15.4. Using password fields

You can control access to a form using a password field and a script. The password field provides a number of options:

- A character-masking option that enables you to specify a character that masks the data the user enters. The password itself does not display in the password field
- An edit pattern, which describes the syntax for entering data into the password field at run time
- A validation pattern, which is used to validate user input at run time
- A setting that specifies the password field as required, recommended, or optional, and includes a message for prompting users to enter a recommended or required value

NOTE: The capture of information and the display of characters in the field is for visual purposes only. No attempt is made to hide the value of the field from the programming interfaces or when submitting data. If you want to secure your information, you must use a secure link (such as SSL) to encrypt the data as it is transmitted.

To add a password field in an interactive form

1) In the Object Library palette, click the Standard category and drag a Password Field object onto the form design.
2) Using the Layout, Font, Paragraph, and Border palettes, set the properties you want for the password field.
3) Add a validation pattern that will be used to validate the password and the message that will appear if the password is not a valid format.
4) (Optional) Change the default masking character.
5) Add any binding information as required.
6) Save and test the form design.

To change the password display character

The password display character that appears by default when a user enters a password is the asterisk symbol (*). You can change this symbol in the Password Display Character option of the Object palette Field tab.

1) In the Object palette, click the Field tab, select the default symbol (*) in Password Display Character, and type the replacement character; for example, type #.

To specify an input format and validation message

You can include a validation pattern that is used to validate user input at run time and a message to prompt the user to re-enter the password if the format is not a valid pattern. When a user enters a new password, it is compared it to the validation pattern. If the pattern does not match, a message appears.

TIP: You can dynamically populate a validation pattern message with a value from a data source. This option allows you to ensure that users enter the correct value in the field.

1) In the Object palette, click the Value tab, click Validation Pattern, and do the following actions:
   – In the Select Type list, select a validation pattern or, in the Pattern box, type a custom pattern.
   – In the Validation Pattern Message box, type a message that will prompt users to enter the correct value.
2) To generate an error at run time, select Error.

RELATED LINKS:

Password patterns

To dynamically populate a validation pattern message

15.5. Using PDF security options

When distributing interactive PDF forms, you may want to restrict some of the capabilities that are normally available to PDF forms. For example, in a PDF form, you can normally add comments, insert and delete pages, sign the form, copy text, and save copies of the form.

You can select the Form Properties dialog box from the File menu and set the PDF security options in the PDF Security tab to restrict access to certain capabilities. You can disable the following options:

- Printing of forms
- Ability to open a form without a password
- Insertion, extraction, or rotation of pages
- Ability to add comments
- Ability to add signatures
- Copying of text and images
- Plain text metadata

When you restrict access to any of the PDF form features, you should be aware that it may also affect the ability of screen readers to read the tagged PDF elements in your file. A *screen reader* is assistive software

technology that allows users with vision impairment or disabilities to interact with computer software. To ensure that the form is available to these users, you need to enable text access for screen reader devices.

To set a password to open a PDF form

You can set a password that a user must enter to open a PDF form. This is a two-step process; first, you specify that a password is required and then you specify the password that the user must enter when opening the form.

1) Select File > Form Properties.
2) Click the PDF Security tab.
3) Select Require A Password To Open The Document, and click OK.
4) Save the form as an Adobe PDF Form (*.pdf).
5) In the Set Document Open Password dialog box, do these tasks:
   – In the Document Open Password box, type a password.
   – In the Confirm box, retype the password.
   – Click OK.
6) Close the form design.

To restrict printing in a PDF form

You can prevent users from print a PDF form. You can specify whether the form can be printed, printed with minimal resolution, or printed with a high resolution output.

1) Select File > Form Properties.
2) Click the PDF Security tab.
3) In the Permissions area, select Use A Password To Restrict Printing And Editing Of The Document And Its Security Settings.
4) In the Printing Allowed list, select one of the following options:
   – None
   – Low Resolution (150 dpi)
   – High Resolution

To restrict editing in a PDF form

You can disable certain editing capabilities for PDF forms:

- Inserting, extracting, and rotating pages
- Adding comments
- Adding signatures
- Copying of text and images

- Enabling plain text metadata

1) Select File > Form Properties.
2) Click the PDF Security tab.
3) In the Permissions area, select Use A Password To Restrict Printing And Editing Of The Document And Its Security Settings.
4) In the Changes Allowed list, select one of the following options:
   – Inserting, deleting, and rotating pages
   – Filling in form fields and signing
   – Commenting, filling in form fields, and signing
   – Any except extracting pages
5) Select one or both of the following options:
   – Enable copying of text, images, and other content
   – Enable plaintext metadata

To enable text access for screen readers in secure forms

If you choose to set options that restrict access to PDF functions, you should be aware that these options can also affect the ability of screen readers to read the tagged PDF material in your form.

To ensure that your forms are still accessible to screen reader devices, you must ensure that the Text Access For Screen Readers option is selected.

1) Select File > Form Properties.
2) Click the PDF Security tab and, in the Permissions area, do these tasks:
   – Select Use A Password To Restrict Printing And Editing Of The Document And Its Security Settings.
   – Select Enable Text Access For Screen Reader Devices For The Visually Impaired.

16. Digital Signatures in PDF Workflows

You can use digital signatures in Adobe document services, Adobe Reader, and Acrobat workflows. Each workflow has different considerations for using the digital signature technology that influence the design and maintenance of a form application.

16.1. Design considerations

- Is more than one signature required?
- Should the signature field be located on the form design?
- Will the signature lock the data?
- Can a signature be unsigned?

16.2. Application design considerations

- Does the document need to be certified?
- Can an existing Public Key Infrastructure (PKI) implementation be used?
- Can Adobe® Acrobat® self-signed certificates be used?
- Will the digital signature be manual client-side (performed by the signing party) or programmatic server-side (performed on the server using Adobe document services)?
- What should the order of operations be in a workflow that performs multiple actions on a PDF document, such as adding usage rights, certifying, and signing?
- What mechanism will be used to store the signed PDF document so that it can be retrieved at a later date if necessary?

  For more information about electronic document security, see the Adobe technical white paper titled A primer on electronic document security: How document control and digital signatures protect electronic documents available at: www.adobe.com/security/pdfs/acrobat_security_wp.pdf

  To use the digital signature functionality, you must configure Adobe document services. (See the Adobe Document Services Configuration Guide).

16.3. How PDF documents are signed

PDF documents are signed using public-key technology. A signing party has a public key and a private key. The public key is stored in the signing party's certificate, which must be available to the recipients of the signed document in order to validate the signature. The private key is stored in a signing party's credential, which must be available at the time of signing. Information about revoked certificates is in the certificate revocation lists (CRLs) distributed by certificate authorities.

Adobe document services require access to information about credentials, certificates, and CRLs. For information about credentials, see the " Installing and Configuring Credentials" section in the *Adobe Document Services Configuration Guide*.

When you sign a document, a digest of the document contents is created using a hashing algorithm. Your private key is used to encrypt the digest, which becomes part of the digital signature. To validate the signature, a recipient must have your public key, which is used to decrypt the digest. The digest is then recalculated and compared with the decrypted document digest to make sure that the document has not been improperly altered.

16.4. How document signature fields are added

Designer provides a document signature field that you can add to a form so that users can sign the data they enter. It provides the ability to verify the form's integrity (template and data). The document signature field is available from the Standard library. When you add a document signature field, you can select fields that are locked when the user signs the form.

When form recipients click a document signature field, they are prompted to select the certificate that they will use to sign the document. A signed document signature contains information about the certificate used to sign the document, including the certificate owner's name, and the date and time when the document was signed.

You must add a document signature field in the form designs, even when the signature or certification is set by the server at run time.

16.5. How digitally signed PDF documents are submitted for validation

A PDF document that is digitally signed must be submitted to Adobe document services for signature validation. Designer provides several ways to configure a submit button. It is important to set the submit button to return the appropriate information so that Adobe document services can validate the signature.

Because the PDF document and data are signed together, if only the data is extracted, the signature is not included. Consequently, the signature is lost and cannot be verified.

Instead of using the button in the Standard library, you can use the custom button in the Web Dynpro library. The custom button is preconfigured to call the `SAPSubmit` method and performs a submit action in the Web Dynpro environment.

If you use Web Dynpro to create an application, you must create a context element for the PDFSource element so that the signed PDF file is returned to the server. If only the data is extracted, the signature is not included because the PDF document and data are signed together.

16.6. Performing multiple operations on a PDF document

If you use the PDF object (Java™) instead of Web Dynpro to create a custom application, you must consider the order of operations. If you use multiple transactions to perform multiple operations on the PDF document, you must perform the operations in the following order:

certify (server side) > apply usage rights > sign (server side)

Although you can perform these operations individually, it is more efficient to combine them into a single call to Adobe document services. To do this, specify all of the appropriate requirements in calls to the PDF object before invoking the `execute` method. Adobe document services resolves the order of operations internally when it gets the `execute` command.

Using scripts in forms that have digital signatures

When using scripts in a form, you should consider whether the script modifies the document or the data, and at what point the script is executed. A script that modifies the data after the PDF document is signed changes the status of the signed document and may invalidate the signature.

16.7. Obtaining digital certificates

To digitally sign a PDF document, a signing party must have a digital certificate, which can be acquired in two ways:

- A PKI system, such as Entrust, issues certificates to users. If no PKI system exists, users can purchase certificates form a third-party vendor, such as VeriSign.
- Acrobat can generate self-signed certificates. For more information, see “Creating digital IDs” in *AdobeReader Help*.

Signing PDF documents

After the PDF document is loaded into the browser, the signing party reviews or fills in the data and clicks the signature field to invoke the signing process.

Signing PDF documents in Adobe document services

Adobe document services can generate PDF documents that have digital signatures or digital certificates. For example, an automated process in the SAP system that creates and sends signed invoices could include an application that has a server-side signature.

Signing PDF documents in Acrobat

To sign the PDF document in Acrobat, the signing party performs the following steps:

- Click the signature field. A message appears indicating that the document has not been certified. (See *Certified documents*.)
- Click the Continue Signing button.
- Select their credential, which has been installed on their computer.
- Select a reason for signing the document from the default list of reasons or type their own reason and click Sign. A message appears indicating whether the signing process was successful.

Signing PDF documents in Adobe Reader

Adobe document services processes the form design and generates a PDF document that may be prepopulated with data. Usage rights are added to the PDF document so that a signing party can sign it in Adobe Reader®.

Validating a signature in Adobe document services

If Adobe document services will validate the signature in a PDF document, the form should include a submit button.

A valid signature has the following characteristics:

- The signing party's identity can be verified by checking the certificate to see whether it is trusted and has not expired or been revoked.
- The integrity of the document's content is confirmed by comparing the stored digest with a recalculated digest.

Comparing document versions

When changes are made to a PDF document, the changes are appended to the end of the file so that versions of the PDF document can be displayed.

After a PDF document is signed, subsequent changes are appended to the PDF file in an incremental save process:

- The `revision` field of the `SignatureInfo` object indicates which revision the signature was applied to.
- The `numRevisions` field indicates the total number of revisions in the document.
- The `MaxRevision` field indicates the highest revision number at which the document is still valid.

  If `revision` is equal to `numRevisions`, the document has not been changed since it was signed. If `revision` is less than `numRevisions`, the document has been changed since it was signed. If `numRevisions` is less than `MaxRevision`, then the document revision is valid. If necessary, you can use the `getSignedVersion` method to retrieve the state of the document at the time it was signed.

  After a document is signed, you can display a list of the changes made to the document after the last version.

To compare two versions of a signed document in Acrobat:

1) Click the Signatures tab and select the signature.
2) Select Options > Compare Signed Version To Current Version.

16.8. Single digital signature workflows

Typically, a signature field is located at the bottom of the page for a single page form and at the bottom of the last page for a multipage form. This positioning helps to ensure that the form recipient fills the form before signing it.

In some cases, the location of the signature field may be dictated by compliance rules that are determined internally or by an outside organization, such as a government agency.

Your application must either prevent unauthorized users from signing a document or include checks to ensure that the signing party is authorized to sign the document.

16.9. Multiple digital signature workflows

When multiple signatures are required, you need to determine if the signatures will be cumulative or parallel.

Cumulative signatures

The first person in the workflow signs the first part of the form. The second person reviews the data and signs the next part of the form. The second signature includes the first signature and the rest of the form.

Parallel signatures

Multiple people sign the entire form, such as a contract that requires review and must be signed by two people.

If parallel signatures are used, the `MaxRevision` value that is valid will be the same for both signatures. For more information, see *Comparing document versions*.

Locking fields

When creating a form design, you need to decide whether you will lock a group of fields that are associated with a particular signature. This is done using form object collections. If the signature applies to a collection of objects, signing the document also lock the fields in the collection. If you add more than one document signature field to address different parts of the form, the first collection of objects is locked after the first signature field is signed and only the remaining fields can be filled in the form. It is not mandatory to use collections, but they help to avoid making accidental changes to signed data that could invalidate a signature.

16.10. Signature workflow in Adobe document services

You can use Adobe document services to sign a document, validate a signature, select the collection of objects to lock, or certify a PDF document programmatically. This workflow is ideal when a large number of documents require a signature because they can be signed in a batch process.

To sign PDF documents in Adobe document services, configure the web service for secure access using the secure sockets layer (SSL). For details about this configuration, see the *Adobe document services Configuration* documentation.

16.11. Certified documents

A PDF document can be certified using Acrobat or Adobe document services. The procedure for certifying a PDF document is basically the same as the procedure for digitally signing a PDF document.

When you save a PDF document as certified, you attest to its contents and specify the types of changes that are permitted for the document to remain certified. For example, suppose that a government agency creates a form that has signature fields. When the form is complete, the agency certifies the document, enabling people to only change the form fields and sign the document. People can fill the form and sign the document, but if they remove pages or add comments, the document loses its certified status. Certifying a document helps ensure that PDF documents are not altered without the author's approval.

When you save a document as certified in Acrobat, the blue ribbon icon appears next to the digital signature and in the Signatures tab.

For information about how to configure a credential so that it is available on the server, see the *Adobe document services Configuration for SAP Web AS 7.0* document.

Acrobat and Adobe Reader can build a list of trusted identities. For more information about configuring this list, search for *trusted identities* in *Acrobat Help* or *Adobe Reader Help*.

16.12. Legal warnings

When a PDF document is certified, it is examined for various types of content that might change the appearance of the document, such as JavaScript™ actions or annotations. Such changes could mislead the signing party or recipient about what is being signed. The information generated when the document is examined is called *legal warnings* and is stored in the Legal dictionary of the PDF document catalog. For the benefit of the recipients, the author can also provide an explanation of the content that may have generated warnings.

For information about legal warnings and the Legal dictionary of the PDF document catalog, see section 8.7.4 in the *PDF Reference*.

Using the PDF object, you can programmatically get the legal warnings from a document. When you certify a document using Adobe document services, the legal warnings are always returned. If you have a certified PDF document, you can use Adobe document services to retrieve the legal warnings from it.

16.13. GeoTrust certificates

Certificate solutions allow individuals and organizations to create PDF files that certify that a trusted organization has verified the author's identity and that the document has not been altered. These solutions enable individuals and organizations to engage in more secure and reliable electronic document exchange.

GeoTrust is an Adobe certified document service partner that provides two certificate solutions:

- True Credentials for Adobe Acrobat is a solution for individual authors and organizations.
- My Credential for Adobe Acrobat is a solution for individuals

  For more information, see Certified Document Services from GeoTrust at: www.geotrust.com/cds/index.htm

  If you use a certificate that was not issued by GeoTrust, a message appears when the recipient opens the document and they must make a trust decision. The message states that the validity of the author is not confirmed.

  A recipient does not need a user certificate to validate a certified document.

16.14. Adobe Root Certificate Authority

The Adobe Root Certificate Authority (CA) is installed with Acrobat and Adobe Reader. You can access it by selecting Trusted Identities > Certificates and then selecting Certificates from the Display list in the Manage Trusted Identities dialog box. Any certificate issued under the Adobe Root CA will be trusted for signing and certifying documents by default.

The Adobe Root CA is a trusted anchor and does not require a revocation check through the CRL or Online Certificate Status Protocol (OCSP).

Chain of trust

The trust chain for a GeoTrust user or corporation (server) certificate is as follows:

Adobe Root CA

An Adobe Root CA certificate is generated first. It is generated through an extremely secure process because this certificate will sign all of the certificates generated from this CA.

GeoTrust CA for Adobe

A GeoTrust CA for Adobe certificate is generated next. It is also generated through a secure process and the certificate is signed by the Adobe Root CA, which is now trusted by the Adobe Root CA.

GeoTrust (user/server) certificate

A GeoTrust user or server certificate can be generated or issued from the GeoTrust CA for Adobe. A secure process is followed to identify an individual correctly, and the certificate is signed and issued by the GeoTrust CA for Adobe. This means that an individual's GeoTrust certificate is trusted

by the GeoTrust CA for Adobe. It is also trusted by the Adobe Root CA because the Adobe Root CA signed the GeoTrust CA for Adobe certificate.

NOTE: This GeoTrust certificate can be issued to a user (My Credential used with Acrobat Professional) or a corporation (True Credentials used with AEM forms security).

This process is known as the *chain of trust* or *path of trust.* Checking the path of trust it is called *walking the chain of trust*.

16.15. Creating and printing RFID labels

This section explains how to create and print Radio Frequency ID (RFID) labels by using Designer and a Zebra printer.

For more information about this example, see the sample form design, data, and PDF file (fit_large_-graphic.*) installed in the ...\EN\Samples folder.

Please keep the following points in mind when creating and printing RFID labels:

- You cannot create your own EPC code.
- You can place only one RFID object on a form because the output can only be to a single RFID chip on the label.

Before you start

You will need the following items before you start:

- Designer 8.1 or later
- A Zebra printer that is capable of generating RFID output, such as a Zebra 110 XiIII plus printer
- Documentation for the Zebra printer, which is available at the Zebra Printing Solutionsfor Business Requirement.
-
- FID labels available from the printer manufacturer
- XDC Editor. The XDC Editor is available for downloading at SAP Community Network.
-
- The only supported output format for RFID is ZPL, which is for Zebra printers. RFID output cannot be rendered in PDF, PostScript, or PCL. This also means that you cannot preview the RFID output in Designer.

Create and print RFID labels

Here are the basic steps to creating and printing RFID labels:

- Add the RFID object to the form design.
- Set the data length of the object.
- Bind the object to the XML data.

- Configure the XDC file.
- Map the XDC file to an SAP print device.
- Verify the printed RFID label.

Step 1 Add the RFID object to the form design

1) In the Object Library palette, click the Barcodes category and then click the RFID object.
2) Click in the area of the form where you want to place the barcode.

Step 2 Set the data length of the object

1) See the printer manual to determine the supported RFID tag type.
2) Click the Field tab and, in the Data Length box, type the length of the data.

The data length of the RFID object depends on the type of RFID tag type you are rendering. For example, if you are using RFID tag type 4, which corresponds to EPC Class 1 96 bit, you need to generate 96 bits of data. 96 bits of data corresponds to 12 bytes, which is represented by 24 hex values. Therefore, the value of the data length you enter is 24. The following table lists the RFID tab types and their values.

RFID Tag types

Tag Type	Value
0	None
1	EPC Class 0
2	EPC Class 0 Plus
3	EPC Class 1 64-bit
4	EPC Class 1 96-bit
5	ISO 18000-06B

Step 3 Bind the object to the XML data

Bind the RFID object to the XML data. (See *To createa data connection to an XML data file*.

The data length in the XML data must match the length that you set in Step 2. In the following example, the data length of item.epc is 24.

```
<data>
<item>
<epc>3074024220403B8000000008</epc>
</item>
</data>
```

Step 4 Configure the XDC file

Using the XDC Editor, modify the following RFID options.

NOTE: The XDC Editor is available for downloading at SAP Community Network

```
<!-- ========================================== -->
<!-- RFID options -->
<!-- ========================================== -->
<option name="rfidBlockRetries">4</option>
<option name="rfidLabelRetries">3</option>
<option name="rfidTagType">4</option>
<option name="rfidTransponderPosition">160</option>
```

You may need to modify the following XDC options:

rfidBlockRetries

Controls the number of times the printer attempts to write to a particular block of a single RFID tag. The accepted values for this option is between 0 to 10. If you do not specify a value for this setting, the printer uses the default value of 0.

rfidLabelRetries

The number of labels that are attempted in case of an encode or write failure. This number is different from the rfidBlockRetries setting. The accepted values for this option are between 0 to 10. If you do not specify a value for this setting, the printer uses the default value of 3.

rfidTagType

Controls the encoding type of the data in the RFID label. The RFID readers should have the same tag type as the RFID printer. The accepted values for this option vary from printer to printer. (See your printer documentation for this information.) Generally, this option uses values between 0 to 5.

rfidTransponderPosition

Represents the distance of the microchip on the label from the top. Specify this value in dots. The accepted values are between 0 to length of the label. If the microchip is located at the beginning of the label, set this option to 0.

Step 5 Map the XDC file to an SAP output device

On your SAP system, run transaction RSPO0022, which prompts you to upload the modified XDC file. This XDC file should be mapped to the SAP output device, which corresponds to your Zebra printer.

Step 6 Verify the printed RFID label

After you print the RFID label, you can use the built-in RFID reader to verify that the original data appears on the RFID label.

17. Creating and printing RFID labels

This section explains how to create and print Radio Frequency ID (RFID) labels by using Designer and a Zebra printer.

For more information about this example, see the sample form design, data, and PDF file (fit_large_-graphic.*) installed in the ...\EN\Samples folder.

Please keep the following points in mind when creating and printing RFID labels:

- You cannot create your own EPC code.
- You can place only one RFID object on a form because the output can only be to a single RFID chip on the label.

17.1. Before you start

You will need the following items before you start:

- Designer 8.1 or later
- A Zebra printer that is capable of generating RFID output, such as a Zebra 110 XiIII plus printer
- Documentation for the Zebra printer, which is available at the Zebra Printing Solutionsfor Business Requirement.
-
- FID labels available from the printer manufacturer
- XDC Editor. The XDC Editor is available for downloading at SAP Community Network.
-
- The only supported output format for RFID is ZPL, which is for Zebra printers. RFID output cannot be rendered in PDF, PostScript, or PCL. This also means that you cannot preview the RFID output in Designer.

17.2. Create and print RFID labels

Here are the basic steps to creating and printing RFID labels:

- Add the RFID object to the form design.
- Set the data length of the object.
- Bind the object to the XML data.
- Configure the XDC file.
- Map the XDC file to an SAP print device.
- Verify the printed RFID label.

Step 1 Add the RFID object to the form design

1) In the Object Library palette, click the Barcodes category and then click the RFID object.
2) Click in the area of the form where you want to place the barcode.

Step 2 Set the data length of the object

1) See the printer manual to determine the supported RFID tag type.
2) Click the Field tab and, in the Data Length box, type the length of the data.

The data length of the RFID object depends on the type of RFID tag type you are rendering. For example, if you are using RFID tag type 4, which corresponds to EPC Class 1 96 bit, you need to generate 96 bits of data. 96 bits of data corresponds to 12 bytes, which is represented by 24 hex values. Therefore, the value of the data length you enter is 24. The following table lists the RFID tab types and their values.

RFID Tag types

Tag Type	Value
0	None
1	EPC Class 0
2	EPC Class 0 Plus
3	EPC Class 1 64-bit
4	EPC Class 1 96-bit
5	ISO 18000-06B

Step 3 Bind the object to the XML data

Bind the RFID object to the XML data. See *To createa data connection to an XML data file*.

The data length in the XML data must match the length that you set in Step 2. In the following example, the data length of item.epc is 24.

```
<data>
<item>
<epc>3074024220403B8000000008</epc>
</item>
</data>
```

Step 4 Configure the XDC file

Using the XDC Editor, modify the following RFID options.

NOTE: The XDC Editor is available for downloading at SAP Community Network

```
<!-- ========================================== -->
<!-- RFID options -->
<!-- ========================================== -->
<option name="rfidBlockRetries">4</option>
<option name="rfidLabelRetries">3</option>
<option name="rfidTagType">4</option>
<option name="rfidTransponderPosition">160</option>
```

You may need to modify the following XDC options:

rfidBlockRetries

Controls the number of times the printer attempts to write to a particular block of a single RFID tag. The accepted values for this option is between 0 to 10. If you do not specify a value for this setting, the printer uses the default value of 0.

rfidLabelRetries

The number of labels that are attempted in case of an encode or write failure. This number is different from the rfidBlockRetries setting. The accepted values for this option are between 0 to 10. If you do not specify a value for this setting, the printer uses the default value of 3.

rfidTagType

Controls the encoding type of the data in the RFID label. The RFID readers should have the same tag type as the RFID printer. The accepted values for this option vary from printer to printer. (See your printer documentation for this information.) Generally, this option uses values between 0 to 5.

rfidTransponderPosition

Represents the distance of the microchip on the label from the top. Specify this value in dots. The accepted values are between 0 to length of the label. If the microchip is located at the beginning of the label, set this option to 0.

Step 5 Map the XDC file to an SAP output device

On your SAP system, run transaction RSPO0022, which prompts you to upload the modified XDC file. This XDC file should be mapped to the SAP output device, which corresponds to your Zebra printer.

Step 6 Verify the printed RFID label

After you print the RFID label, you can use the built-in RFID reader to verify that the original data appears on the RFID label.

18. Creating Forms for Optimal Performance

Many factors affect performance when rendering and using forms, including the type of form. Non-interactive forms can be rendered more quickly than interactive forms.

Other factors that affect performance include the fonts used in the form and the number, type, and complexity of objects. For forms that are bound to data sources, the structure of the data also affects performance.

Designer provides many options for designing forms. Often, you can produce the same results on a form by using different design options. When making design choices, you should consider the effects on performance of each option.

18.1. General design considerations for performance

While many factors affect performance, one of the primary ways to optimize performance is to keep the file size of the forms as small as possible.

The file size of forms affects performance when the server renders forms, when transferring forms over slow communication channels, or when storing or retrieving forms from near-line storage.

Consider these general tips to help minimize the file size of forms:

- Avoid using fonts that must be embedded, particularly for field objects. Embedded fonts cannot be subset.
- Use embedded images instead of linked images. By default, images are linked. To know more about the differences between embedding versus linking images, see this article.
- Use a compressed graphic format such as JPEG or GIF.
- Use Palette or Monochrome images for color. Avoid using 24-bit color formats.
- Be aware that generating accessible forms affects performance.
- Use only the minimum number of objects in forms and keep objects as simple as possible.

Consider these general tips to improve the performance of interactive forms:

- For forms that submit data to Forms, choose the XML Data (XML) format for the data. See *About submitting data using a button*.
- Enable form caching to increase the performance of the rendering of a form. When a form is cached, the data is merged into a pregenerated presentation. Forms whose layout adjusts to accommodate data can always be cached. Forms that have a fixed layout can also be cached, but some restrictions apply. For information about the restrictions, see *Form caching*. For Forms to cache forms that have a fixed layout, you must select the form caching option in Designer for each form you create.

Consider these general tips to improve the performance of non-interactive forms:

- Avoid validation constructs and other interactive form features, such as data entry validations and edit masks.

These features cause Forms to render interactive forms. In general, non-interactive forms have a smaller file size than interactive forms and can be rendered more quickly.

If the form is intended for printing, interactive features provide no added benefit. For example, Designer can generate validation scripts for particular objects for forms that are bound to XML schemas. For interactive forms, these validation scripts ensure that data entered by the user complies with the schema specification. For example, a validation script may check that a value provided for a numeric object is an integer between 4 and 9. Non-interactive forms do not accept user input and, therefore, the validation scripts are not used. Turning off the generation of these scripts improves performance by eliminating the processing to execute them. See To set field generation options.

- Batch forms for processing.

  When rendering multiple instances of a single form, such as bills for different customers, gather the data for all the instances in a single data file and render the individual forms from this file. This approach is more efficient than processing forms individually. However, when batching different forms, group forms before rendering them. For example, if Invoices and Packing Lists are run at the same time, group and run all the Invoices, followed by all Packing Lists.

RELATED LINKS:

Optimizing performance for images

Using images

Performance considerations for accessible forms

Tips for reducing the number of objects

Tips for reducing the complexity of objects

18.2. Optimizing performance for objects

RELATED LINKS:

General design considerations for performance

Subform properties in the Subform tab

Text field properties in the Field tab

Performance considerations for importing forms

Formatting captions

To merge text objects

Border properties in the Border palette

Choosing fonts for performance

Data binding options for performance

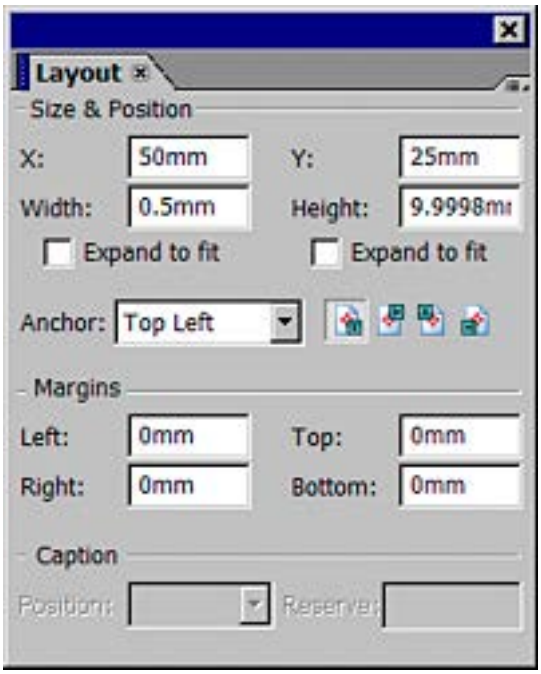
Optimizing performance for images

Images require additional processing time when the form is rendered. If you choose to use images, use compact file formats and the smallest possible image dimensions to minimize the file size. Size the

images to the intended dimensions before adding them to the form. Avoid resizing images after adding them to the form.

Ensure that corners are square and lines are exactly horizontal and vertical to optimize performance when the form is rendered. Lines that appear to be exactly vertical or horizontal may have a slight slope.

You can check the slope of horizontal and vertical lines in the Layout palette. The Height option represents the slope of a horizontal line and the Width option represents the slope of a vertical line. In the following example, a vertical line has a small slope of approximately 0.5 millimeters. A width of 0mm makes the line exactly vertical.



Optimizing performance for discrete option choices

Several objects provide discrete option choices: check boxes, list boxes, and radio buttons. List boxes and check boxes are rendered more quickly than radio buttons.

Optimizing performance for subforms

Designer distinguishes boilerplate or static objects, such as text, lines, and images, from objects that contain variable content, such as text fields and image fields.

For example, interactive forms can include text labels that provide information to the user and text field objects that capture information from the user.

A subform is a type of object that contains content. Subforms are often grouped based on the structure of the data that is bound to each subform. The ability to bind repeating, optional, and conditional data groups to subforms reduces the risk of design errors that may occur if you use scripts to achieve the same results.

For example, you can create a subform that includes objects for repeating data groups. When rendering the form, Forms creates as many instances of the subform as necessary to represent all the data groups.

When using subforms, keep the following in mind for optimal performance:

- Repeating and nested subforms require additional processing to render the form. Avoid using them unless they offer better performance than alternative options. For example, a repeating subform may eliminate the need for a number of repeating objects.
- Allowing page breaks in subforms causes additional processing, even if Forms does not apply page breaks. For example, the location, size, or content of a subform may prevent a page break. To optimize performance, turn off page breaks in subforms. By default, page breaks are allowed.

Optimizing performance for text fields

By default, entries in text field objects are formatted as plain text. You can change this option to allow rich text formatting.

However, for plain text entries, the font, style, and size used at design time are applied to user input. The form can be rendered more efficiently by using the text properties used in the form design.

For rich text entries, users can customize the font style and size for text entered in the object. Although rich text entries may be useful in some situations, such as for narrative information, the extra attributes decrease performance.

Performance of objects on master pages versus pages

Objects on master pages require additional processing time when Forms renders the form. If possible, place objects on pages.

Tips for reducing the number of objects

The file size of a form is affected by the number of objects on the form. Consider these general tips to minimize the number of objects on forms:

- When creating a box, use a single rectangle object instead of joining four individual lines.
- When creating a border for a field or subform object, use the object's border attributes instead of creating a separate box.
- When creating backgrounds for objects, use a background fill instead of creating a separate shaded box object.
- For captions, use an object's caption property instead of using a separate text object.
- Combine multiple text objects into a single text object. This strategy is particularly useful after importing forms into Designer.

Tips for reducing the complexity of objects

The file size of a form is affected by the complexity of the objects on the form. Simple objects can be processed more efficiently than complex objects. Consider these general tips to minimize the complexity of objects:

- Keep the number of fonts, styles, and sizes to a minimum to take advantage of cached font information.
- Use fixed objects instead of dynamic objects that shrink and grow, if possible.
- Avoid building objects on top of each other, especially more than three layers deep.

18.3. Structuring data for performance

RELATED LINKS:

Working with Data Sources

Subform properties in the Binding tab

Tips for reducing the complexity of objects

Choosing fonts for performance

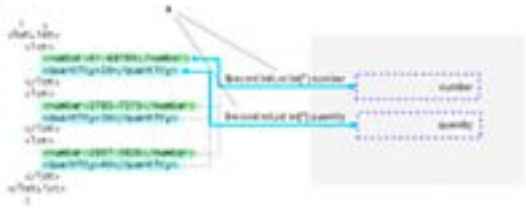
Data binding options for performance

For forms that are bound to data sources, you can associate a data element with a form object, such as a field or a subform. This association is called data binding. You can choose objects that shrink or grow depending on the amount of data they display.

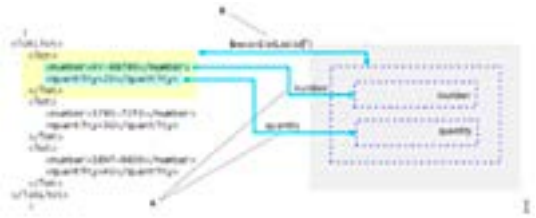
Many data binding definitions can produce the same results when a form is rendered. However, the more obvious the relationship between the form object and its data element, the more efficiently Forms can process the data binding.

The following diagram shows a field binding example where two field objects in the form are bound to data elements. The binding is ambiguous because multiple instances of the field objects and their data exist. More processing is required to match the nth occurrence of each object to the nth occurrence of the corresponding data when rendering the form.

Field bindings(number and quantity)



The following diagram shows the use of subform binding to group objects and thereby simplify the processing to render the form. The subform is bound to the repeating group in the data, eliminating the ambiguous binding. The binding is evaluated for the first group and does not need to be re-evaluated for subsequent repetitions.



A.

Field bindings(number and quantity)

B.

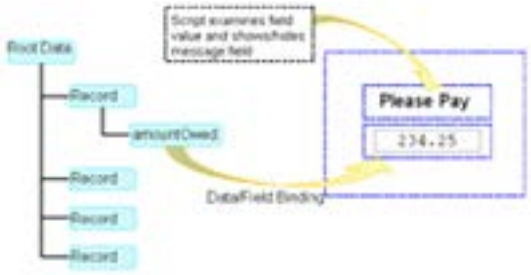
Subform binding

TIP: You can use subforms to group and organize objects without binding the subform to a data element. To prevent Forms from searching for a data element for the subform when merging data, change the subform's data binding type from the default value of Normal to None.

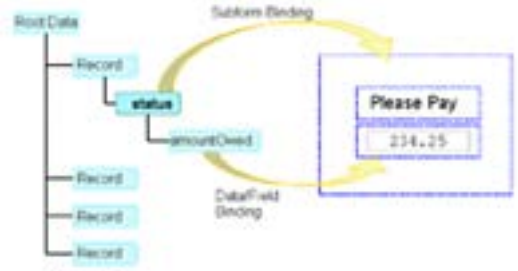
Evaluating data structure for performance

Selective field display is one way to demonstrate how data structure affects performance.

Assume that we want to display a message only if the value of the amountOwed data element is greater than zero. One approach is to use a script to examine the value of amountOwed and alter the show/hide property of the message field, depending on the value of the data element.



Alternatively, you could bind the data to a subform. Binding data to a subform is more efficient than binding data to a field, although both methods produce the same result. Here, we add the status element to the data structure and bind this element to a subform. The subform includes the message field and the`amountOwed` field. With the minimum count of the subform set to zero, the message appears only if the status element exists.



In the data context, the decision to add the status node to the data file should be done at generation time, only if the value is greater than 0. This technique moves the logic to determine whether a subform is required in the data context design.

18.4. Choosing fonts for performance

When you are choosing the fonts to use when creating a form, it is important to remember that not all of the fonts that are installed with Designer are installed with Acrobat and Adobe Reader. Consequently, when a user opens a form in Acrobat that uses unavailable fonts, the missing fonts are substituted, which can cause the layout of the form to change from the original design in Designer. To prevent this effect, either form users can install the missing fonts on their computers or the form designer can select the Embed Fonts option to embed the fonts into the saved PDF form.

However, forms with a smaller file size can be rendered more quickly than forms with a larger file size. To minimize the file size, you must keep the number of fonts to a minimum and avoid using fonts that must be embedded to prevent font substitution.

Consequently, the following fonts are always available in Acrobat and Adobe Reader, and do not need to be embedded to prevent font substitution:

- Courier Std, Courier Std Bold, Courier Std Bold Oblique, Courier Std Oblique
- Minion Pro Bold, Minion Pro Bold It, Minion Pro It, Minion Pro Regular
- Myriad Pro Bold, Myriad Pro Bold It, Myriad Pro It, Myriad Pro Regular
- Symbol (Type 1)

The following fonts are always available in Acrobat; however, they are only available in Adobe Reader when the corresponding Language Package is installed:

- Adobe Arabic/Hebrew/Thai with all 4 styles (Bold, BoldItalic, Italic, and Regular). Available in Adobe Reader when the Extended Language Package is installed
- Kozuka Gothic Pro-VI M (KozGoPro VI-Medium.otf) and Kozuka Mincho Pro-VI R (KozMin-ProVI-Regular.otf). Available in Adobe Reader when the Japanese Language Package is installed

To ensure that a form design looks the same across PDF, PCL, and PostScript output devices, you should use these fonts:

- Courier
- Arial
- Times New Roman

For form designs that are rendered only as PDF, use any of the fonts listed above that are included with Adobe Reader to optimize performance.

RELATED LINKS:

Font palette

Evaluating data structure for performance

Performance considerations for importing forms

18.5. Design considerations for printing

When a form is sent to print, both the form and the data are loaded into memory and the resulting layout also builds in memory. This means that there is a limit to the size of the final output.

The amount of data and the number of pages in a form are the most significant factors that affect memory usage; however, there are other factors in a form that can also affect memory usage. Keep the following factors in mind when designing forms for print.

- Avoid using choice subforms.
- Use a caption in a text field to label a field instead of static text.
- Avoid using rich text.
- Avoid overuse of subforms and nested subforms.
- Avoid the overuse of rectangle and line objects.

18.6. Performance considerations for importing forms

You can import existing forms into Designer. During the import, unnecessary objects may be created.

For example, when importing a PDF form, one text object may be created for each line or word in the original form. The number and complexity of objects increases the file size of the form and decreases performance when rendering the form.

Designer provides options to reduce the number of unnecessary objects created. When importing the form, you can choose the Create an Interactive Form with Fixed Pages option or the Create an Interactive Form with a Flowable Layout.

Each option produces different results in terms of the number and complexity of objects on the resulting form. You may find it useful to import the form once using the Maintain Editability option and once using the Preserve Appearance option. You can produce the final form by combining the most appropriate elements from each resulting form.

After importing the form, you can reduce the number of objects on the form by combining related text objects. For example, you can select each individual text object that makes up a sentence and combine them into a single text object.

RELATED LINKS:

Importing PDF files

To merge text objects

Tips for reducing the number of objects

Tips for reducing the complexity of objects

Choosing fonts for performance

Performance considerations for accessible forms

18.7. Performance considerations for accessible forms

When designing accessible forms, you can define text that is read by screen readers. You must create a tagged PDF form so that the screen reader can read the text. You do this by generating accessibility tags when saving the form design as a PDF file.

In Designer, the default behavior is to create tagged PDF forms.

The tagging applied for accessibility increases the file size, which reduces performance. To help offset the effects on performance for accessible forms, choose design elements carefully.

For example, for discrete option choices, choose list boxes or check boxes. These objects can be rendered more quickly than radio buttons.

RELATED LINKS:

To create a PDF form withaccessibility tags

Performance considerations for importing forms

Performance considerations for scripts

18.8. Performance considerations for scripts

In general, scripts increase the processing required to render forms. Before using scripts, determine if a better performance alternative is available. For example, character masks offer better performance than scripts, with the same results.

You can use JavaScript or FormCalc for calculations and validations in forms. If possible, avoid using validation scripts. These scripts affect performance because of the time required to process them.

If you choose to use scripts, FormCalc usually provides better performance than JavaScript for the following reasons:

- FormCalc executes simple calculations and validations more quickly.
- FormCalc interprets the XML Form Object Model syntax directly, which means it can evaluate reference syntax expressions more efficiently.

You select the script language using the Script Editor.

RELATED LINKS:

Performance considerations for accessible forms

Performance considerations for XML Form Object Model expressions

18.9. Performance considerations for XML Form Object Model expressions

The Adobe XML Form Object Model uses an approach to building forms that distinguishes among form layout, the form design, and the form data.

In a form design, reference syntax expressions associate objects with values, objects, and properties within the tree hierarchy of a Document Object Model (DOM). Reference syntax expressions are usually written into form logic, in validations, calculations, or other business rules.

Consider these tips to reduce the processing time required for resolving reference syntax expressions:

- Keep expressions as simple as possible.
- Avoid using ".." (double periods) in expressions.

In addition, after evaluating reference syntax in a script, store the result in an object variable. This result can be retrieved if the reference syntax is used again. Reusing a result is more efficient than re-evaluating the expression.

In the following example, the `resolveNode` method is used once, and the resulting object reference is saved and reused to manipulate various related objects. This script can be processed more efficiently than one that uses a reference syntax expression for each object and uses the `resolveNode` method on each one.

```
var oSubform = xfa.resolveNode("xfa.form.form1.Subform1");
if (oSubform.all.length< 3) {
  Subform.SubformUpButton1.presence = "invisible";
  Subform.SubformLabel1.presence = "invisible";
  Subform.SubformUpButton2.presence = "invisible";
  Subform.SubformLabel2.presence = "invisible";
 }
```

RELATED LINKS:

Performance considerations for scripts

Resolving log messages for optimal performance

18.10. Resolving log messages for optimal performance

When you save or preview the form design using the Preview PDF tab, a variety of errors appear in the Report palette. Some of these errors appear in the Log tab. Generating log messages reduces performance. Try to resolve as many of these messages as possible.

For example, you may be able to resolve messages about font substitutions by ensuring that the appropriate fonts are available. Some messages are generated even when forms are working properly and you may not be able to resolve them.

As you work in the form design, other messages, such as target warning and scripting errors, generated by Designer appear in the Warnings tab of the Report palette. Although the warnings and error messages

that appear in the Warnings tab do not affect performance, they need to be fixed before you deploy the form.

You can check for messages in the Designer Report palette. You can also see these messages in the Forms log files. For more information about the messages in the Report palette, see *Addressing warning messages in the Report palette*.

You should always review both types of generated messages. Even if Designer does not generate messages, the Forms environment may be significantly different from your desktop environment.

The following list provides examples of common messages generated when Forms renders the form:

- "Fonts ZaDb was not found. It was degraded to Myriad Pro."

  The form uses a font that is not available to Forms and an available font was substituted. If the form appears correctly using the substitution font, you can eliminate this error by changing the form objects to use the substitution font. You can also resolve this error by installing the missing font for Forms.

- "Script failed (language is JavaScript; context is...)"

  The script cannot execute because of scripting errors. If the script does not produce results in the form, this error may not be detected when the form is tested. However, generating the error affects performance.

  The following example shows a script error:

  ```
  script = function testFractionDigits(maxDigits)
  {
     index = this.rawValue.lastIndexOf(".");
     if (index >= 0)
     return (((this.rawValue.length1)index) <= maxDigits);
     return true;
  }
  ```

  In this example, index is an undefined variable. To resolve this error, add `var` in front of the first instance of `index`.

- "Fonts used in fields cannot be subset. Font ArialMT not subset and will be embedded."

  A font cannot be subset. To resolve this error, use a standard font to eliminate font subsetting. This change also reduces the size of the file produced.

- "ImageField access attribute should explicitly be set to 'nonInteractive'. Field will be drawn as boilerplate."

  An image field object is rendered as non-interactive in Acrobat 6.0.2 Professional and Acrobat 6.0.2 Standard and Adobe Reader 6.0.2 forms. To resolve this message, convert the object to a static image object.

- "Invalid layout attribute on subform AccountDetails. Using default."

  A subform has a row layout type that should be nested inside a table parent, but it is not enclosed in a table. To resolve this error, move this subform into a table or change the subform type to Positioned Content or Flowed Content.

RELATED LINKS:

Performance considerations for XML Form Object Model expressions

Addressing warning messages in the Report palette

19. Considerations for Creating Forms for Server Processing

19.1. Creating forms for AEM forms

Forms generator enables organizations to extend their intelligent data capture processes by deploying electronic forms in PDF, HTML, and Flash format over the Internet. Forms also retrieves form data from central repositories and merge it with the specified form. End users can access online forms without downloading any additional software, fill them online, or save them to fill offline.

Forms offers the following key features:

- Renders PDF, HTML, or Adobe Flash-based interactive forms. Enables form data integration, which is the ability to import data into and extract data from PDF forms.
- Supports guide rendering.
- Supports document encryption.
- Performs form assembly.

For more information about Forms, see *ConsiderationsforCreatingFormsforServer Processing* .

RELATED LINKS:

How Forms processes captured data

How Forms renders forms

How Forms prepopulates forms

19.2. How Forms processes captured data

You can use interactive forms to capture user-entered data. When you provide a Submit button in the form, you can enable users to upload the entered data to Forms.

When a user clicks a Submit button at run time, the data is submitted to Forms. If set up to do so, and providing no validation errors occur, Forms can remerge the data with the form and return it to the user. Alternatively, Forms may be set up to forward the data to a custom application for further processing.

It is important to understand how to define the properties of a Submit button to meet the requirements of the custom application because data can be submitted to Forms in a number of formats.

RELATED LINKS:

Using buttons

Submitting data using a button

To validate user input

19.3. How Forms renders forms

Forms can render form designs as PDF, HTML, or Flash-based interactive forms.

When form designs are deployed to Forms, rendered forms can be distributed to users directly through a web-based user interface. When a user requests a form, Forms applies an appropriate transformation of your form design to render the form.

For example, if the user asks for an interactive PDF form, Forms uses the PDFForm transformation to render the form.

RELATED LINKS:

Form transformations

Saving a form design: XDP or PDF

19.4. How Forms prepopulates forms

Forms can prepopulate a form with data from an external data source while the form is being rendered.

For some simple applications, by setting up data binding in Designer, you may be able to create forms that work with external data without having to write a script. Binding alone can usually be used to obtain data from a WSDL data source or an OLEDB database.

Forms provides support for simple data binding. Simple binding enables you to bind a single value to a field.

RELATED LINKS:

Binding fields to a data source

Considerations for testing form designs with data

Prepopulating form fields with data

19.5. Designing forms for Forms

The process of creating a form design for Forms is identical to creating any other type of form design. However, when you create a form design for Forms, the same form design can be used to render PDF or HTML forms.

Using Forms, you can post PDF and HTML forms on the Internet or an intranet. When Forms renders an HTML form, the form typically opens in a web browser. When Forms renders a PDF form, the form may open directly in Acrobat or Adobe Reader, depending on the settings of Acrobat or Adobe Reader on the user's computer.

When you distribute forms in a web environment, you must consider the limitations of the environment. For example, some of the features that you can assign to form designs by using Designer are not supported in a web environment, and different web browsers display the same objects in different ways.

Furthermore, some form capabilities are not directly supported by web browsers or HTML technology. The full range of form capabilities is available only when a PDF form is opened using Acrobat or Adobe Reader.

For information about the properties supported by different web browsers, see *Transformation Reference*.

Considerations for creating form designs for Forms

If you will be using the same form design to render PDF it is important to understand that some behavioral differences exist between the two types of output.

To create a single form design that lessens these behavioral differences, follow the guidelines outlined below:

Consult the Transformation Reference

Because PDF forms can be viewed by using Acrobat or Adobe Reader, the form supports the full range of object properties that you define in the form design. If you will be distributing HTML forms based on the same form design, some client applications (for example, web browsers) do not provide the same level of support for individual object properties. The *Transformation Reference* will help you to determine how objects behave in a particular client application. When creating the form design, try to work around any limitations in the client applications by finding ways to implement the form without relying on unsupported object properties. See *Formtransformations*.

Enable form caching

Form caching can increase the performance of forms when rendered. In addition, if you are designing a form that has a fixed layout and want to render the form in HTML, you must enable form caching. See *Formcaching*.

Include a layout that works for both PDF forms and HTML forms

When an HTML form is rendered, page sizes (required for paginating PDF forms) have no meaning. Because a form that has a flowable layout can expand on an infinitely long HTML page, it is important for you to avoid adding footers on the master page. A footer beneath the content area on a master page could overwrite HTML content that flows past what would otherwise (in a PDF form) be a page boundary.

Provide unique names for fields, exclusion groups, and subforms

For HTML output, all fields, exclusion groups, and subforms must have unique names. This will prevent possible data merge issues.

Consider accessibility

Read the section about creating accessible forms and use the guidelines to build accessibility into your form design. See *Aboutaccessibleforms*.

Determine where to run scripts that exist in the form design

By default, scripts are run on the client. If the scripts that you include in a form design should be run on the server, or on the client and server instead, you may have to change or override the default setting. See *ConsiderationsforCreatingFormsforServer Processing*.

Preview the form

Periodically preview the form by using Designer (for PDF forms) or the target client application (for HTML forms) to troubleshoot problems early in the design process. See *Topreviewandtestformsin the Preview PDF tab*.

Test the form design with sample data

If Forms will be merging forms with data, use test data to thoroughly test your form designs before you make finalized versions available to Forms. See *Topreviewaformusingsample data*.

Consider web browser limitations

Some web browsers have limited capabilities. It is a good idea to consider the limitations of the lowest common denominator and design your forms accordingly. See *Workingaroundwebbrowser-limitations*.

RELATED LINKS:

Client- and server-side processing

Previewing an HTML form

Specifying the format for submitting data

Considerations for testing form designs with data

Saving a form design: XDP or PDF

Copying form designs to the local network

Additional requirements for submitting form designs to Forms

Before you can submit form designs to Forms, some additional work needs to be completed:

- The developer of the custom application must define the requirements of the application. The file format of your form design (XDP or PDF) depends on these requirements.
- Transformation options must be defined in the Forms API to support the required transformations. The developer of the custom application will set the options based on input from you.
- If you will be using a signature object in a PDF form design, the developer of the custom application has to integrate a third-party solution to support digital signatures.

RELATED LINKS:

Saving a form design: XDP or PDF

Copying form designs to the local network

Specifying the format for submitting data

When you create a form, you can specify the format in which to submit its data. You specify the data submission format in Designer in one of two ways:

- By placing a Button object on the form, specifying it as a submit button, and choosing the format to submit the data.
- By placing an HTTP Submit button or Email Submit button on your form.

After you add a Button object to your form design, you can specify a format to submit data in the Submit tab of the Object palette. HTTP Submit buttons and Email Submit buttons are preconfigured to use specific submission formats. For more information about data submission formats, see *Submitting-datausingabutton*.

Designer supports four data submission formats:

- XML Data Package (XDP) This format is the only format that supports the inclusion of attachments.
- PDF
- XML Data (XML)
- URL-Encoded Data (HTTP Post)

The URL-Encoded Data (HTTP Post) format is the only data submission format for HTML forms.

RELATED LINKS:

Button properties in the Submit tab

Considerations for testing form designs with data

Optimizing and improving performance for Forms

Form transformations

Forms renders forms in numerous formats through transformations, which render forms to suit the capabilities of client applications such as Acrobat, Adobe Reader, a variety of web browsers, or a screen reader.

Forms supports any HTML client that follows the CSS2 specification. Because browsers vary widely in their support of CSS2, and older browsers do not provide support, several browsers and generic user agents require their own specific transformation.

For a list of all objects and the supported properties for each transformation type, see *Transformation Reference*.

NOTE: If you choose the HTML4 transformation to support Netscape 4.7.x, any JavaScript designated to run on the client is automatically run on the server instead.

Copying form designs to the local network

To supply form designs to Forms, you will need write access to the location from which Forms retrieves files at run time. The developer of the custom application will know the location, and your network administrator can set up the appropriate permissions.

Completed form designs are placed in the local file system for the developer of the custom application to access. You have to include any other files that are required to support the form design (for example, images).

If you are using the stand-alone version of Designer, you can publish the form designs to that location. After your access rights are set up, use the Designer Publish command (select File > Publish to Repository). See *Publishingforms*.

RELATED LINKS:

Considerations for creating form designs for Forms

Saving a form design: XDP or PDF

Saving a form design: XDP or PDF

Forms accepts form designs in either of the following formats:

- XDP
- PDF

XDP is the file format created by Designer. Choose this format to submit the form design, any form data, annotations, accessibility tags, and all other relevant information needed for Forms to subsequently render the form at run time. You must choose this format if the form will initiate server-side processing.

Save the form in PDF if the form will always be opened in Acrobat or Adobe Reader.

Choose PDF if the form contains a signature field or if users will be expected to save data directly in the PDF form at run time. If the rendered form will have a fixed layout and you expect the form to be requested often, providing the form design to Forms in PDF can improve run time performance.

Do not choose PDF in these situations:

- If the form will initiate server-side processing
- If Forms will be used to render forms that have a flexible layout.

RELATED LINKS:

Considerations for creating form designs for Forms

Copying form designs to the local network

19.6. Optimizing and improving performance for Forms

Designer provides a number of ways that you can optimize and improve the performance of your forms when using Forms. Using these improvements, you can configure time-saving features such as faster

form rendering times, rendering forms on the client to reduce burden on the server, and prepopulating forms with known data to reduce burden on form fillers.

Form caching

Form caching is an effective way to increase the performance of form rendering. When a form is cached, the data is merged into a pregenerated presentation. Using Forms, you can cache forms to improve rendering performance.

Forms that have a layout that adjusts to accommodate data can always be cached. Forms that have a fixed layout may also be cached but the following restrictions apply:

- If you have floating fields in forms that have a fixed layout and you select the Allow Form Rendering To Be Cached On Server option, the data in the fields will not render in the output PDF. To render the data in floating fields, ensure that this option is not selected.
- If the form can be filled by using Acrobat or Adobe Reader 6.0.2, only forms that have a fixed layout are eligible for caching. Form caching with forms that have a flowable layout is only supported by Acrobat and Adobe Reader 7.0.5 and later.
- All server-side scripting against the form layout is ignored. For example, you cannot script against such things as the fill color, font color, border width, or border color.
- Server-side scripting that changes the page content, the number of fields, the position of fields, or the appearance is ignored.
- When using the PDF or PDFForm transformations, you cannot change the layout of the form using client-side scripting. However, using the HTML transformations, client-side scripting to change the appearance of the form at the client is still possible, even when form caching is enabled.
- Usage rights are applied to the form before caching to further improve form rendering performance.
- Caching requires that each form be uniquely identified. If you want to create a new form by using an existing form as the starting point, do not use the operating system Copy command to copy the form. Instead, you should use the Designer File > Save As command to create the new form that is uniquely identified for caching.
- If you open an existing form in Designer and save the changes, the cache will be automatically updated.

  For Forms to cache forms that have a fixed layout, you must select the form caching option in Designer for each form that you create.

To specify form caching for a form that has a fixed layout

1) Select File > Form Properties.
2) Click the Defaults tab and select Allow Form Rendering To Be Cached On Server.
3) Click OK.

RELATED LINKS:

Form caching

Considerations for testing form designs with data

Form transformations

Considerations for Creating Forms for Server Processing

Rendering a form design that has a flowable layout on the client

If your form fillers are using Acrobat 7.0.5 or later, or Adobe Reader 7.0.5 or later, you can choose to have your forms render on the client instead of the server.

Rendering interactive or non-interactive forms that have a flowable layout, whether through Acrobat or Adobe Reader on the client side, achieves better performance than rendering on the server. This is because the Acrobat and Adobe Reader client applications, not the server, perform the rendering operation. Even forms that have a flowable layout and that involve data merging can be rendered on the client.

In addition, through client-side rendering, you optimize the delivery of PDF content and improve the ability of Forms to handle network load.

To render a form at the client, Forms must be set to render forms at the client and also generate a shell PDF.

The shell PDF file is a container that lets you deliver an XDP file (as part of the data stream) to the Acrobat or Adobe Reader client. It acts as the shell from which a dynamically rendered PDF is displayed and may contain embedded fonts that the XDP file requires. With the shell PDF file, Acrobat and Adobe Reader are able to open the XDP file and render the PDF on the client.

RELATED LINKS:

Working with forms that have a flowable layout

Prepopulating form fields with data

All form types can be prepopulated with data by using Forms. The data can come from a variety of sources, such as a database, another form, or another application.

Prepopulating a form has several advantages:

- Enables the user to view custom data in a form
- Reduces the amount of typing the user does to fill a form
- Ensures data integrity by having control over where data is placed

Prepopulating forms is faster and more secure when the prepopulation occurs during form rendering on the server instead of on the client.

Verifying your XML data source for data merging

When prepopulating forms with data, it is important to ensure that either the structure of the data conforms to the structure of your form design or that your form design conforms to the structure of your data.

In other words, an XML element must exist within your data source for every form field you want to prepopulate. Any discrepancies between the structures of your form and data source can lead to incorrect output. The XML element name must match a form field name, and XML elements that do not correspond to form fields are ignored.

The following two types of data sources can prepopulate a Designer form:

- An XDP data source that is XML and that conforms to XML Forms Architecture syntax
- An arbitrary XML data source that contains name/value pairs matching the form field names

An XML data source is used to prepopulate forms. However, an XML data source that prepopulates a form that has a flexible layout contains repeating XML elements that are used to prepopulate subforms that are repeated within the form itself.

RELATED LINKS:

Binding fields to a data source

Considerations for testing form designs with data

Working with forms that have a flowable layout

Using subforms

19.7. Creating HTML forms

In addition to PDF forms and guides, Forms has the ability to render interactive HTML forms for data capture. To distribute HTML forms, you must save your form designs as XDP files and deploy the XDP files and any other supporting files, such as images, to Forms.

Because HTML forms are displayed in a client application such as a web browser, they are subject to the constraints of the client application environment. Any type of object except a signature field can be included in an HTML form. However, depending on the limitations of the client application, not all object properties may be supported.

For a list of all objects and the supported properties for each transformation type, see *Transformation Reference*.

NOTE: When creating a form design for HTML output, you must ensure that all fields, exclusion groups, and subforms have unique names.

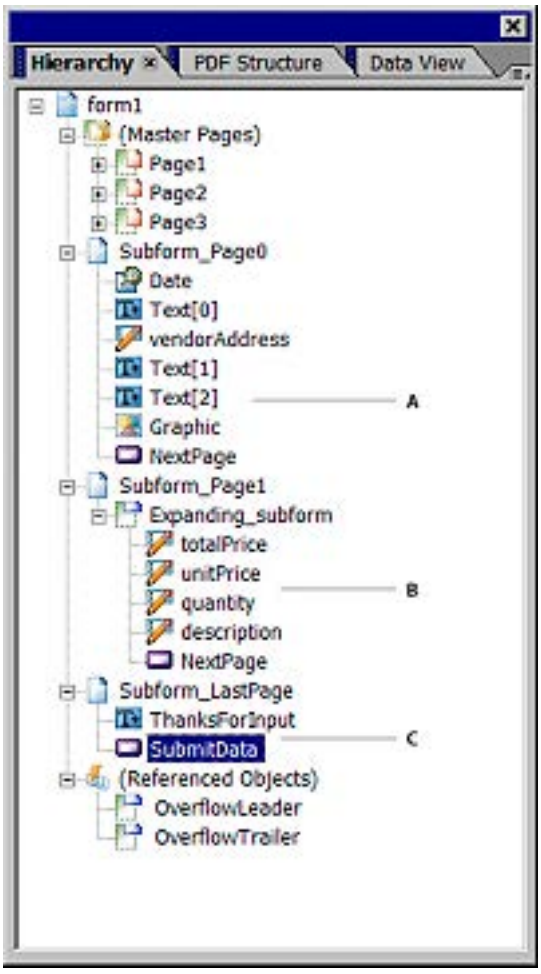
Using subforms to create HTML pages

When a form design contains subforms that expand, it is difficult to know exactly how many pages will be rendered in the resulting form at run time. The page size setting is used to paginate PDF forms, but it is ignored when an HTML form is rendered because HTML pages can be any length.

To implement the concept of a multipage HTML form, you can include a one page-level subform in your form design. You will need one of these subforms for every page that you want rendered in the form. For forms containing flowable elements, each subform that flows content should be nested inside one of the page-level subforms. Afterward, when the form is rendered, all content pertaining to the same page-level subform is displayed on the same HTML page.

NOTE: To implement the concept of a multipage HTML form, you must include one or more buttons in each page-level subform so that users can move from one page to the next. You will need to write a script to handle processing whenever a user clicks one of these buttons.

For example, consider the form design in the following illustration. The form design includes three page-level subforms: Subform_Page0, Subform_Page1, and Subform_LastPage. If Forms were to render an HTML form based on this form design, one HTML page would be created automatically from each of the page-level subforms.



A.

Subform_Page0 positions content and is set up to display its objects once on the first page of the form using the layout of master page Page1.

B.

Subform_Page1 contains a subform that flows content. Expanding_subform can repeat the rendering of its objects as many times as required in response to merged data. The content of Subform_Page1 can be displayed on one or more pages using the layout of master page Page2_etc.

C.

Subform_LastPage positions content and has been set up to display its objects on the last rendered page using the layout of master page Page3.

When Forms renders the HTML form at run time, the first page-level subform is displayed on the first page. For the purposes of this example, the first page might look similar to the page shown here. Notice the Next Page button, which must be included so that users can move to the next page of the form.

Assuming that at least one record would be available to merge with Expanding_subform at run time, the second page might look similar to the following example page. Again, a Next Page button is required so that users can move to the next page of the form. Also, you might choose to include a Previous Page button in your own form.

The third page-level subform, Subform_LastPage, is displayed on the last page of the rendered form. In this example, a message on the last page thanks users for filling the form and provides a Submit button for submitting the user-entered data to Forms.

For information about how to write a script so that users can move between HTML pages, see *Howtowriteascriptto handle HTML pages*.

RELATED LINKS:

Layout considerations for HTML forms

Image guidelines for creating HTML forms

Working around web browser limitations

Layout considerations for HTML forms

The following information will help you design HTML forms that are visually pleasing and easy to read:

- Do not use an object's border properties to draw lines, boxes, or grids on your form. Some browsers may not line up borders exactly as they appear in a Designer preview. Objects may appear layered or may push other objects off their expected position.
- If users will be using Microsoft Internet Explorer, you can design your form with lines, rectangles, and circles to define the background. All other web browsers support only vertical and horizontal lines; therefore, rectangles and circles on your form design will not display at run time.
- If users will be using Opera browsers, design the fields slightly larger than you want them to be. Fields in Opera always have a sunken inner border. Any border styles that you apply will be placed around the outside of this border. Because the sunken border takes up space in the field, the fillable area will shrink.
- Create static text slightly larger than what seems to be required to accommodate the text. Designer and Acrobat may use different kerning than that of a particular web browser, and some text may not be displayed correctly.

RELATED LINKS:

Image guidelines for creating HTML forms

Working around web browser limitations

Image guidelines for creating HTML forms

Consider the following guidelines when adding images to form designs rendered as HTML.

Supported image files

You can include any image file (except animated GIF files), subject to the limitations of the web browsers that users have to view HTML forms.

NOTE: Keep in mind that Internet Explorer processes the combination of button and image objects differently than other web browsers. For example, if you create a button object with a custom transparent appearance and place it on top of an image object, Internet Explorer may render the resulting HTML incorrectly such that users cannot actually click the button using a mouse. See *Workingaroundwebbrowserlimitations*.

Do not embed images in the form

Forms does not support embedded images. Instead, use relative path names to insert image files. For example, the path may be relative to the Forms forms root directory, which, by default, is the folder designated as the forms repository. The images folder in the following path is on the same level as the forms folder:

```
../images/graphic.jpg
```

RELATED LINKS:

Layout considerations for HTML forms

Working around web browser limitations

Working around web browser limitations

If you expect that some of the users will be using browsers that have limited capabilities, you need to consider the limitations of the lowest common denominator and design your forms accordingly. Alternatively, if your organization uses browsers that support XHTML, you have more options in laying out the form design.

Keep the following considerations in mind when designing forms that will be viewed in a variety of browsers:

- Specify a page size to ensure that the HTML4 transformation (for Netscape Navigator 4.7.x) displays all static objects properly. Otherwise, enough space for fields only will be reserved when the form is rendered.
- Incorporate a margin of at least 1/4 inch at the top and left sides of the form. Anything above and to the left of this margin will not be visible.
- Objects or portions of objects drawn at negative grid coordinates are not displayed. For example, if you draw an object starting at the vertical grid coordinate -.50, the portion of the object from -.50 to the inside edge of a 1/4-inch margin will not be displayed.
- Allow enough space around fields to compensate for the visual degradation that may occur in a low-end browser. For example, in some browsers, check boxes and radio buttons are displayed larger than they were designed.
- Any left alignment of data may not be maintained when a low-end browser is used to view the form, especially if users alter the default fonts associated with their browsers.
- Internet Explorer processes the combination of button and image objects differently than other web browsers. For example, if you create a button object with a custom transparent appearance and place it on top of an image object, Internet Explorer may render the resulting HTML incorrectly such that users cannot actually click the button using a mouse. However, it is possible for users to tab to the button and press it with the Enter key or spacebar. This processing error occurs because the size of the button object depends on the length of its caption. The amount of empty space in the caption area needs to be large enough so that the size of the button is bigger than the size of the image. To resolve this issue, replace the caption text with enough empty spaces to allow users to click the button using a mouse.

RELATED LINKS:

Layout considerations for HTML forms

Image guidelines for creating HTML forms

Form caching

Previewing an HTML form

To test the operation of your form design with Forms and to set up command buttons accurately in a form design, you need to know the URL that will be associated with requests to Forms. The developer of the custom application will know the URL.

To preview an HTML form, you must make the form design available to Forms so that it can store the form design. Afterward, you can request the form through the URL that is associated with Forms. Use a web browser or one of the target client applications (such as a screen reader) to open the form.

RELATED LINKS:

Considerations for Creating Forms for Server Processing

Copying form designs to the local network

Where calculations and scripts are run

You can embed calculations and scripts in a form design to execute calculations, methods, or operations when any of an object's events occur at run time. For example, an event occurs at run time when the user performs the action that the event specifies. You can call any of an object's supported methods and examine or set properties by defining a script.

In Designer, scripts and calculations are designated, by default, to run on the client device. The default processing location is defined in the Preview tab of the Form Properties dialog box (select File > Form Properties). To override the default processing location, you can explicitly specify a different processing location by using the Run At option in the Script Editor when you attach a script or calculation to an object.

When you have Forms, processing can be done on the client, the server, or both. When you specify that a script/calculation is to run on the client and server, the client and server may both attempt to run the script/calculation. Forms always attempts to process the script/calculation if the client cannot do so. If you designate scripts/calculations to run on the server, Forms runs the scripts and/or calculations, remerges the results into the form, and returns both to the client.

Client-side scripts and calculations are run on the client device. When creating PDF forms for Acrobat or Adobe Reader, all processing must be done on the client. However, if the client cannot run the script or calculation, Forms will attempt to process the script or calculation.

To run a client-side script in an HTML form successfully, certain conditions must be met:

- The client application must be Microsoft Internet Explorer 5.x, Netscape 6.0 or later, or Opera 5 or later.
- JavaScript can be used only to write scripts (you cannot include FormCalc calculations in the form design).
- JavaScript must be enabled in the client application.

RELATED LINKS:

About FormCalc

Client- and server-side processing

Processing can be executed on the client, the server, or both. Scripts and calculations behave differently when they are executed on the client compared to the server.

Client-side processing

At run time, if processing has been set to take place on the client, any scripts/calculations execute in real time on the user's computer. The code and variables declared become available almost as soon as the form is opened. This information remains available, and the state of the data is maintained until one of these situations occurs:

- Another script is executed.
- Another script removes the associated object.
- The form is closed.

Server-side processing

Forms can manage any scripts/calculations that the client application cannot manage. For example, to prepopulate a form, you may have to use a script to connect to a database or web service that might not be available at the client. When Forms runs a script or calculation, the execution is done while the form is being rendered. No code or variables remain available after processing is completed. In other words, if you add variables to a script or calculation, they remain valid only for the duration of processing.

NOTE: If you choose the HTML4 transformation to support Netscape Navigator 4.7.x, any JavaScript designated to run on the client is automatically run on the server instead.

RELATED LINKS:

About FormCalc

Events that you can reference in a script or calculation

Server-side scripts

Acrobat and Adobe Reader recognize all the events that Designer supports. The client application sends user-initiated events to Forms for server-side processing. No other types of events trigger server-side processing. Forms performs server-side processing whenever it is rendering a form, executing server events initiated from the client, or processing submitted data.

When you specify that a script or calculation should be run on the server, the following user-initiated events cause Acrobat or Adobe Reader to send the event to Forms for processing:

- exit
- mouseEnter
- mouseExit
- change
- mouseUp
- mouseDown
- click

During event processing, Forms runs all the scripts /calculations that are designated to run on the server and remerges the results into the form before returning it to the client application.

If any of these events are referenced in a server-side script/calculation, Acrobat or Adobe Reader ignores them:

- initialize
- calculate
- validate
- docReady
- docClose

The following table identifies the events that you can reference in server-side scripts/calculations only. These events are not recognized by HTML client applications.

Event	For more information, see
form:ready	*form:readyevent*
layout:ready	*layout:readyevent*

When a script has been designated to run on the server, the click event (of a regular button) is the only event that causes an HTML client to initiate server-side processing. During processing, Forms remerges the results into the HTML form before returning it to the client application. All other events are ignored by the HTML client and are executed only when Forms performs server-side processing.

Client-side scripts

For client-side scripting and calculations, Acrobat and Adobe Reader support the full list of events that you can define in Designer.

However, not all of these events are recognized by HTML clients. If you will be using a single form design to create both PDF forms and HTML forms, your client-side scripts should reference only a subset of the supported events.

If a client-side script references an event that the HTML client does not recognize, the script will stop executing at the point where the unrecognized event is referenced.

You can use the following events to run client-side scripts from HTML forms.

Event	For more information, see
initialize	*initializeevent*
enter	*initializeevent*
exit	*initializeevent*
calculate	*calculateevent* ***NOTE:*** *In HTML forms only, activities triggered by calculate events are run when the cursor exits a field. Processing does not start when a user changes a field value, but you can invoke processing explicitly at this time, if required, by using the execCalculate() method.*
validate	*validateevent*

Event	For more information, see
change	*changeevent* ***NOTE:*** *In PDF and HTML forms, this event is supported for drop-down lists and list boxes only.*
mouseUp	*mouseUpevent*
mouseDown	*mouseDownevent*
click	*clickevent* ***NOTE:*** *The click event cannot be used for Submit buttons in PDF or HTML forms. Use the preSubmit event instead.*
preSubmit	*preSubmitevent* ***NOTE:*** *In PDF and HTML forms, this event is supported for Submit buttons only. When you use the preSubmit event to run a client-side script, processing is done before the data is submitted. If the event runs a script on the server, processing is done while the data is being submitted.*

RELATED LINKS:

Scripting Using Designer

Summary of supported properties, methods, and events

NOTE: *Guides is deprecated.* With the exception of a few host object methods, Acrobat and Adobe Reader support the full range of properties, methods, and events. HTML clients recognize a limited number of these properties, methods, and events.

The following tables summarize which client applications recognize the various properties, methods, and events that you can reference in client- and/or server-side scripts.

Host object - Properties	Acrobat / Adobe Reader
currentPage (read only)	Yes
numPages (read only)	Yes
name (read only)	Yes
validationsEnabled	Yes
calculationsEnabled	Yes

Host object - Methods	Acrobat / Adobe Reader
pageUp	Yes
pageDown	Yes

Host object - Methods	Acrobat / Adobe Reader
exportData	Yes
importData	Yes
gotoURL	Client only.
messageBox	Client only.
resetData	Yes
setFocus	Client only.

Host object - Events	Acrobat / Adobe Reader
docReady	Yes
docClose	Yes
postPrint	Yes
postSave	Yes
prePrint	Yes
preSave	Yes

Form object - Methods	Acrobat / Adobe Reader
resolveNodes	Yes
resolveNode	Yes
execCalculate	Yes
execValidate	Yes
execInitialize	Yes

Subform object - Properties	Acrobat / Adobe Reader
name (read only)	Yes
index (read only)	Yes
x	Yes
y	Yes
w	Yes

Subform object - Properties	Acrobat / Adobe Reader
h	Yes
validationMessage	Yes

Subform object - Methods	Acrobat / Adobe Reader
resolveNodes	Yes
resolveNode	Yes
execCalculate	Yes
execValidate	Yes
execInitialize	Yes

Subform object - Events	Acrobat / Adobe Reader
enter	Yes
exit	Yes
initialize	Yes
validate	Yes
calculate	Yes

Field objects - Properties	Acrobat / Adobe Reader
name (read only)	Yes
index (read only)	Yes
rawValue	Yes
formattedValue	See Note 1.
x	Yes
y	Yes
w	Yes
h	Yes
presence	Yes
mandatory	Yes

Field objects - Properties	Acrobat / Adobe Reader
fontColor	Yes
fillColor	Yes
borderColor	Yes
borderWidth	Yes
validationMessage	Yes

Field objects - Methods	Acrobat / Adobe Reader
execCalculate	Yes
execInitialize	Yes
execValidate	Yes
addItem	Yes, for drop-down lists and list boxes only.
clearItems	Yes, for drop-down lists and list boxes only.
resolveNodes	Yes
resolveNode	Yes

Field objects - Events	Acrobat / Adobe Reader
exit	Yes
enter	Yes
calculate	Yes
validate	Yes
initialize	Yes
click	Yes, but not for Submit buttons. See Note 2.
change	Yes
mouseUp	Yes
mouseDown	Yes
preSubmit	Yes

Note 1: For list boxes, formattedValue does not return the display text.

Note 2: The click event is not supported for Submit buttons in PDF. Use preSubmit instead.

ScriptObject object	Acrobat / Adobe Reader
See Note 3.	Yes

Note 3: The ScriptObject object can be created and used with any other script. See To create a script object.

Event object - Properties	Acrobat / Adobe Reader
prevText	Yes
prevContentType	Yes
newText	Yes
newContentType	Yes
fullText	Yes
commitKey	Yes
keyDown	Yes
modifier	Yes
name	Yes
selEnd	Yes
selStart	Yes
shift	Yes
target	Yes
change	Yes

Note 1: For list boxes, formattedValue does not return the display text.

Note 2: The click event is not supported for Submit buttons in PDF forms. Use preSubmit instead.

Note 3: The ScriptObject object can be created and used with any other script. See To create a script object.

Expressions supported by HTML clients

A simplified subset of reference syntax expressions are supported by HTML clients:

- FormCalc calculations are not valid in HTML browsers and are removed prior to the form being rendered in HTML.
- The ellipsis (...) syntax is not supported.

- When writing a client-side script for HTML forms, you must use the JavaScript resolveNode expression to locate nodes in the hierarchy. The script cannot walk the hierarchy using object notation. For example, the following expression is not supported:

  `xfa.form.subform1.TextEdit1`

The following expression is supported:

`xfa.form.resolveNode("subform1.TextEdit1")`

Unqualified field references may be used to locate sibling fields in the hierarchy.

NOTE: If you choose the HTML4 transformation to support Netscape Navigator 4.7.x, any JavaScript designated to run on the client is automatically run at the server instead.

RELATED LINKS:

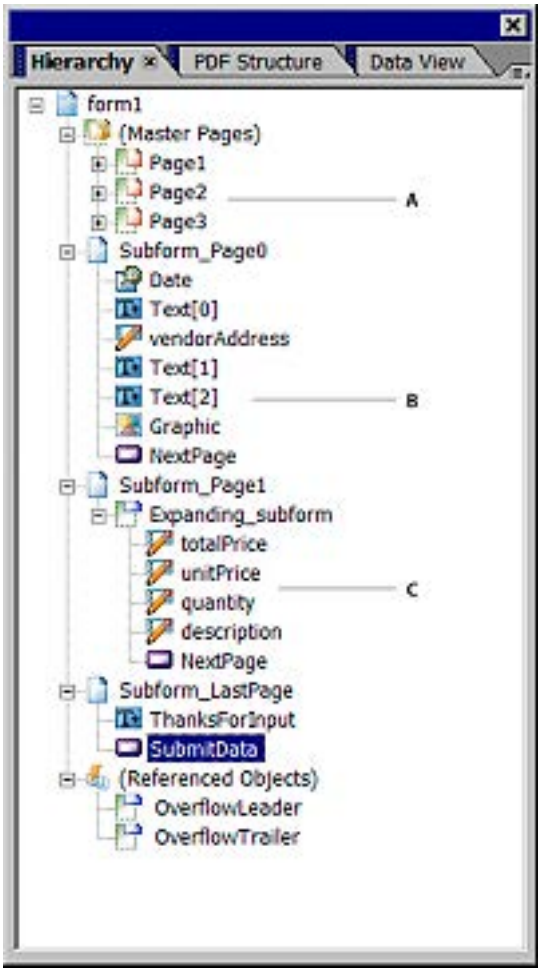
Scripting Using Designer

How to write a script to handle HTML pages

When the same form design is used to render PDF and HTML forms, the page size setting is used to paginate PDF forms, but it is ignored when an HTML form is rendered. To deal with the differences between PDF and HTML pages, form authors may optionally use page-level subforms to set up artificial HTML pages. If HTML pages are set up this way, JavaScript is needed to enable users to move from one HTML page to the next at run time.

You can use the pageUp() and pageDown() methods to enable users to move between HTML pages by using a regular command button that triggers processing when the button's click event occurs. For both PDF and HTML forms, processing must be run at the client and server.

The pageUp() and pageDown() methods operate on page-level subforms. For example, assume that the form author has set up the following structure in the form design:



A.

This page-level subform corresponds to the first page.

B.

This page-level subform corresponds to the second page.

C.

This page-level subform corresponds to the third page.

If the user were viewing the HTML page that corresponds to Subform_Page0, calling xfa.host.pageDown() would cause Subform_Page1 to be displayed in the browser. Similarly, calling xfa.host.pageUp() while Subform_Page1 is on display would cause Subform_Page0 to be displayed in the browser. Page numbers would be assigned to the following properties to manipulate these HTML pages:

```
xfa.host.currentPage = 0 //moves to the first page
xfa.host.currentPage = 1 //moves to the second page
xfa.host.currentPage = 2 //moves to the third page
xfa.host.currentPage = xfa.host.numPages -1 //moves to the last page
```

As users move between HTML pages, the state of the data is maintained, but the presentation state of an object (for example, the background color of a field) may change from one page to the next. You can

maintain presentation settings between pages using hidden fields that contain state values for the various settings and command buttons that enable users to page back and forward through the pages of the form. Your script would update the presentation states of fields based on the values in the hidden fields. When a user clicks either of the command buttons, the calculate event for the button could be used to run the script.

For example, the following JavaScript maintains the fill color of a field based on the value of a field named hiddenField. The script is triggered when the calculate event occurs.

```
If (hiddenField.rawValue == 1)
this.fillColor = "255,0,0"
else
this.fillColor = "0,255,0"
```

NOTE: When running scripts in an HTML form on the client device, you can script against the current subform/HTML page only.

RELATED LINKS:

pageUp

pageDown

currentPage

19.8. Optimizing and improving performance for Adobe document services

Designer provides a number of ways that you can optimize and improve the performance of your forms when using Adobe document services. Using these improvements, you can configure time-saving features such as faster form rendering times, rendering forms on the client to reduce burden on the server, and prepopulating forms with known data to reduce burden on form fillers.

Form caching

Form caching is an effective way to increase the performance of form rendering. When a form is cached, the data is merged into a pregenerated presentation. Using Adobe document services, you can cache forms to improve rendering performance.

Forms that have a layout that adjusts to accommodate data can always be cached. Forms that have a fixed layout may also be cached but the following restrictions apply:

- If you have floating fields in forms that have a fixed layout and you select the Allow Form Rendering To Be Cached On Server option, the data in the fields will not render in the output PDF. To render the data in floating fields, ensure that this option is not selected.
- If the form can be filled by using Acrobat or Adobe Reader 6.0.2, only forms that have a fixed layout are eligible for caching. Form caching with forms that have a flowable layout is only supported by Acrobat and Adobe Reader 7.0.5 and later.

- All server-side scripting against the form layout is ignored. For example, you cannot script against such things as the fill color, font color, border width, or border color.
- Server-side scripting that changes the page content, the number of fields, the position of fields, or the appearance is ignored.
- Reader usage rights are applied to the form before caching to further improve form rendering performance.
- Caching requires that each form be uniquely identified. This identification occurs automatically when you create forms using Designer embedded in an SAP development environment. If you create forms using Designer outside an SAP development environment, and you want to create a new form by using an existing form as the starting point, do not use the operating system Copy command to copy the form. Instead, you should use the Designer Edit > Form Properties command to create the new form that is uniquely identified for caching.
- If you open an existing form in Designer and save the changes, the cache will be automatically updated.

  For Adobe document services to cache forms that have a fixed layout, you must select the form caching option in Designer for each form that you create.

To specify form caching for a form that has a fixed layout

1) Select Edit > Form Properties.
2) Click the Defaults tab and select Allow Form Rendering To Be Cached On Server.
3) Click OK.

Rendering a form design that has a flowable layout on the client

If your form fillers are using Acrobat 7.0.5 or later, or Adobe Reader 7.0.5 or later, you can choose to have your forms render on the client instead of the server.

Rendering interactive or non-interactive forms that have a flowable layout, whether through Acrobat or Adobe Reader on the client side, achieves better performance than rendering on the server. This is because the Acrobat and Adobe Reader client applications, not the server, perform the rendering operation. Even forms that have a flowable layout and that involve data merging can be rendered on the client.

In addition, through client-side rendering, you optimize the delivery of PDF content and improve the ability of Adobe document services to handle network load.

To render a form at the client, Adobe document services must be set to render forms at the client and also generate a shell PDF.

The shell PDF file is a container that lets you deliver an XDP file (as part of the data stream) to the Acrobat or Adobe Reader client. It acts as the shell from which a dynamically rendered PDF is displayed and may contain embedded fonts that the XDP file requires. With the shell PDF file, Acrobat and Adobe Reader are able to open the XDP file and render the PDF on the client.

For more information, see *Working with forms that have a flowable layout*.

19.9. Creating scripts for use with Adobe document services

Where calculations and scripts are run

You can embed calculations and scripts in a form design to execute calculations, methods, or operations when any of an object's events occur at run time. For example, an event occurs at run time when the user performs the action that the event specifies. You can call any of an object's supported methods and examine or set properties by defining a script.

In Designer, scripts and calculations are designated, by default, to run on the client device. The default processing location is defined in the Preview tab of the Form Properties dialog box (select Edit > Form Properties). To override the default processing location, you can explicitly specify a different processing location by using the Run At option in the Script Editor when you attach a script or calculation to an object.

When you have Adobe document services, processing can be done on the client, the server, or both. When you specify that a script/calculation is to run on the client and server, the client and server may both attempt to run the script/calculation. Adobe document services always attempt to process the script/calculation if the client cannot do so. If you designate scripts/calculations to run on the server, Adobe document services run the scripts and/or calculations, remerge the results into the form, and return both to the client.

Client-side scripts and calculations are run on the client device. When creating PDF forms for Acrobat or Adobe Reader, all processing must be done on the client. However, if the client cannot run the script or calculation, Adobe document services will attempt to process the script or calculation.

Client- and server-side processing

Processing can be executed on the client, the server, or both. Scripts and calculations behave differently when they are executed on the client compared to the server.

Client-side processing

At run time, if processing has been set to take place on the client, any scripts/calculations execute in real time on the user's computer. The code and variables declared become available almost as soon as the form is opened. This information remains available, and the state of the data is maintained until one of these situations occurs:

- Another script is executed.
- Another script removes the associated object.
- The form is closed.

Server-side processing

Adobe document services can manage any scripts/calculations that the client application cannot manage. For example, to prepopulate a form, you may have to use a script to connect to a database or web service that might not be available at the client. When Adobe document services runs a script or calculation, the execution is done while the form is being rendered. No code or variables remain available after processing is completed. In other words, if you add variables to a script or calculation, they remain valid only for the duration of processing.

Events that you can reference in a script or calculation

Acrobat and Adobe Reader recognize all the events that Designer supports. The client application sends user-initiated events to Adobe document services for server-side processing. No other types of events trigger server-side processing. Adobe document services perform server-side processing whenever they are rendering a form, executing server events initiated from the client, or processing submitted data.

When you specify that a script or calculation should be run on the server, the following user-initiated events cause Acrobat or Adobe Reader to send the event to Adobe document services for processing:

- exit
- mouseEnter
- mouseExit
- change
- mouseUp
- mouseDown
- click

During event processing, Adobe document services run all the scripts /calculations that are designated to run on the server and remerges the results into the form before returning it to the client application.

If any of these events are referenced in a server-side script/calculation, Acrobat or Adobe Reader ignores them:

- initialize
- calculate
- validate
- docReady
- docClose

For client-side scripting and calculations, Acrobat and Adobe Reader support the full list of events that you can define in Designer.

Summary of supported properties, methods, and events

With the exception of a few host object methods, Acrobat and Adobe Reader support the full range of properties, methods, and events. HTML clients recognize a limited number of these properties, methods, and events.

If you use a single form design to create PDF forms, your scripts may reference a subset of the supported properties, methods, and events.

The following tables summarize which client applications recognize the various properties, methods, and events that you can reference in client- and/or server-side scripts.

Host object - Properties	Acrobat / Adobe Reader
currentPage (read only)	Yes
numPages (read only)	Yes
name (read only)	Yes
validationsEnabled	Yes
calculationsEnabled	Yes

Host object - Methods	Acrobat / Adobe Reader
pageUp	Yes
pageDown	Yes
exportData	Yes
importData	Yes
gotoURL	Client only.
messageBox	Client only.
resetData	Yes
setFocus	Client only.

Host object - Events	Acrobat / Adobe Reader
docReady	Yes
docClose	Yes
postPrint	Yes
postSave	Yes
prePrint	Yes
preSave	Yes

Form object - Methods	Acrobat / Adobe Reader
resolveNodes	Yes
resolveNode	Yes
execCalculate	Yes
execValidate	Yes
execInitialize	Yes

Subform object - Properties	Acrobat / Adobe Reader
name (read only)	Yes
index (read only)	Yes
x	Yes
y	Yes
w	Yes
h	Yes
validationMessage	Yes

Subform object - Methods	Acrobat / Adobe Reader
resolveNodes	Yes
resolveNode	Yes
execCalculate	Yes
execValidate	Yes
execInitialize	Yes

Subform object - Events	Acrobat / Adobe Reader
enter	Yes
exit	Yes
initialize	Yes
validate	Yes
calculate	Yes

Field objects - Properties	Acrobat / Adobe Reader
name (read only)	Yes
index (read only)	Yes
rawValue	Yes
formattedValue	See Note 1.
x	Yes
y	Yes
w	Yes
h	Yes
presence	Yes
mandatory	Yes
fontColor	Yes
fillColor	Yes
borderColor	Yes
borderWidth	Yes
validationMessage	Yes

Field objects - Methods	Acrobat / Adobe Reader
execCalculate	Yes
execInitialize	Yes
execValidate	Yes
addItem	Yes, for drop-down lists and list boxes only.
clearItems	Yes, for drop-down lists and list boxes only.
resolveNodes	Yes
resolveNode	Yes

Field objects - Events	Acrobat / Adobe Reader
exit	Yes
enter	Yes

Field objects - Events	Acrobat / Adobe Reader
calculate	Yes
validate	Yes
initialize	Yes
click	Yes, but not for Submit buttons. See Note 2.
change	Yes
mouseUp	Yes
mouseDown	Yes
preSubmit	Yes

Note 1: For list boxes, formattedValue does not return the display text.

Note 2: The click event is not supported for Submit buttons in PDF forms. Use preSubmit instead.

ScriptObject object	Acrobat / Adobe Reader
See Note 3.	Yes

Note 3: The ScriptObject object can be created and used with any other script. See “To create a script object” in *Scripting Basics*.

Event object - Properties	Acrobat / Adobe Reader
prevText	Yes
prevContentType	Yes
newText	Yes
newContentType	Yes
fullText	Yes
commitKey	Yes
keyDown	Yes
modifier	Yes
name	Yes
selEnd	Yes
selStart	Yes
shift	Yes

Event object - Properties	Acrobat / Adobe Reader
target	Yes
change	Yes

With the exception of a few host object methods, Acrobat and Adobe Reader support the full range of properties, methods, and events. The following tables summarize the various properties, methods, and events that you can reference in client- and/or server-side scripts.

Note 1: For list boxes, formattedValue does not return the display text.

Note 2: The click event is not supported for Submit buttons in PDF forms. Use preSubmit instead.

Note 3: The ScriptObject object can be created and used with any other script. See “To create a script object” in *Scripting Basics*.

20. Creating Forms for forms workflow

Designer can create form designs that are used in AEM forms applications. An application is a solution that consists of a process and its associated assets (form design, images, and fragments), which represent an automated process. An AEM forms application is typically developed in Workbench. To learn more about creating a AEM forms application, go to Creating Your First AEM formsApplication.

NOTE: The Flex Worksapce is deprecated for AEM forms.

Workspace is the browser-based component of forms workflow, which lets end users participate in automated form-based business processes. Using workspace, users can perform these tasks:

- Fill forms that initiate a business process.
- Open forms that have been routed to the user or to a group queue to review, approve, or provide more information.
- Search for forms that are part of a completed business process or active processes that the user has initiated or participated in.

  Suppose a business process involves the handling of mortgage applications. A form developer creates the form design for use in the process, which is created in Workbench. Then, mortgage applicants can fill the application form electronically by using Workspace. A loan officer or bank manager can also approve the application using Workspace.

  With forms workflow, you can render a form design to a client in these output types:
- HTML
- PDF
- **NOTE:** To use a form that was created in Acrobat in Workspace, you must use Acrobat to prepare the form design. For instructions, see “Leveraging Acrobat forms in Workspace” in Application Development Using Workbench.

20.1. Preparing a form for use in Workspace

A form design intended for use in Workspace must be properly configured to display in Workspace. Depending on your application and the desired output, the form design should contain one or more of the following objects:

- Submit button (available in the Object palette). If clients are using Adobe Reader 9.1 or greater, adding a submit button to the form design is not required.
- Process Fields object (available in the Custom category of the Object Library palette)
- Form Bridge object (available in the Custom category of the Object Library palette). The Form Bridge object enables communication to PDF forms outside of Acrobat.

  The New Form wizard automatically adds a Submit button and the Process Fields Form Bridge objects to the form design based on the selections you make in the wizard. The New Form wizard also sets the proper submit format of the Submit button based on the settings in the wizard.

  By default, Workspace contains a Complete button for PDF forms and guides. Typically, having a submit button and a complete button on a single form is not desirable. To understand how Work-

space handles the Complete and Submit buttons in PDF forms, see HowWorkspacehandles buttons in PDF forms.

The Process Fields object contains a button object, a drop-down list, and a number of additional hidden objects. For more information about the Process Fields object, see Fieldsinthe Process Fields object.

The following table describes some common scenarios related to rendering forms in Workspace and the object(s) that are required on the form design.

NOTE: If clients are using Adobe Reader 9.1 or later, a Submit button is not required on the form design.

NOTE: Guides is deprecated.

Task	What to use on the form design
Render an XDP form design in Workspace for online PDF output.	Submit button.
Render a PDF form design in Workspace for online PDF output.	Submit button.
Render a guide in Workspace. The form design can be a PDF or an XDP.	Submit button. **Note**: *This applies to guides created in Designer 8.x. To create a guide in Designer 9.x and later, use Guide Builder available in Workbench.*
Enable users to send forms by email.	Process Fields object
Render an XDP form design in offline capable PDF format. This enables Workspace users to take PDF forms offline.	Process Fields object **Note**: *You cannot render a guide in offline capable PDF format*.
Render an XDP form design in Workspace in HTML format.	Process Fields object **Note**:*In HTML forms, the Complete button in Workspace is not displayed. The location of the Process Fields object in the form design is where the buttons in the HTML form appear.*
Certify, encrypt, or apply usage rights to a PDF or XDP form design before its first delivery to a user by Forms or Workspace.	Submit button and Form Bridge object.
Save the form as a PDF form in Designer and certify, encrypt, or apply usage rights before saving it in the repository.	Submit button and Form Bridge object.
Enable the signing of PDF forms in Workspace.	Submit button.

Using the Form Bridge object with form designs that contain Artwork

When you add the Form Bridge object to a form design that contains Artwork, you will not see the object in the Design View.

To submit form designs that contain Artwork to Workspace:

1) Open the PDF form that contains Artwork.
2) Select the root node.
3) Add a script object to it and name the object ContainerFoundation_JS.
4) Open a blank form and add the Form Bridge object to it.
5) Select the ContainerFoundation_JS object.
6) In the Script Editor, select the form ready event and the JavaScript language.
7) Copy the script into the ContainerFoundation_JS object of the PDF form you opened in step 1.
8) In the blank form, select the root node.
9) In the Script Editor, select the form ready event and the JavaScript language.
10) Copy the script into the root node of the PDF form you opened in step 1.

20.2. How Workspace handles buttons in PDF forms

The Workspace Complete button allows users to submit PDF forms.

NOTE: If you want the user submission data to include the PDF, instead of only the XDP data, you need to use Acrobat Reader DC extensions to provide the appropriate rights to the form.

Here is a summary of how Workspace handles buttons in a PDF form.

- When a form design does not contain a submit button, Workspace disables the Complete button, and users cannot submit the form.
- When a form design contains one submit button, Workspace hides the submit button and enables the Workspace Complete button.
- When a form design contains an indirect submit button, that is, a regular button that points to a submit button using script, Workspace hides the indirect submit button and enables the Workspace Complete button. When a user clicks the Complete button, Workspace executes the indirect submit button script, which in turn points to the submit button that performs the form data submission. It is assumed that the form author sets the submit button that performs the form data submission to be hidden.
- When a form design has multiple indirect submit buttons pointing to one or more submit buttons, Workspace disables the Workspace Complete button. The user must click the appropriate button on the form to submit it. The user can still save a draft version of the form or take the form offline.

  NOTE: If clients are using Adobe Reader 9.1 or later, a submit button is not required

RELATED LINKS:

Submitting data using a button

20.3. Fields in the Process Fields object

The following table lists the fields in the Process Fields object and any configuration required for the field to work correctly.

NOTE: You must not group the process fields. If you group these fields, the form will not work with forms workflow.

Field name	Object	Description
AWS_SUBMIT	Button	Initiates the request to submits the form to forms workflow. Validates whether the submission has already occurred and determines the host environment (email or web) before initiating the request to submit. To convert forms and form data to a PDF document and save the reference in a document variable, you need to configure the FSSUBMIT_button object to use a submit format of PDF.
AWS_ACTION	Drop-down List	Lists the choices available for completing the work item. This field is populated by the AWS_CHOICE field. If you prefer not to provide choices, leave the AWS_CHOICE field empty.
AWS_ASSIGNED_ID	Text field (invisible)	A hidden text field that stores the name of the participant who submits the form.
AWS_STATUS	Text field (invisible)	A hidden text field that is automatically populated with status information.
AWS_TASKID	Text field (invisible)	A hidden text field that is automatically populated with task identity information. This field contains the identity of the task associated with the form.
AWS_MAILTO	Text field (invisible)	A hidden text field that contains the email address that forms workflow uses.
AWS_CHOICE	Text field (invisible)	A hidden text field that is automatically populated with data. The data is a comma-delimited list of route names.

Field name	Object	Description
AWS_PROCESSTYPE	Text field (invisible)	A hidden text field that contains the process type with which the form is associated. If the AWS_TASKID field is empty, forms workflow invokes the process type that is specified in the AWS_PROCESSTYPE field.
AWS_SUBMIT_TYPE	Text field (invisible)	A hidden text field that specifies the submit type as XDP or PDF. The default value is XDP. The New Form wizard populates this field with the correct submit format. If desired, you can manually change this field. If the AWS_SUBMIT_TYPE field is set to PDF, then the PDF_SUBMIT will be invoked. If the AWS_SUBMIT_TYPE field is set to XDP, then the FSSUBMIT_ will be invoked.
FSSUBMIT_	Button (invisible)	A hidden submit button that is triggered when a user clicks the Workspace Complete button. When the user clicks the Complete button, this button submits the form to AEM forms. If the AWS_SUBMIT_TYPE field is set to XDP, FSSUBMIT_ is automatically invoked.
PDF_SUBMIT	Button (invisible)	A hidden submit button that is configured to submit as PDF. If the AWS_SUBMIT_TYPE field is set to PDF, PDF_SUBMIT is automatically invoked.

RELATED LINKS:

Preparing a form for use in Workspace

21. Working with the Keyboard

21.1. Default keyboard shortcuts

Keys for common actions

Result	Action
Activate or deactivate menu bar mode.	Alt or F10
Cancel the current task, or close an open menu or dialog box.	Esc
Display the shortcut menu for the selected item.	Shift+F10
Move the cursor to the beginning of the text in the current line.	Home
Move the cursor to the end of the text in the current line.	End
Move the cursor to the end of the text.	Ctrl+End
Move the cursor to top of the text.	Ctrl+Home
Open the Designer Help window.	F1
Select (the same as primary mouse button click).	Spacebar
Opens the New Form Assistant, by default. Opens the New dialog box, if you clear the Show When Creating New Documents option, under Form Assistance on the Wizards and Tips panel in the Options dialog box.	Ctrl + N

Keys for undoing and redoing an action

Result	Action
Redo the undone action.	Ctrl+Shift+Z or Ctrl+Y

Result	Action
Undo the last action.	Ctrl+Z or Alt+Backspace

Keys for deleting text

Result	Action
Cut the selected text and send it to the clipboard.	Ctrl+X or Shift+Delete
Delete one character to the left.	Backspace
Delete one character to the right or delete selected text without sending it to the clipboard.	Delete
Delete one word to the left.	Ctrl+Backspace
Delete one word to the right.	Ctrl+Delete

Keys for selecting text

Result	Action
Leave edit mode.	F2
Select from the cursor position to the beginning of the paragraph.	Ctrl+Shift+Up Arrow
Select from the cursor position to the end of the paragraph.	Ctrl+Shift+Down Arrow
Select or deselect from cursor position to the end of the word.	Ctrl+Shift+Right Arrow
Select or deselect from the cursor position to the beginning of the current line.	Shift+Home
Select or deselect from the cursor position to the beginning of the word.	Ctrl+Shift+Left Arrow
Select or deselect from the cursor position to the end of the current line.	Shift+End
Select or deselect from the cursor position to the same position on the next line. If the cursor is on the last line, the selection ends at the end of the current line.	Shift+Down Arrow

Result	Action
Select or deselect from the cursor position to the same position on the previous line. If the cursor is on the first line, the selection ends at the beginning of the current line.	Shift+Up Arrow
Select or deselect one character to the left	Shift+Left Arrow
Select or deselect one character to the right.	Shift+Right Arrow
Select or deselect one screen to the same position on the next screen.	Shift+Page Down
Select or deselect one screen to the same position on the previous screen.	Shift+Page Up
Select or deselect to the beginning of the object's text (top, left text position of the object).	Ctrl+Shift+Home
Select or deselect to the end of the object's text (bottom, right text position of the object).	Ctrl+Shift+End
Selects all text in a text field or in the Script Editor. Selects all objects in a page.	Ctrl+A

Keys for copying and moving text

Result	Action
Copy the selected text to the clipboard.	Ctrl+C or Ctrl+Insert
Cut the selected text and put it on the clipboard.	Ctrl+X
Delete one character to the right or delete selected text without sending it to the clipboard.	Delete
Delete one word to the right.	Ctrl+Delete
Paste the clipboard content at the insertion point.	Ctrl+V or Shift+Insert

Keys for applying character formats to selected text

Result	Action
Apply bold formatting (font must support Bold).	Ctrl+B

Result	Action
Apply italic formatting (font must support Italic).	Ctrl+I
Apply underline formatting.	Ctrl+U

Keys for special characters

Result	Action
Add a non-breaking space.	Ctrl+Shift+Spacebar

Keys for setting line spacing

Result	Action
Apply 1.5-line spacing.	Ctrl+Shift+5
Apply double-line spacing.	Ctrl+Shift+2
Apply single-line spacing.	Ctrl+Shift+1

Keys for aligning paragraphs

Result	Action
Center a paragraph.	Ctrl+E
Decrease a hanging indent.	Ctrl+Shift+T
Increase a hanging indent.	Ctrl+T
Increase indent of a paragraph from the left.	Ctrl+M
Justify a paragraph.	Ctrl+J
Left align a paragraph.	Ctrl+L
Decrease a paragraph indent from the left.	Ctrl+Shift+M
Reset the paragraph formatting to the default before changes were made.	Ctrl+Q
Right align a paragraph.	Ctrl+R

Keys for controlling windows

Result	Action
Display the Preview PDF tab.	F5
Move to the next open document.	Ctrl+F6
Move to the next pane or palette.	Ctrl+Tab
Move to the next tab within the current palette	Crtl+,
Move to the previous pane or palette.	Ctrl+Shift+Tab
Move to the previous tab within the current palette	Crtl+.
Quit Designer.	Alt+F4
Turn on/off display of grid.	Ctrl+’
Turn on/off Snap To Grid.	Ctrl+Shift+’
Expand/collapse workspace palettes	Ctrl+F3 ***NOTE:*** *The keyboard shortcut does not work in the Preview PDF view.*

Keys for controlling palettes

Result	Action
Open palette menu.	Ctrl+Spacebar Ctrl+Spacebar may conflict with Microsoft Windows IMEA (Asian character input system) if configured. By default, Windows IMEA uses Crtl+Spacebar. You can change the keyboard mapping for IMEA in Windows or change the keyboard shortcut for the Palette Menu in Designer.
Restore the size of a docked or undocked palette.	Ctrl+Spacebar, R

Result	Action
Move a palette	Crtl + Spacebar, M Press arrow keys to position palette. Press Ctrl + Enter to end movement and undock palette. Press Enter to end movement and dock palette. Palette docks if moved in a docking area. Palette frame changes thickness when palette enters a docking area. Press ESC to exit move mode and return palette to original position.
Size a palette.	Crtl + Spacebar, S Press arrow keys to change palette size. Press Enter to complete sizing. Press ESC to exit sizing mode and return palette to original dimensions.
Dock a palette to the last docked location.	Ctrl + Spacebar, D
Undock a palette to the last floating location.	CTRL + Spacebar, U
Close a palette.	CTRL + Spacebar, C

Keys for navigating a form design

Result	Action
Move to the first page.	Ctrl+Home
Move to the last page.	Ctrl+End
Move to the next page.	Page Down
Move to the previous page.	Page Up
Switch between the form design and the first active palette.	F6
Switch to the next tabbed property page in the palette.	Ctrl+.
Switch to the previous tabbed property page in the palette.	Ctrl+,
Zoom in.	Ctrl+=
Zoom out.	Ctrl+-

Result	Action
Zoom to fit the actual size.	Ctrl+1
Zoom to fit the width of the page.	Ctrl+2
Zoom to fit to the page.	Ctrl+0

Keys for controlling menus

Result	Action
Displays palette menu.	Ctrl + M
Activate the menu bar and enter menu mode.	F10 or Alt
Activate the menu bar and open corresponding menu to the access key.	Alt+*[access keys on menu or selected dialog box]*
Display the shortcut menu (or context menu) for the selected item.	Shift+F10
Display the shortcut menu for the active child window.	Alt+Hyphen (-)
Display the shortcut menu for the active window.	Alt+Space
When in Menu mode (Alt+), carries out the corresponding command.	Access key
When in Menu mode (Alt+), moves to the one item above, or moves to the bottom of the menu if focus is located in the top of the menu.	Up Arrow
When in Menu mode (Alt+), opens menu item, moves to the one item below, or moves to the top of the menu if focus is located in the bottom of the menu.	Down Arrow
When in Menu mode (Alt+), opens the next menu to the left, or closes a submenu.	Left Arrow
When in Menu mode (Alt+), opens the next menu to the right, or opens a submenu.	Right Arrow

Keys for the Script Editor or XML Source Tab

Result	Action
Find next.	F3
Find previous.	Shift+F3
Insert an abbreviated reference syntax for an object on the form.	Ctrl+click the object you want to reference (while the Script Editor has focus).
Open the Find dialog box.	Ctrl+F
Open the FormCalc functions list in the Script Source field of the Script Editor.	F10
Open the Go To Line dialog box.	Ctrl+G
Open the Replace dialog box.	Ctrl+H
Turn on/off display of the Script Editor.	Ctrl+Shift+F5

Keys for manipulating objects

Result	Action
Align bottom.	Ctrl+Down Arrow
Align horizontal center.	Alt+Ctrl+Right Arrow
Align left.	Ctrl+Left Arrow
Align right.	Ctrl+Right Arrow
Align to grid.	Alt+Ctrl+Left Arrow
Align top.	Ctrl+Up Arrow
Align vertical center.	Alt+Ctrl+Up Arrow
Duplicate the specified object.	Ctrl+D
Expand and collapse an object in the Hierarchy palette.	Left Arrow or Right Arrow
Insert an object.	Alt+I. Press Arrow keys to select object, and then press Enter
Keep drawing tool.	Ctrl+K
Nudge down by 10 units.	Shift+Down Arrow

Result	Action
Nudge down one unit.	Down Arrow
Nudge left by 10 units.	Shift+Left Arrow
Nudge left one unit.	Left Arrow
Nudge right by 10 units.	Shift+Right Arrow
Nudge right one unit.	Right Arrow
Nudge up by 10 units.	Shift+Up Arrow
Nudge up one unit.	Up Arrow
Rename an object using the Hierarchy palette.	Ctrl+F2
Rename the selected object.	F2
Select an object using the Hierarchy palette.	Up Arrow or Down Arrow
Select multiple objects adjacent to each other in the Hierarchy palette.	Shift+Up Arrow or Shift+Down Arrow
Select multiple objects that are not adjacent to each other in the Hierarchy palette.	Ctrl+Up Arrow or Ctrl+Down Arrow, and then Spacebar
Turn on/off display of the Accessibility palette.	Shift+F6
Turn on/off display of the Border palette.	Shift+F8
Turn on/off display of the Font palette.	Shift+F4
Turn on/off display of the Hierarchy palette.	Shift+F11
Turn on/off display of the Layout palette.	Shift+F9
Turn on/off display of the Object Library palette.	Shift+F12
Turn on/off display of the Object palette.	Shift+F7
Turn on/off display of the Paragraph palette.	Shift+F5
Turn on/off display of the Script Editor.	Ctrl+Shift+F5

Keys for changing the tabbing order

Result	Action
Edit tabbing order for this item	F2
Copy	Ctrl+C

Result	Action
Paste	Ctrl+V
Move up in the tabbing order	Ctrl+Up Arrow
Move down in the tabbing order	Ctrl+Down Arrow

Keys for selecting and navigating within a table

Result	Action
Extend the selection to the beginning of the row.	Shift+Home
Extend the selection to the beginning of the table.	Ctrl+Shift+Home
Extend the selection to the end of the row.	Shift+End
Extend the selection to the end of the table.	Ctrl+Shift+End
Extend the selection.	Shift+Left Arrow, Right Arrow, Up Arrow, or Down Arrow
Move between cells.	Left Arrow, Right Arrow, Up Arrow, or Down Arrow
Select all the cells in a column.	Ctrl+Spacebar
Select all the cells in a row.	Shift+Spacebar
Select all the cells in the table.	Ctrl+Shift+Spacebar
Select only the active cell, when multiple cells are selected.	Shift+Backspace
Select the first cell in the row.	Home
Select the first cell in the table.	Ctrl+Home
Select the last cell in the row.	End
Select the last cell in the table.	Ctrl+End
Toggle text edit mode on/off in the active cell.	F2

Keys for navigating the Designer Help

Result	Action
Close the Help Viewer.	Alt+F4
Display Designer Help.	F1
Display the Contents tab.	Alt+C
Display the Index tab.	Alt+N
Display the Options menu. The commands for all of the buttons in the Help Viewer toolbar are available on the Options menu.	Alt+O
Display the Search tab.	Alt+S
Display the selected topic. Move back to the previous topic. Move forward to the next topic	Enter or Alt+D Alt+Left Arrow Alt+Right Arrow
Open and close a book.	Plus Sign and Minus Sign, or Left Arrow and Right Arrow
Select a keyword in the list.	Up Arrow and Down Arrow
Select a topic in the Search results list.	Alt+T, and then Up Arrow and Down Arrow
Select a topic.	Down Arrow and Up Arrow
Start a search.	Alt+L

21.2. Controlling the palettes with a keyboard

You can use a keyboard to restore, move, size, dock, undock, and close palettes.

To open a palette menu using the keyboard, tab to the palette and press Ctrl + Spacebar.

- To restore the size of a docked or undocked palette, press Ctrl + Spacebar + R.
- To move a palette, press Ctrl + Spacebar + M. Press the arrow keys to position the palette. Press Enter to complete the move. Press ESC to exit move mode and return the palette to the original location.
- To size a palette, press Ctrl + Spacebar + S. Press the arrow keys to change the palette size. Press Enter to complete the sizing. Press ESC to exit sizing mode and return the palette to the original dimensions. When you size the width of a palette on the left or right docking areas, you resize the docking area and all palettes in the area. When you size the height of a palette in the top or bottom docking areas, you resize the docking area and all palettes in the area.
- To dock a palette, press Ctrl + Spacebar+ D. Palette moves to the last docked location.

- To undock a palette, press Ctrl + Spacebar + U. Palette moves to the last floating location.
- To close a palette, press Ctrl + Spacebar + C.

To size the Data View Info area

1) Select the Data View palette, and type Ctrl+M.
2) Select Show Info, by typing the letter S.
3) Press Ctrl+M.
4) Select Size Info Area, by typing the letter I.
5) Size the Info Area, using the Up Arrow or Down Arrow keys.
6) Press Enter to complete the sizing.

To size the Object Library Preview area

1) Select the Object Library palette, and type Ctrl+M.
2) Select Show Object Preview, by typing the letter S.
3) Type Ctrl+M.
4) Select Size Object Preview, by typing the letter I.
5) Size the Object Preview area, using the Up Arrow or Down Arrow keys.
6) Press Enter to complete the sizing.

To size the Fragment Library Preview area

1) Select the Fragment Library palette, and press Ctrl+M.
2) Select Show Fragment Preview, by typing the letter S.
3) Press Ctrl+M.
4) Select Size Fragment Preview, by typing the letter I.
5) Size the Fragment Preview area, using the Up Arrow or Down Arrow keys.
6) Press Enter to complete the sizing.

21.3. Customizing keyboard shortcuts

To create a customized keyboard shortcut set

A keyboard shortcut set is a named collection of keyboard shortcuts. You can create one or more customized keyboard shortcut sets to better suit your own needs or the needs of multiple users.

For example, if several people work with the same copy of Designer at different times, each person can create their own customized keyboard shortcut set. Having multiple sets makes it easier for everyone to

manage the various keyboard shortcuts they use without altering the (Default) set that other people may be more familiar with and prefer to use.

NOTE: You cannot directly change, replace, or delete the (Default) keyboard shortcut set. As a result, after you modify the (Default) set (or an existing customized set) in any way, Designer adds another instance of that set to the Set list with the word (Modified) appended to the set name. You can then rename the modified set as you want.

1) Select Tools > Keyboard Shortcuts.
2) In the Set list, select the keyboard shortcut set you want to modify. Use the (Default) set or an existing customized set.
3) Click Save As.
4) (Optional) In the Name box, type a name for the set. If you do not type a new name for the set, the word (Modified) remains appended to the existing set name.
5) Click OK.

To assign a keyboard shortcut to a command

You can assign a keyboard shortcut to a command. It is possible to assign one or more keyboard shortcuts to the same command. However, to avoid unnecessary duplications, it is recommended that you assign only one keyboard shortcut to a command.

If the keyboard shortcut you want to use is already assigned to another command, Designer prompts you to remove the keyboard shortcut from the other command and assign it to the command you have currently selected.

After you assign a keyboard shortcut to a command that previously had none, or assign an additional keyboard shortcut to a command that already has a default keyboard shortcut assigned to it, an asterisk (*) appears before the command name shown in the Command list to indicate the keyboard shortcuts associated with the command have changed.

NOTE: When you assign a keyboard shortcut to a command, you modify the keyboard shortcut set to which the command belongs, either the (Default) set or an existing customized set. As a result, Designer adds another instance of the keyboard shortcut set to the Set list with the word (Modified) appended to the set name to indicate the set has changed. To save the keyboard shortcuts you assign to commands, you must save the associated keyboard shortcut set.

If you do not save the changes you make to a keyboard shortcut set, Designer retains the changes until you select another set in the Set list. When you select another keyboard shortcut set, Designer prompts you to save the changes you previously made to the (Default) or customized set.

1) Select Tools > Keyboard Shortcuts.
2) In the Set list, select the set in which you want to save the keyboard shortcut.
3) In the Product Area list, select (All Commands) or select the specific menu that contains the command you want to access.
4) In the Commands list, select the command you want. Existing keyboard shortcuts are listed in the Current Shortcuts list.
5) (Optional) In the Current Shortcuts list, select any other shortcuts assigned to the command that you want to delete, and then click Remove.

6) In the New Shortcut box, type the shortcut you want to assign to the command, and then click Assign.
7) Click Save As to save the changes made to the keyboard shortcut set.
8) (Optional) In the Name box, type the name of the new set as needed. If you do not type a new name for the set, the word (Modified) remains appended to the existing set name.
9) Click OK.

To remove a keyboard shortcut from a command

You can remove a keyboard shortcut including the default keyboard shortcuts from any command.

For example, you may want to remove the keyboard shortcuts you no longer use or to remove a keyboard shortcut so that you can replace it with another one that you prefer.

If you remove a default keyboard shortcut from a command, an asterisk (*) appears before the command name shown in the Command list to indicate the keyboard shortcut associated with the command has changed.

NOTE: When you remove a keyboard shortcut from a command, you modify the keyboard shortcut set to which the command belongs, either the (Default) set or an existing customized set. As a result, Designer adds another instance of the keyboard shortcut set to the Set list with the word (Modified) appended to the set name to indicate the set has been changed. To save the changes you make, you must save the associated keyboard shortcut set.

If you do not save the changes you make to a keyboard shortcut set, Designer retains the changes until you select another set in the Set list. When you select another keyboard shortcut set, Designer prompts you to save the changes you previously made to the (Default) or customized set.

1) Select Tools > Keyboard Shortcuts.
2) In the Set list, select the set that contains the keyboard shortcut you want to remove.
3) In the Product Area list, select (All Commands) or select the specific menu that contains the command you want to access.
4) In the Commands list, select the command you want. Existing keyboard shortcuts are listed in the Current Shortcuts list.
5) In the Current Shortcuts list, select the keyboard shortcut you want to delete, and then click Remove.
6) (Optional) To rename the keyboard shortcut set, click Save As, and then in the Name box, type a name for the set as needed.
7) Click OK.

To verify the current keyboard shortcuts for a command

1) Select Tools > Keyboard Shortcuts.
2) In the Set list, select the keyboard shortcut set that contains the keyboard shortcuts you want to verify.

3) In the Product Area list, select (All Commands) or select the specific menu on which the command you want to access is located.

4) In the Commands list, select the command you want. Any keyboard shortcuts assigned to the command are listed in the Current Shortcuts list.

22. Menu, Command, Toolbar, and Dialog Box Reference

22.1. Menus and Commands

A menu displays a list of commands. Some of these commands have icons next to them so that you can easily associate the command with the corresponding button in the toolbar. Others have assigned accelerator keys, which are identified by an underline, and keyboard shortcuts that you can use to quickly perform common tasks.

In addition to the menus at the top of the Designer window and many of the palettes, context menus display commands relevant to the active palette or page.

To open a menu using the keyboard

1) To open a menu using the keyboard, press Alt + *[accelerator key]*, and then select a menu command by pressing the accelerator key for that command. For example, to open the Edit menu and select the Cut command, press Alt+E T. You can also use the shortcut key Ctrl+X to select the Cut command.

To display a context menu

1) Position the pointer over the palette or page.
2) Click the right mouse button. The context menu appears where you clicked.

Menu bar commands

File menu

The File menu provides commands for performing the primary operations in Designer.

New

Creates a new form design. If you are using the stand-alone version of Designer, click the New button list on the Standard toolbar to quickly select the template on which to base the new form design. These templates are not available when Designer is integrated with Workbench.

Open

Opens an existing form design.

Close

Closes the active form design. You are prompted to save your changes.

Save

Saves the contents of the active form design.

Save As

Saves the contents of the active form design in one of the following file types: Adobe Static PDF Form (.pdf), Adobe Dynamic XML Form (.pdf), Adobe XML Form (.xdp), or Designer Template (.tds).

Publish to Repository

Saves the form design and its linked files in a shared folder or web folder where it can be accessed by many users or other applications.This command is available only in the stand-alone version of Designer.

Distribute Form

Closes the form in Designer and the Acrobat Distribute wizard appears to guide you through the distribution process.This command is available only in the stand-alone version of Designer.

New Data Connection

Establishes a data connection, which provides a description of the connection between the data source and the form design.

Form Object Collections

Creates, deletes, duplicates, renames, and modifies collections for signature fields and paper forms barcodes.

Form Properties

Displays the properties for the active form design.

Print

Prints the contents of the active Design View or Master Pages tab in the Layout Editor. You can also print the form with or without sample data.

Recent File List

Lists the recently opened form designs.

Exit

Closes Designer. You are prompted to save your changes.

Edit menu

The Edit menu provides commands for editing a form design.

Undo

Reverses up to 10 previous actions.

Redo

Reapplies up to 10 undone actions one at a time.

Cut

Deletes the current selection and moves a copy of it to the clipboard.

Copy

Adds a copy of the current selection to the clipboard.

Paste

Copies the contents of the clipboard onto the page.

Delete

Deletes the current selection without placing a copy on the clipboard.

Delete Page

Deletes the current page. This command only appears when the Design View tab is active.

Delete Master Page

Deletes the current master page. This command only appears when the Master Pages tab is active.

Duplicate

Duplicates the current selection without placing a copy on the clipboard.

Copy Multiple

Creates a specified number of copies of the selected objects.

Select All > Select All

Selects all top-level objects, groups, and subforms in the active Layout Editor tab. It also selects the contents of the Script Editor.

Select All > Select All Static Objects

Selects all static objects, including text, circles, lines, and rectangles.

Select All > Select All Fields

Selects all fields, including buttons, check boxes, date/time fields, decimal fields, signature fields, drop-down lists, image fields, list boxes, numeric fields, paper forms barcodes, password fields, radio buttons, and text fields

Find

Searches for every occurrence of a specific word or phrase in the XML Source tab or the Script Editor.

Find Next

Searches for the next occurrence of the specified word or phrase in the XML Source tab or the Script Editor.

Find Previous

Searches for the previous occurrence of the specified word or phrase in the XML Source tab or the Script Editor.

Replace

Finds and replaces text in the XML Source tab or the Script Editor.

Go To Line

Goes to the line number that you specify when you are in the XML Source tab or the Script Editor.

Lock > Lock Text

When enabled, you use a single-click to select an object and a double-click to edit captions. When disabled, you can single-click to select an object and select the caption directly.

Lock > Lock Static Objects

When enabled, static objects such as text, lines, images, and shapes cannot be selected or edited.

Lock > Lock Fields

When enabled, field objects cannot be selected or edited.

Replace Artwork

Replaces the background artwork in a PDF document with another PDF document, which you select. This command is unavailable when a PDF document does not contain background artwork.

Fragments > Create Fragment

Creates a new fragment from the selected objects.

Fragments > Edit Fragment

Opens the source file for the selected fragment reference for editing.

Fragments > Fragment Info

Displays information about the selected fragment or fragment reference. You can edit the information for a fragment, but you can only view the information for a fragment reference.

Fragments > Convert to Embedded Object

Converts the selected fragment reference to an embedded copy of the fragment. The embedded objects are no longer connected to the fragment source file and can be edited in the current form design.

Patterns > Display

Opens the Patterns dialog box on the Display tab.

Patterns > Edit

Opens the Patterns dialog box on the Edit tab.

Patterns > Validation

Opens the Patterns dialog box on the Validation tab.

Patterns > Data

Opens the Patterns dialog box on the Data tab.

Delete Leader

Removes the selected leader.

Delete Tab Stops

Opens the Delete Tab Stops dialog box.

Manage Action Profiles

Switches to Workbench and opens the Manage Action Profiles dialog box. A default action profile is created for each form.This command is available only when using Designer with Workbench.

View menu

The View menu provides commands that change your workspace.

Rulers

Shows or hides the rulers.

Grids & Guidelines > Show Grid

Shows or hides the grid.

Grids & Guidelines > Snap to Grid

Snaps the object to the nearest grid point.

Grids & Guidelines > Show Guidelines

Shows or hides the guidelines.

Grids & Guidelines > Snap to Guideline

Snaps the object to the guidelines.

Objects > Object Boundaries

Shows or hides the boundary that appears around objects on the page.

Objects > Snap to Object

Snaps the edge of an object to the edge of another object.

Objects > Snap to Center of the Page

Snaps the center lines of an object to a line at the vertical or horizontal center of the page. The center ines are displayed only when you are positioning objects near them and Snap to the Center of the Page is enabled.

Keep Drawing Tool

When selected, you can draw as many copies of an object as you require without having to select it from the Object Library palette each time. To disable this feature, click the Keep Drawing Tool button again.

Object Editor

Shows or hides the Object Editor that appears around the selected object on the page. The Object Editor provides an easy way to access common properties.

Design View

Displays the Design View tab.

Master Pages

Displays the Master Pages tab.

XML Source

Displays the XML Source tab.

Preview PDF

Displays the Preview PDF tab.

Zoom

Zooms in to view details; zooms out to see more.

Actual Size

Sizes the page to 100%.

Fit Page

Sizes the page to fit in the Layout Editor.

Fit Width

Sizes the page to fit the width of the Layout Editor.

Zoom Area

Zooms in to the selected area on the page.

Show/Hide Tab Order

Shows or hides tabbing order numbers on the form.

Insert menu

The Insert menu provides commands for adding objects to a form design.

Floating Field

Inserts a floating field in the form design.

Current Page Number

Inserts the current page number of the finished form.

Number of Pages

Inserts the total number of pages making up the finished form.

Current Date/Time

Inserts the current date and time.

Viewer Locale

Inserts the value of the locale setting for the application that is processing the form.

Viewer Name

Inserts the name of the application that is processing the form.

Viewer Version

Inserts the version number of the application that is processing the form.

Current Sheet Number

Inserts the current sheet number in the selected text object.

Number of Sheets

Inserts a placeholder for the current number of sheets in the selected text object.

Leader

Opens the leader dialog box (just above the cursor position) where you can define and edit leader properties such as the end position (width), pattern, character spacing, and alignment.

Standard > [object]

Lists the drawing objects that you can add to the form. The objects in this command match the objects in the Standard category of the Object Library palette.

Custom > [object]

Lists the custom drawing objects that you can add to the form. The objects in this command match the objects in the Custom category of the Object Library palette.

Barcodes > [object]

Lists the barcode objects that you can add to the form. The objects in this command match the objects in the Barcodes category of the Object Library palette.

Wrap in Subform

Creates a new subform that includes the selected objects.

Unwrap Subform

Removes the subform, leaving its objects in the form design.

Wrap in Subform Set

Creates a new subform that groups two or more subforms so that they stay together.

Wrap in Choice Subform Set

Creates a new subform that groups two or more subforms so that they stay together, but only one will appear on the final form based on user-defined conditions.

Unwrap Subform Set

Removes the grouping of two or more subforms that were wrapped so that they would stay together.

Hyperlink

Opens the Insert Hyperlink dialog box, which you use to create hyperlinks that connect to external websites, email addresses, as well as PDF and HTML files.

Fragment

Inserts a reference to the selected fragment in the form design.

New Page

Adds a new page after the last page in the form design. This command only appears when the Design View tab is active.

New Master Page

Adds a new master page after the last master page in the form design. This command only appears when the Master Pages tab is active.

Table menu

The Table menu provides commands for working with tables.

Insert Table

Opens the Insert Table dialog box.

Insert > Columns to the Left

Inserts one or more columns to the left of the leftmost selected columns. The number of columns inserted matches the number of columns selected.

Insert > Columns to the Right

Inserts one or more columns to the right of the rightmost selected columns. The number of columns inserted matches the number of columns selected.

Insert > Rows Above

Inserts one or more rows above the selected row cells or rows. The number of rows inserted matches the number of rows selected.

Insert > Rows Below

Inserts one or more rows below the selected row cells or rows. The number of rows inserted matches the number of rows selected.

Insert > Section Above

Inserts one section above the selected row cells, rows, or sections.

Insert > Section Below

Inserts one section below the selected row cells, rows, or sections.

Delete > Rows

Deletes any row containing one or more selected cells.

Delete > Columns

Deletes any column containing one or more selected cells.

Delete > Sections

Deletes any section containing one or more selected cells.

Delete > Table

Deletes any table containing one or more selected cells.

Select > Cell

Selects the cell the cursor is in when in text edit mode.

Select > Row Cells

Selects all the cells in any row containing one or more selected cells. Selects only the row cells (Text objects). Does not select the row itself.

Select > Column Cells

Selects all the cells in any column containing one or more selected cells.

Select > Row

Selects any row containing one or more selected cells.

Select > Section

Selects any section containing one or more selected cells or rows.

Select > Table

Selects any table containing one or more selected cells, rows, or sections.

Merge Cells

Combines two or more adjoining cells in the same row into a single cell.

Split Cells Horizontally

Divides two or more combined cells in a row into separate cells.

Distribute Rows Evenly

Distributes selected rows evenly in one or more sections or one or more tables. All rows adjust to the height of the tallest selected row.

Distribute Columns Evenly

Distributes selected columns evenly in one or more sections or one or more tables. The total width of all the selected columns is divided evenly between the number of columns selected so that all columns are the same width. The table width does not change.

Group as Section

Groups one or more selected rows or sections into one section.

Group as Choice Section

Helps create a data choice table row section. Designer wraps the row in a section, sets the condition of the row set to Select One Subform From Alternatives, promotes the row's occurrence constraints to the table section, and changes the row's minimum count and maximum count settings to 1. This command is available when a single table row is selected.

Ungroup Section

Ungroups one or more selected rows or sections into separate rows or sections.

Convert To Table

Converts one or more selected objects into a table.

Convert To Subform

Converts one or more selected objects into a subform.

Go to Row

Opens the Go To Row dialog box.

Layout menu

The Layout menu provides commands for arranging objects in a form design.

Align > Left

Aligns the left edges of the selected objects to match the position of the object in the selection that was selected last.

Align > Right

Aligns the right edges of the selected objects to match the position of the object in the selection that was selected last.

Align > Top

Aligns the top edges of the selected objects to match the position of the object in the selection that was selected last.

Align > Bottom

Aligns the bottom edges of the selected objects to match the position of the object in the selection that was selected last.

Align > Vertical Center

Aligns the vertical midpoint of the selected objects.

Align > Horizontal Center

Aligns the horizontal midpoint of the selected objects.

Align > To Grid

Aligns the selected objects to the closest grid point.

Distribute > Across

Evenly distributes the selected objects horizontally.

Distribute > Down

Evenly distributes the selected objects vertically.

Distribute > In Rows & Columns

Aligns and evenly distributes the selected objects in rows and columns.

Center in Page > Horizontally

Centers the selected objects horizontally on the form design.

Center in Page > Vertically

Centers the selected objects vertically on the form design.

Make Same Size > Width

Changes the size of the selected objects to match the width of the object in the selection that was selected last.

Make Same Size > Height

Changes the size of the selected objects to match the height of the object in the selection that was selected last.

Make Same Size > Both

Changes the size of the selected objects to match the height and width of the object in the selection that was selected last.

Group

Combines the selected objects so that they function as a single unit. A group of objects can contain any number of objects and can be nested to any depth.

Ungroup

Breaks a group object into separate units.

Merge Selected Text Objects

Merges the contents of multiple text objects into a single object. The new object's size and position matches the total boundaries of the merged objects.

Merge as Caption

Merges the contents of a text object and a field object that has its caption set to none into a single object. The new object's caption becomes the text from the text object. The new object's size and position matches the total boundaries of the merged objects.

Wrap in New Radio Button Group

Wraps the selected radio button objects into a single object.

Merge Radio Button Groups

Merges selected radio button groups (or at least one selected radio button group and one or more selected radio buttons) into a single radio button group.

Bring to Front

Moves selected objects to the foreground.

Bring Forward

Moves selected objects forward relative to other objects that are in front of the selected objects.

Send Backward

Moves selected objects backward relative to other objects that are behind the selected objects.

Send to Back

Moves selected objects to the background.

Tools menu

The Tools menu provides commands for specifying options.

Check Spelling

Opens the Check Spelling dialog box from which you can correct individual occurrences or all occurrences of misspelled words in selected objects or in the entire form, add words to My Custom Dictionary, create a list of words to ignore, and display the Spelling panel in the Options dialog box.

Edit Dictionary

Opens the Edit Dictionary dialog box from which you can view the list of words available in My Custom Dictionary under individual or all languages, change the spelling of any of the listed words, as well as add words to and delete words from individual or all languages.

Check Script Syntax

Checks all of the scripts in a form for correct syntax and reports any errors on the Warning tab in the Report palette.

Macros

Lists the macros you can run. Available only if a macros folder exists in the Designer installation folder, and at least one JavaScript file (JS) is in a subfolder of the macros folder.

Template Manager

Opens the Template Manager dialog box from which you can manage templates and set the default template.

Keyboard Shortcuts

Opens the Keyboard Shortcuts dialog box enabling you to customize keyboard shortcuts.

Options

Opens the Options dialog box from which you can define options for document handling, workspace, spelling, wizards and tips, bidirectional text, data binding, and international languages and measurements.

Missing Fonts

Opens the Missing Fonts dialog box where you can specify the appropriate font to use in the form design. This option is only available if the document you have opened uses one or more fonts that are not available on your system.

Window menu

The Window menu provides commands for showing or hiding form designs and palettes in Designer.

Arrange > Cascade

Displays the open form designs in cascading order.

Arrange > Tile

Displays all open form design windows.

Arrange > Arrange Icons

Arranges minimized windows at the bottom of the Designer window.

Workspace > Top Palettes

Shows or hides palettes docked at the top of the Designer window.

Workspace > Bottom Palettes

Shows or hides palettes docked at the bottom of the Designer window.

Workspace > Left Palettes

Shows or hides palettes docked at the left of the Designer window.

Workspace > Right Palettes

Shows or hides palettes docked at the right of the Designer window.

Workspace > Floating Palettes

Shows or hides floating palettes.

Workspace > Collapse/Expand Palettes

Shows or hides all palettes.

Workspace > Reset Palette Locations

Resets the palettes to their default locations.

Toolbars

Shows or hides the toolbar.

Script Editor

Shows or hides the Script Editor.

Status Bar

Shows or hides the status bar.

Hierarchy

Shows or hides the Hierarchy palette.

PDF Structure

Shows or hides the PDF Structure palette.

Data View

Shows or hides the Data View palette.

Tab Order

Shows or hides the Tab Order palette.

Object Library

Shows or hides the Object Library palette.

Fragment Library

Shows or hides the Fragment Library palette.

Style Catalog

Shows or hides the Style Catalog palette.

Layout

Shows or hides the Layout palette.

Border

Shows or hides the Border palette.

Object

Shows or hides the Object palette.

Accessibility

Shows or hides the Accessibility palette.

Font

Shows or hides the Font palette.

Paragraph

Shows or hides the Paragraph palette.

Drawing Aids

Shows or hides the Drawing Aids palette.

Info

Shows or hides the Info palette.

Report

Shows or hides the Report palette.

How To

Shows or hides the How To palette.

Switch to Workbench

Switches from Designer to Workbench. Designer remains open and you can switch back to it when required. This command is available only when using Designer with Workbench.

Palettes menu

The Palettes menu provides commands for showing or hiding form designs and palettes in Designer.

Workspace > Top Palettes

Shows or hides palettes docked at the top of the Designer window.

Workspace > Bottom Palettes

Shows or hides palettes docked at the bottom of the Designer window.

Workspace > Left Palettes

Shows or hides palettes docked at the left of the Designer window.

Workspace > Right Palettes

Shows or hides palettes docked at the right of the Designer window.

Workspace > Floating Palettes

Shows or hides floating palettes.

Workspace > Collapse/Expand Palettes

Shows or hides all palettes.

Workspace > Reset Palette Locations

Resets the palettes to their default locations.

Toolbars

Shows or hides the toolbar.

Script Editor

Shows or hides the Script Editor.

Hierarchy

Shows or hides the Hierarchy palette.

PDF Structure

Shows or hides the PDF Structure palette.

Data View

Shows or hides the Data View palette.

Tab Order

Shows or hides the Tab Order palette.

Object Library

Shows or hides the Object Library palette.

Fragment Library

Shows or hides the Fragment Library palette.

Layout

Shows or hides the Layout palette.

Border

Shows or hides the Border palette.

Style Catalog

Shows or hides the Style Catalog palette. Available when connected to a WebDav server.

Object

Shows or hides the Object palette.

Accessibility

Shows or hides the Accessibility palette.

Font

Shows or hides the Font palette.

Paragraph

Shows or hides the Paragraph palette.

Drawing Aids

Shows or hides the Drawing Aids palette.

Info

Shows or hides the Info palette.

Report

Shows or hides the Report palette.

How To

Shows or hides the How To palette.

Help menu

The Help menu provides commands for getting help and provides Designer version information.

Designer Help

Displays Designer Help topics.

Welcome

Shows or hides the Welcome screen.

How To

Shows or hides the How To palette.

Designer Quick Starts

Opens the Help topic About the Quick Start Tutorials, which describes the tutorials that are designed to guide you through creating basic forms with Designer.

Samples

Opens the Help topic About the Sample Forms, which describes the selection of complete sample forms included with Designer.

Scripting Reference

Shows the Scripting Reference. The Scripting Reference describes the objects, properties, and methods that you can use in the Designer scripting environment.

Designer Online

Displays the Designer product website.

Adobe Developer Connection

Displays the Adobe Developer Connection website.

About Designer

Displays the version of Designer and copyright information.

Palette menus

Hierarchy palette menu

The Hierarchy palette menu displays commands for working with the Hierarchy palette. You access this menu by clicking the palette menu button. You can also access the context menu for the Hierarchy palette when you click the right mouse button.

Help

Opens Designer Help and displays a description of the Hierarchy palette.

Hide Palette

Hides the palette.

Show Both

Displays both the name and caption of the object.

Show Script Indicators

Displays script indicator icons in the Hierarchy palette adjacent to the applicable objects.

Show Binding Indicators

Displays binding indicator icons in the Hierarchy palette adjacent to the applicable objects.

Data View palette menu

The Data View palette menu displays commands for working with the Data View palette. Access this menu by clicking the palette menu button. You can also access the context menu for the Data View palette when you click the right mouse button.

Help

Opens Designer Help and displays a description of the Data View palette.

Hide Palette

Hides the palette.

New Data Connection

Establishes a data connection, which provides a description of the connection between the data source and the form design.

Delete Data Connection

Removes the selected data connection from the form design.

Connection Properties

Displays properties of the current data connection; some of them can be modified.

Generate Fields

Creates new fields, subforms, and subform sets in the form design that correspond to the data hierarchy that is represented by the data connection.

Filter Data View

Lets you choose the nodes to display in the Data View palette.

Clear Data View Filter

Clears the filter that you applied to the Data View palette.

Options

Displays the options that are applied when you generate a field or group of fields by using the Generate Fields command.

Show Data Names

Displays the data name of the object which is the default.

Show Data Descriptions

Displays the data description of the object, up to a maximum of 25 characters. If the object has no description, the data name is displayed.

Show Both

Displays both the data name and the description of the object.

Show Info

Shows or hides the properties of the node that is selected in the Data View.

Refresh Data View

Refreshes all data connections in the current template.

Tab Order palette menu

The Tab Order palette menu displays commands for working with the Tab Order palette. You access this menu by clicking the palette menu button. You can also access the context menu for the Tab Order palette when you click the right mouse button.

Help

Opens Designer Help and displays a description of the Tab Order palette.

Hide Palette

Hides the palette.

Copy

Adds a copy of the current selection to the clipboard.

Paste

Copies the contents of the clipboard above the current selection in the list.

Edit Tab Order

Makes the tab order number beside the selected object editable. You can then type a new tab order number for the object.

Move Up

Moves the selected item or multiple adjacent items one line up in the tabbing order.

Move Down

Moves the selected item or multiple adjacent items one line down in the tabbing order.

Move First

Moves the selected item or multiple adjacent items to the beginning in the tabbing order.

Move Last

Moves the selected item or multiple adjacent items to the end of the tabbing order.

Make Sequential

Creates sequential tabbing order for multiple nonadjacent items selected in the list.

Make Geographic

Creates left to right, top to bottom tabbing order for the selected multiple items.

Make Geographic Right to Left

Creates right to left, top to bottom tabbing order for the selected multiple items.

Show Fields Only

Displays tabbing order only for the input or display fields.

Show Visual Aids

When a mouse cursor hovers over a field, displays blue arrows indicating tabbing order for the two preceding and the two following fields.

PDF Structure palette menu

The PDF Structure palette menu displays commands for working with the PDF Structure palette. You access this menu by clicking the palette menu button. You can also access the context menu for the PDF Structure palette when you click the right mouse button.

Help

Opens Designer Help and displays a description of the PDF Structure palette.

Hide Palette

Hides the palette, which you can display again from the Window menu.

Add Field to Structure

Creates a new field structure element from the selected field object in the Design View tab. Each field can only have one structure element in the structure view.

Remove Field from Structure

Removes the selected field structure element.

Previous Tag

Moves from the selected structure tag to the previous structure tag.

Next Tag

Moves from the selected structure tag to the next structure tag.

Move Field Up

Moves the selected field up one position under the current structure tag. If the selected field is the first field under the tag, moves it to the previous structure tag.

Move Field Down

Moves the selected field down one position under the current structure tag. If the selected field is the last field under the tag, moves it to the next structure tag.

Find Untagged Fields

Locates and identifies fields that do not have structure.

Show Context Tags

Toggles the display of context tags.

Style Catalog palette menu

The Style Catalog menu displays commands for working with Style Catalog palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Style Catalog palette.

Hide Palette

Hides the palette.

New Style Sheet

Designer opens a blank Designer Style Sheet (XFS) file in which to create new styles.

Add Style Sheet

Opens the Select Style Sheet dialog box.

Apply Default Styles To New Objects

Applies the default style to an object as you add it to your form design. Before you can apply default styles to new objects, select a default style for each object type you add to the form design.

Edit Default Style Settings

Opens the Default Styles dialog box.

Embed Style Sheets

Designer adds the styles from all of the external style sheet panels in the Style Catalog to the internal style sheet panel.

Sort By

Unsorts or sorts the Designer Style Sheet file (XFS) file listed in the Style Catalog by name or by type.

Layout palette menu

The Layout palette menu displays commands for working with the Layout palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Layout palette.

Hide Palette

Hides the palette.

Absolute Coordinates

Displays the X and Y coordinates relative to the upper- left corner of the subform. Absolute coordinates are most useful on forms that have a fixed layout.

Relative Coordinates

Displays the X and Y coordinates relative to the parent objects. Relative coordinates are most useful on forms whose layout adjusts to accommodate data.

Clear Fragment Property Overrides

Removes any overrides from the properties of the fragment reference.

Border palette menu

The Border palette menu displays commands for working with the Border palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Border palette.

Hide Palette

Hides the palette.

Clear Fragment Property Overrides

Removes any overrides from the properties of the fragment reference.

Object palette menu

The Object palette menu displays commands for working with the Object palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Object palette.

Hide Palette

Hides the palette.

Show Dynamic Properties

Turns the active labels for dynamic properties on or off. Active labels indicate which properties can be dynamic. They are identified by a green underline that you can click and dynamically bind the property to a data source.

Clear Fragment Property Overrides

Removes any overrides from the properties of the fragment reference.

Accessibility palette menu

The Accessibility palette menu displays commands for working with the Accessibility palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Accessibility palette.

Hide Palette

Hides the palette.

Show Dynamic Properties

Turns the active labels for dynamic properties on or off. Active labels indicate which properties can be dynamic. They are identified by a green underline that you can click and dynamically bind the property to a data source.

Clear Fragment Property Overrides

Removes any overrides from the properties of the fragment reference.

Font palette menu

The Font palette menu displays commands for working with the Font palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Font palette.

Hide Palette

Hides the palette.

List Font Names in Their Font

Shows the font names using the font to display the name.

Currently Editing palette menu

The Currently Editing palette menu displays commands for working with the Font palette. You access this menu by clicking the palette menu button.

NOTE: The Currently Editing palette menu is available only when the entire object is selected and the Edit > Lock Text option is not selected.

Edit Caption and Value

Applies changes that you make in the Font palette to both the caption and the value.

Edit Caption

Applies changes that you make in the Font palette to the caption.

Edit Value

Applies changes that you make in the Font palette to the value.

Paragraph palette menu

The Paragraph palette menu displays commands for working with the Paragraph palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Paragraph palette.

Hide Palette

Hides the palette.

Currently Editing palette menu

[The Currently Editing palette menu displays commands for working with the Paragraph palette. You access this menu by clicking the palette menu button.

NOTE: The Currently Editing palette menu is available only when the entire object is selected and the Edit > Lock Text option is not selected.

Edit Caption and Value

Applies changes that you make in the Font palette to both the caption and the value.

Edit Caption

Applies changes that you make in the Font palette to the caption.

Edit Value

Applies changes that you make in the Font palette to the value.

Drawing Aids palette menu

The Drawing Aids palette menu displays commands for working with the Drawing Aids palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Drawing Aids palette.

Hide Palette

Hides the palette.

Decimal Inches/Centimeters/Millimeters/Points

Displays a value as n in. or n cm.

Lines per Inch/Centimeters/Millimeters/Points

Displays a value as n/in. or n/cm.

Guideline Sets > [guideline]

Adds horizontal and vertical guidelines.

Info palette menu

The Info palette menu displays commands for working with the Info palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Info palette.

Hide Palette

Hides the palette.

Report palette menu

You access this menu by clicking the palette menu button.

Warnings tab (Report palette)

Use this tab to view messages that are reported as you work in the form design. There are two types of messages:

Target

Appears when Designer detects a feature that is not supported in the Acrobat and Adobe Reader target version for the form.

Marker

Associated with warning markers, such as the yellow triangle or white “X” in the red circle, that appear when a problem occurs with an object.

Syntax checking warnings are also displayed in the Warnings tab.

You access this menu by clicking the palette menu button.

Options

Opens the Options dialog box where you select the types of warnings to show in the Warnings tab.

Form Properties

Opens the Form Properties dialog box where you specify a number of standard settings that Designer uses for each form design, including the Acrobat and Adobe Reader target version.

Help

Opens Designer Help and displays a description of the Info palette.

Binding tab (Report palette)

Use this tab to view lists of fields based on how you defined their binding data. For example, you can list only fields with Global Data Binding or only those with no data binding defined. This feature is especially useful on forms that have a large number of data bound fields.

You access this menu by clicking the palette menu button. Here are the lists of fields included:

- Fields with Normal Data Binding
- Fields with Data Binding by Reference
- Fields with Global Data Binding
- Fields with No Data Binding
- Fields with Import Data Binding
- Fields with Export Data Binding
- Unbound Data Connection Nodes
- Objects with Dynamic Property Binding

Help

Opens Designer Help and displays a description of the Info palette.

Log tab (Report palette)

Use this tab to view a log of actions reported by Designer and to view updates when you preview a form design by using the Preview PDF tab. You access this menu by clicking the palette menu button.

Clear Warning

Clears the Log tab.

Help

Opens Designer Help and displays a description of the Info palette.

RELATED LINKS:

Addressing warning messages in the Report palette

How To palette menu

The How To palette menu displays commands for working with the How To palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the How To palette.

Hide Palette

Hides the palette.

Object and Fragment Library palette menus

Object Library palette menu

The Object Library palette menu displays commands for working with the Object Library palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Object Library palette.

Hide Palette

Hides the palette.

Collapse All

Hides the objects in the Object Library palette lists.

Expand All

Shows the objects in the Object Library palette lists.

Show Object Preview

Shows or hides a preview pane of the selected object.

Preview Zoom > [%]

Displays the zoom level of the preview pane.

Add Group

Adds a new category to the Object Library palette where you can create and store objects.

Shared Library Location

Displays the shared library location.

Restore Default Objects for All Groups

When permitted, restores all objects that were located in the Object Library palette at install time.

Standard menu

The Standard menu displays commands for working with the Standard objects in the Object Library palette. You access this menu by clicking the menu button.

View > Large Icons

Displays the objects as large icons.

View > Small Icons

Displays the objects as small icons.

View > Sorted List

Displays the objects by name, in alphabetical order.

Move Object To > [tab]

Moves the selected object to the Object Library palette list that you select.

Remove Object from Library

Removes the selected object from the Object Library palette.

Object Info

Displays object information. If modifying the object is permitted, you can edit object information using this command.

Remove Group

Removes the selected category and its objects from the Object Library palette.

Group Properties

Displays the properties of the selected category in the Object Library palette. You cannot edit the properties of a shared library.

Restore Default Objects

When permitted, restores the objects that were located in the Standard category at install time and adds them to the current list.

Custom menu

The Standard menu displays commands for working with the Custom objects in the Object Library palette. You access this menu by clicking the menu button.

View > Large Icons

Displays the objects as large icons.

View > Small Icons

Displays the objects as small icons.

View > Sorted List

Displays the objects by name, in alphabetical order.

Move Object To > [tab]

Moves the selected object to the Object Library palette list that you select.

Remove Object from Library

Removes the selected object from the Object Library palette.

Object Info

Displays object information. If modifying the object is permitted, you can edit object information using this command.

Remove Group

Removes the selected category and its objects from the Object Library palette.

Group Properties

Displays the properties of the selected category in the Object Library palette. You cannot edit the properties of a shared library.

Restore Default Objects

When permitted, restores the objects that were located in the Custom category at install time and adds them to the current list.

Barcode menu

The Barcode menu displays commands for working with the Barcode objects in the Object Library palette. You access this menu by clicking the menu button.

View > Large Icons

Displays the objects as large icons.

View > Small Icons

Displays the objects as small icons.

View > Sorted List

Displays the objects by name, in alphabetical order.

Move Object To > [tab]

Moves the selected object to the Object Library palette list that you select.

Remove Object from Library

Removes the selected object from the Object Library palette.

Object Info

Displays object information. If modifying the object is permitted, you can edit object information using this command.

Remove Group

Removes the selected category and its objects from the Object Library palette.

Group Properties

Displays the properties of the selected category in the Object Library palette. You cannot edit the properties of a shared library.

Restore Default Objects

When permitted, restores the objects that were located in the Barcode category at install time and adds them to the current category.

Fragment Library palette menu

The Fragment Library palette menu displays commands for working with the Fragment Library palette. You access this menu by clicking the palette menu button.

Help

Opens Designer Help and displays a description of the Fragment Library palette.

Hide Palette

Hides the palette.

Open Fragment Library

Displays the Browse for Fragment Library dialog box where you can select a folder containing fragments that you want to work with. The fragments in the folder appear in a new panel within the palette.

Expand All

Expands all panels to display the fragments contained within each one.

Collapse All

Collapses all panels to display only the panel names.

View > List

Panels display the names of the fragments.

View > Details

Panels display the names of the fragments plus a description file name of each fragment.

Show Fragment Preview

Shows or hides the preview pane, which shows a preview of the selected fragment.

Fragment Library panel menu

The Fragment Library panel menu displays commands for working with an individual panel within the Fragment Library palette. You access this menu by clicking the panel menu button.

Close [fragment] Library

Closes the current library and panel.

Edit Fragment

Opens the file that contains the selected fragment.

Fragment Info

Displays the name, description, and file location of the fragment.

Delete Fragment

Removes the selected fragment.

Template Manager menu

The menu that appears in the New and Template Manager dialog boxes provides commands for managing form templates.

Help

Opens Designer Help and displays a description of the dialog box.

View > Large Icons

Displays the objects as large icons.

View > List

Displays the objects by name, in alphabetical order.

Add Template

Opens the Add Template File dialog box where can locate a template to add to the dialog box.

Delete Template

Removes the selected template from the dialog box.

Rename Template

Renames the template in the dialog box.

Move Template To > [tab]

Moves the template to another tab in the dialog box.

Add Category

Adds a tab to the dialog box.

Remove Category

Removes the current tab from the dialog box. Not all tabs can be removed.

Category Properties

Displays tab properties, which can be edited.

Restore Default Templates

Restores the default templates to their original settings.

22.2. Toolbars

About toolbars

Toolbars contain buttons for commonly used commands. The following toolbars are installed with Designer:

- Standard
- Text Formatting
- Layout
- Styles
- Table
- Tools

You can modify the toolbars or create new ones to suit your requirements.

RELATED LINKS:

Standard toolbar

Text Formatting toolbar

Layout toolbar

Styles toolbar

Table toolbar

Tools toolbar

Standard toolbar

The Standard toolbar provides quick access to common commands in Designer.

New

Opens a new form design.

Open

Opens an existing form design.

Save

Saves the contents of the active form design.

Print

Prints the contents of the active Design View or Master Pages tab in the Layout Editor. You can also print the form with or without sample data.

Distribute Form

Closes the form in Designer and the Acrobat Distribute wizard appears to guide you through the distribution process. This command is available only in the stand-alone version of Designer.

Undo

Reverses up to 10 previous actions.

Redo

Reapplies up to 10 undone actions one at a time.

Find

Searches for every occurrence of a specific word or phrase in the XML Source tab or the Script Editor.

Show Grid

Shows or hides the grid.

Snap to Grid

Turns Snap to Grid on or off. If it is on, snaps the object being drawn or dragged to the nearest grid point.

Keep Drawing Tool

When selected, you can draw as many copies of an object as you require without having to select it from the Object Library palette each time. To disable this feature, click the Keep Drawing Tool button again.

Object Boundaries

Shows or hides the boundary that appears around objects on the page.

Actual Size

Sizes the page to 100%.

Fit to Page

Sizes the page to fit in the Layout Editor.

Fit to Width

Sizes the page to fit the width of the Layout Editor.

Zoom Out

Zooms out to see more.

Zoom

Zooms in to view details; zooms out to see more.

Zoom In

Zooms in to view details.

How To Palette

Shows or hides the How To palette.

Check Spelling

Performs spell checking of all or part of the current document.

Switch to Workbench

Switches from Designer to Workbench. Designer remains open and you can switch back to it when required.

Text Formatting toolbar

The Text Formatting toolbar provides quick access to common text and paragraph formatting commands.

Font (typeface)

Displays the font name for the selected text.

Font Size

Displays the font size for the selected text.

Bold

Applies or removes bold formatting to selected text.

Italic

Applies or removes italic formatting to selected text.

Underline

Applies or removes underline formatting to selected text.

Align Left

Aligns text to the left.

Align Center

Aligns text to the center.

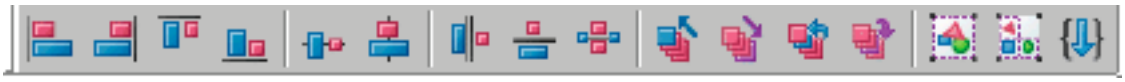
Align Right

Aligns text to the right.

Justify

Justifies the text.

Layout toolbar



The Layout toolbar provides quick access to common layout commands.

Align Left

Aligns the left edges of the selected objects.

Align Right

Aligns the right edges of the selected objects.

Align Top

Aligns the top edges of the selected objects.

Align Bottom

Aligns the bottom edges of the selected objects.

Align Vertical Center

Aligns the vertical midpoint of the selected objects.

Align Horizontal Center

Aligns the horizontal midpoint of the selected objects.

Distribute Evenly Across

Evenly distributes the selected objects horizontally.

Distribute Evenly Down

Evenly distributes the selected objects vertically.

Distribute Evenly in Rows and Columns

Aligns and evenly distributes the selected objects in rows and columns.

Bring to Front

Moves selected objects to the foreground.

Send to Back

Moves selected objects to the background.

Bring Forward

Moves selected objects forward relative to other objects that are in front of the selected objects.

Send Backward

Moves selected objects backward relative to other objects that are behind the selected objects.

Group

Combines selected objects so that they function as a single unit. A group of objects can contain any number of objects and can be nested to any depth.

Ungroup

Breaks a group object into separate units.

Merge

When two or more text objects are selected, merges the objects into a single object. When one or more radio buttons is selected, creates a new exclusion group containing the selected radio buttons.

Styles toolbar

The Styles toolbar provides quick access to common style commands.

Applied Style

Displays the name of the style applied to the currently selected object. Lists all available styles for the selected object type. Use this list to apply, change, or remove styles. If the style contains overrides, an asterisk character (*) is appended to the name.

Create New Style Sheet

Creates a new unnamed style sheet. When you save the style sheet the default file type is Designer Style Sheet (*.xfs).

Add New Style Sheet

Opens web browser to select and add a Designer Style Sheet (XFS) file to the current form.

Apply Default Styles To New Objects

Enables or disables the automatic application of default styles to new objects.

Table toolbar

The Table toolbar provides quick access to common table commands.

Insert Table

Opens the Insert Table dialog box.

Insert Rows Below

Inserts one or more rows below the selected row cells or rows. The number of rows inserted matches the number of rows selected.

Insert Columns to the Right

Inserts one or more columns to the right of the rightmost selected columns. The number of columns inserted matches the number of columns selected.

Merge Cells

Combines two or more adjoining cells in the same row into a single cell.

Split Cell Horizontally

Divides two or more combined cells in a row into separate cells.

Delete Rows

Deletes any row containing one or more selected cells.

Delete Columns

Deletes any column containing one or more selected cells.

Convert to Table

Converts one or more selected objects into a table.

Convert to Subform

Converts one or more selected objects to a subform.

Tools toolbar

The Tools toolbar provides quick access to common tools.

Select Tool

Selects the object.

Hand Tool

Moves the page so that you can view all areas of it. Moving a Designer page with the Hand Tool is like moving a piece of paper on a desk with your hand.

Field Edit Tool

Changes the object type.

Zoom Tool

Zooms in to view details.

Show/Hide Tab Order

Shows or hides the current tabbing order.

22.3. Dialog Box Reference

Action Builder dialog box

Use this dialog box to build, view, and edit actions.

To display this dialog box, select Tools > Action Builder.

Add a new action

Adds a new action to the Actions list. Click the default name (New Action) to rename action.

Remove an existing action

Deletes the action that is selected in the Actions list

Actions

Lists all actions in the form.

Condition

Adds or removes a condition. Click the add or delete buttons to add or remove conditions.

When object (link)

Opens the Select an Object dialog box, which lists every object in the form that you can select to build a condition.

Result

Adds or removes a result. Click the add delete buttons to add or remove results.

select a result

Lists the result options.

RELATED LINKS:

Build an action

Additional Assets dialog box

Use this dialog box to add, remove, and embed assets like video, sound, image, text, XML, and SWC files.

To display this dialog box, select a Flash Field object, in the Object palette, click the Field tab, and then click Edit.

Embed Asset

Embeds asset in form when selected. Assets are not embedded, by default. After adding a new asset, the Embedded column indicates No for the new asset.

Add

Opens the Browse for Additional Assets dialog box to enable file selection.

Delete

Remove the selected asset from the list.

Name

Lists the name of the asset.

URL

Lists the URL of the asset.

Embedded

Indicates Yes if an asset is embedded and No if an asset is not embedded.

RELATED LINKS:

Flash field

Using flash fields

Binding Properties dialog box

Use this dialog box to change a form object's properties when you associate a data binding with an existing field on the form design. You can associate a data binding with a form object by dragging a node from the Data View palette onto the object or by using the Binding tab in the Object palette. See *To create a bound field or subform*.

This dialog box is only displayed for form objects that support data binding. Notice that this dialog box is not displayed when setting the binding of radio button objects.

To display this dialog box, drag a node from the Data View palette onto an existing field on the form design or when you specify the binding node using the Binding tab in the Object palette.

Update all related properties

Updates all the possible properties of the form object to match the properties of the binding object.

If the form object type is different from the binding type, and you select Update All Related Properties, the form object type will change to match the binding type. For example, if the form object is a text field, and the binding object is a numeric field, selecting Update All Related Properties changes the form object to a numeric field. To control whether the object type is changed, use the option Update The Following Properties Only.

Update the following properties only

Displays a list of all the possible properties of the form object that can be updated by the properties of the binding object. You can select the specific properties that you want to update. Accepting all the options produces the same result as selecting Update All Related Properties.

The list of properties displayed varies according to the combination of form object type and binding. If the form object and binding object are the same, the following list of possible options may appear as updateable properties:

- Name
- Caption
- Descriptive Information
- Occurrence
- Validation Scripts

The Name and Caption properties are always displayed. The other properties vary according to what is appropriate for the form object and binding. For example, if the binding has validation script information, the Validation Scripts option is displayed.

Depending on the form object and binding combination, Designer also displays options related to the object's type. The following list of type-dependent options may appear as updateable properties:

- Data Pattern (Text Field and Date/Time Field only)
- Default Value (all fields except Image Field)
- Image URL (Image Field only)

- List Items (Drop-down List only)
- Max Chars (Text Field only)

  If the form object type is the same as the binding object type, Designer displays the type-dependent properties along with the other properties mentioned above. However, if the form object type is different from the binding object type, Designer displays the Type option, followed by all the type-dependent properties. In this situation, the Type option must be selected to update any of the type-dependent properties. If you do not select the Type option, none of the type-dependent properties will be updated.

  If you do not want the form object type to change, deselect the Type option.

Don't update any related properties

Specifies that no additional properties of the object, other than the binding itself, will be changed.

Don't Show Again

Stops the Binding Properties dialog box from displaying when you drag a node from the Data View palette onto the Layout Editor.

None of the object's properties will be updated to match the properties of the data item in the data source.

To start the Binding Properties dialog box displaying again, right-click the Data View palette, select Options, and select the Show Binding Properties Dialog option.

RELATED LINKS:

Binding fields to a data source

Check Spelling dialog box

Use this dialog box to correct individual occurrences or all occurrences of misspelled words in selected objects or in the entire form, add words to your custom dictionary, create a list of words to ignore, and open the Spelling panel in the Options dialog box.

NOTE: By default, Designer checks the spelling in all of the text associated with the various objects in a form, such as text objects, captions, tool tips, custom screen reader text, list box entries. To control the type of text that Designer checks, you can select options in the Spell Check list in the Spelling panel in the Options dialog box. For more information, see *Spelling (Options dialog box)*.

To display this dialog box, select Tools > Check Spelling.

Not in Dictionary

Displays misspelled words in context. For example, if the text for a tool tip contains a misspelled word, all of the text for the tool tip is displayed with the misspelled word highlighted (bold red). You can type directly in this box to make corrections. When you correct any of the text in this box, the bold red highlighting is removed from the misspelled word, and the Ignore Once, Ignore All, Add to Dictionary, Change All, and Undo buttons are unavailable.

Found In

Displays the reference syntax describing the object that you are spell checking.

Suggestions

Lists alternate spelling suggestions for the highlighted word.

Ignore Once

Ignores this occurrence of the misspelled word. Designer highlights any subsequent misspellings of the word.

Ignore All

Ignores all occurrences of the misspelled word. Adds the word to the list of ignored words so that Designer temporarily does not identify the word as being misspelled. All spell checks will disregard the word until you restart Designer. The list of ignored words is cleared each time Designer is started.

Add to Dictionary

Adds the highlighted word to all languages in My Custom Dictionary. Designer no longer identifies the word as being misspelled.

Change

Replaces the misspelled (highlighted) word with the word that is selected in the Suggestions list or typed in the Not in Dictionary box.

Change All

Replaces all occurrences of the misspelled (highlighted) word with the word that is selected in the Suggestions list.

Undo

Reverses the previous change(s).

Options

Opens the Spelling panel in the Options dialog box. Any changes that you make in the Spelling panel take effect immediately and apply to the current spell check. For example, if you clear the Tool Tip check box in the Spell Check list, Designer will not check any Tool Tip text.

Close

Closes the Check Spelling dialog box.

Collection Editor dialog box

Use this dialog box to select the objects to include in the collection for signature fields or paper forms barcodes.

To display this dialog box, click Modify in the Value tab in the Object palette (when the Paper Forms Barcode object is selected).

Name

Displays the name of the selected collection.

Collection Type

Specifies the type of collection that you are modifying. The two types of collection are Signature and Paper Forms Barcode.

The list shows the objects in a hierarchy where you can select and deselect the objects to include in the collection.

Hide unsupported nodes

Hides the objects that are not supported for the selected collection. Hiding the unsupported nodes makes it easier to locate the objects that you want to include in the collection.

RELATED LINKS:

To add a signature field

Signature field properties in the Signature tab

To control how a barcode obtains data

Using form object collections

Collection List dialog box

Use this dialog box to create, duplicate, delete, organize, and modify collections of objects for signature fields and paper forms barcodes.

To display this dialog box, click File > Form Object Collections.

Collections

Lists the collections for signature fields and paper forms barcodes. A collection applies to either signature fields or to paper forms barcodes, but not to both because they use different kinds of objects. The collections that are available for signature fields appear in the Signature tab in the Object palette. The collections that are available for paper forms barcodes appear in the Value tab in the Object palette.

New

Creates a new collection at the end of the Collections list.

Duplicate

Creates a copy of the selected collection in the Collections list.

Move Up

Moves the selected collection up in the Collections list.

Move Down

Moves the selected collection down in the Collections list.

Modify

Opens the Collection Editor dialog box so that you can modify the list of objects that are included in the collection.

RELATED LINKS:

To add a signature field

Signature field properties in the Signature tab

To control how a barcode obtains data

Using form object collections

Connection Properties dialog box

Use this dialog box to change the name of the selected data connection.

To display this dialog box, select a data connection and then select Connection Properties in the Data View palette menu.

Connection Name

Displays the data source connection name. You change the data source connection name in this box.

RELATED LINKS:

Working with Data Sources

To create a data connection to an XML schema

To create a data connection to an OLE database

To create a data connection using a WSDL file

Create Fragment dialog box

Use this dialog box to create a fragment from the selected objects.

To display this dialog box, select the object or objects that you want to include in a fragment and select Edit > Fragments > Create Fragment.

When you create a fragment in a AEM forms application, it is saved in the repository and appears in the Applications view in Workbench.

Name

Enter a name for the fragment. If the selection was a named subform or script object, the name of the object is the default value. If the selection was an unnamed object or if multiple objects were selected, the default value is Fragment <n> where <n> is a number.

Description

Optional. Type a description for the fragment

Create New Fragment in Fragment Library

Creates a fragment file within the specified library.

Fragment Library

The Fragment Library where the new fragment is saved. Select Open Fragment Library if the library where you want to save the fragment is not in the list.

File Name

The filename for the XDP file that contains the fragment. The default is the fragment name.

Replace selection with reference to new form fragment

Replaces the selection with a reference to the new fragment file. If the fragment file is later updated, the fragment reference will also be updated.

Create New Fragment in Current Document

Defines the selection as a fragment within the current file. No new fragment file is created. For example, select this option to create multiple fragments in the same XDP file and library.

Customize Toolbars dialog box

Toolbars (Customize Toolbars dialog box)

Use this dialog box to select the toolbars to show or hide, create new toolbars, and rename or delete user-created toolbars. You can also select whether to show or hide tool tips.

To display this dialog box, select Tools > Customize and click Toolbars.

Show/Hide Toolbars

Indicates which toolbars are shown and hidden. Select a check box beside the toolbars that you want to show.

Show Tool Tips

Indicates whether the tool tips are shown or hidden when the mouse pointer is positioned over a command in a toolbar.

New

Creates a toolbar. You customize the toolbar by adding and removing tools.

Rename

Renames the selected toolbar. You can only rename user-created toolbars.

Delete

Deletes the selected toolbar. You can only delete user-created toolbars.

Commands (Customize Toolbars dialog box)

Use this dialog box to add or remove commands and separators to a toolbar. You can also reset the toolbar back to its default setting.

To display this dialog box, select Tools > Customize and click Commands.

Show

Shows the list of commands available for selection as toolbar buttons. You can filter the list by type of command. Select All Commands to display all available commands in the list.

Add Tool to Toolbar

Click to add the selected command to the toolbar.

Add Separator to Toolbar

Click to add a separator to the toolbar. Separators are useful for grouping common commands on the toolbar. To remove a separator, select it and then click Remove Tool From Toolbar.

Remove Tool from Toolbar

Click to remove the selected command or separator from the toolbar.

Edit Toolbar

Indicates which toolbar you are currently editing. The list shows the buttons on the selected toolbar.

Reset Toolbar to Default

Restores the default buttons on the selected toolbar. Not available for user-created toolbars.

Move Tool Up

Click to move the selected command or separator up in the list.

Move Tool Down

Click to move the selected command or separator down in the list.

Data View Options dialog box

Use this dialog box to set the options that are applied when you generate a field or group of fields using the Generate Fields command.

To display this dialog box, select Options from the Data View palette menu.

Generate Validation Scripts

Generates validation scripts for new fields. The validation scripts generated are JavaScript scripts on the field. For example, a validation script can check that a field value is one from the allowed list

Generate Descriptive Information

Generates descriptive information.

Allow Tables To Be Generated

Creates a table when you drag a table node from the Data View palette onto the Layout Editor. If you deselect this option, Designer creates nested flowed or positioned subforms.

Create A Header Row For Any Table

Creates a header row for each table that is created.

Show Binding Properties Dialog

Shows the Binding Properties dialog box when you drag a node from the Data View palette onto the Layout Editor.

If you deselect this option, none of the object's properties will be updated to match the properties of the data item in the data source when you drag a node from the Data View palette onto the Layout Editor.

RELATED LINKS:

To set field generation options

Delete Tab Stops dialog box

Use this dialog box to remove unused tab stops.

To display this dialog box, select Edit > Delete Tab Stops.

Clear

Removes the tab stops that you select from the list.

Clear All

Removes all of the listed tab stops.

Digital IDs dialog box

Use this dialog box to select the digital ID used to access the Web Service Description Language (WSDL) document that you are using to create a data connection, or to add digital IDs to the Designer Digital IDs File store.

To display this dialog box, select Tools > Options > Security > Digital IDs.

Windows Certificate Store tab (Digital IDs dialog box)

Import Digital ID

Opens the Windows Import Certificate Wizard, where you specify the location of the digital file (*.p12 or *.pfx) to import into the Windows Certificate Store and enter the password required to access the file. See *Importing digital IDs*.

Remove Digital ID

Deletes the selected digital ID file from the list that is currently displayed (Windows Certificate Store or Designer Digital IDs File store).

View Digital ID

Lists the digital IDs that are available in the Windows Certificate Store.

Digital IDs Files tab (Digital IDs dialog box)

Add ID File

Opens the Add Digital ID dialog box, where you specify the location of the digital file (*.p12 or *.pfx) to add to the Designer Digital IDs File store and enter the password required to access the file. See *Importing digital IDs*.

Remove ID File

Deletes the selected digital ID file from the list that is currently displayed (Windows Certificate Store or Designer Digital IDs File store).

View Digital ID

Opens the Certificate Viewer dialog box, which displays general and detailed information about the selected certificate (digital ID).

RELATED LINKS:

Security (Options dialog box)

Importing digital IDs

To create a data connection using a WSDL file

Dynamic Properties dialog box

Use this dialog box to dynamically bind form object properties to values from a data source. This enables you to dynamically populate form object properties, such as the list items for a drop-down list or list box, with values from a data source.

To display this dialog box, ensure active labels are on, select the form object property that has dynamic data binding attached to it, and click one of the following active labels:

- List Items and Specify Item Values for a drop-down list or list box
- Caption
- Validation Pattern Message

- Validation Script Message
- Tool Tip
- Custom Screen Reader Text

Data Connection

Specifies the data connection you want to use for the dynamic data binding. If you are not already connected to a data source, Default Data Binding is the only selection in the Data Connection list.

As with normal binding, you can specify one or more WSDL connection bindings.

Items

Specifies the set of nodes that will be used to populate the items of the list for drop-down lists or list boxes.

You can select <None> to clear the binding for the selected property. You can also type a string into the box instead of selecting it.

This option is available when you click the active label for List Items and Specify Item Values for Drop-down Lists and List Boxes. In addition, for OLEDB data connections, the Items box is not available. Use the Item Text and Item Value boxes to select the data base columns to use to populate the list.

Item Text

Specifies the user-friendly label that is displayed at run time. The bindings that you can pick are descendants of the set of nodes selected in the Items box.

You can select <None> to clear the binding for the selected property. You can also type a string into the box instead of selecting it.

This option is available when you click the active label for List Items and Specify Item Values for Drop-down Lists and List Boxes.

Item Value

Specifies the data value that is submitted to the data source. The bindings that you can pick are descendants of the set of nodes selected in the Items box. This box is optional. If you leave this blank, the Item Text is submitted to the data source.

You can select <None> to clear the binding for the selected property. You can also type a string into the box instead of selecting it.

This option is available when you click the active label for List Items and Specify Item Values for Drop-down Lists and List Boxes.

Binding

Sets the binding for the selected property. The bindings that you can pick are the ones available for the data connection you selected.

You can select <None> to clear the binding for the selected property. You can also type a string into the box instead of selecting it.

This option is available when you click the active label for Caption, Validation Pattern Message, Validation Script Message, Tool Tip, and Custom Screen Reader Text.

RELATED LINKS:

Dynamically populating form object properties from a datasource

Edit Conditional Breaks dialog box

Use this dialog box to create conditional break statements for subforms, subform sets, tables, header rows, body rows, and sections. Conditional breaks are more flexible than traditional breaks and allow for multiple breaks of a single object.

To display this dialog box, perform one of the following actions:

- Select a subform object and click Edit on the Pagination tab of the Object palette. Note that the subform object you select must exist on the form design within a flowed subform object.
- Select a subform set object and click Edit on the Pagination tab of the Object palette.
- Select a table object and click Edit on the Pagination tab of the Object palette. Note that the table object you select must exist on the form design within a flowed subform object.
- Select a table header row and click Edit on the Pagination tab of the Object palette. Note that the table associated with the header row must exist on the form design within a flowed subform object.
- Select a table body row and click Edit on the Pagination tab of the Object palette. Note that the table associated with the body row must exist on the form design within a flowed subform object.
- Select a table footer row and click Edit on the Pagination tab of the Object palette. Note that the table associated with the footer row must exist on the form design within a flowed subform object.
- Select a table section and click Edit on the Pagination tab of the Object palette. Note that the table section you select must exist on the form design within a flowed subform object.

Add

Adds a new breaking condition for the current object. If a conditional break is currently selected, a new entry is created prior to the selected entry.

Delete

Removes the currently selected breaking condition for the object.

Move Up

Moves the currently selected conditional break up in the list.

Move Down

Moves the currently selected conditional break down in the list.

When

Specifies an expression indicating when a conditional break should occur.

Use the Insert Sample Expression button to select a valid object to use as the breaking condition. Designer inserts a sample statement which forces a conditional break if it evaluates to true.

The Insert Sample Expression button is disabled unless there are valid objects on the form design.

Language

Specifies which scripting language you want to use for the current calculation or script. There are two options:

FormCalc

A native Adobe calculation language ideal for shorter scripts.

JavaScript

A powerful and flexible scripting language for more complex scripts.

The scripting language that is displayed in the Language list matches the scripting language option you select as the default for new forms in the Workspace panel in the Options dialog box. However, if you change the scripting language setting for the current form on the Defaults tab in the Form Properties dialog box, each new conditional break in the Edit Conditional Breaks dialog box will use that same language.

Run At

Specifies where the calculation or script will execute. There are three options:

Client

Calculations and scripts execute while the client application (for example, Acrobat or web browser) processes the form.

Server

Calculations and scripts execute while the server application (for example, Forms processes the form.

Client and Server

Calculations and scripts execute while being processed by either the client or server application, depending on which application processes the form.

Break

Specifies where the conditional break should occur:

Before

Immediately before the current subform, subform set, table, header row, body row, footer row, or section.

After

Immediately after the current subform, subform set, table, header row, body row, footer row, or section.

To

Determines where to place the remaining fields and data from the broken subform after the conditional break:

Content Area > [name_of_content_area]

Resume in the specified content area.

Top of Next Content Area

Resume at the top of the next content area.

Top of Content Area > [name_of_content_area]

Resume at the top of the specified content area.

Page > [name_of _page]

Resume on the specified page (introduces a page break if the page specified is not the current page).

Top of Next Page

Resume at the top of the next page (introduces a page break).

Top of Page > [name_of _page]

Resume at the top of the page, whenever the specified page is rendered (introduces a page break if one does not occur naturally).

Odd Page

Resume on the next odd page.

Top of Next Odd Page

Resume at the top of the next odd page (introduces a page break if one does not occur naturally).

Even Page

Resume on the next even page.

Top of Next Even Page

Resume at the top of the next even page (introduces a page break if one does not occur naturally)

No Break, Just Leader and Trailer

Place the object after the previous subform, subform set, table, header row, body row, or footer row. Only switch to a new content area if the subform, subform set, table, header row, body row, footer row, or section does not fit in the current content area.

Leader

Specifies a leader subform to use for the conditional break.

[name_of_subform/table]

Use this existing subform, subform set, table, header row, or body row as the overflow leader.

None

Do not use an overflow leader.

Trailer

Specifies a trailer subform to use for the conditional break.

[name_of_subform/table]

Use this existing subform, subform set, table, header row, or body row as the overflow trailer.

None

Do not use an overflow trailer.

RELATED LINKS:

Using subforms

Using subform sets

Using tables

To create a repeating subform

To control subform and subform set breaks by using conditional statements

To control table, header row, body row, footer row, and section breaks using conditional statements

Edit Data Nominated Subforms dialog box

Use this dialog box to control the display of subforms from a choice subform set or rows from a choice section.

To display this dialog box, perform one of the following actions:

- Select a choice subform set and click Edit Alternatives on the Subform Set tab of the Object palette.
- Select a choice section and click Edit Alternatives on the Section tab of the Object palette.

Choose Subform Whose Name Matches Data Element or Attribute

Matches a single subform from the choice subform set, or a row from the choice section, with a like-named data node from the data connection. Only the subform or row with a name matching one of the data nodes will appear in the form layout.

Choose Subform Using Expression

Associates a single subform from the choice subform set, or a single row form the choice section, with a user-defined data node from within the data connection. You can then use relational

scripting expressions, such as equality and inequality expressions, in the Expression field to indicate the conditions that must occur for the named subform or row to appear in the form layout. The first entry in the Alternative Subforms list with an expression that evaluates to true displays on the form.

Data Connection

Specifies the name of the current data connection:

Default Data Binding

No data connection is currently specified for the form design.

[name_of_data_connection]

The name of the data connection currently defined for the form design.

Add

Adds a new subform object to the choice subform set or row to the choice section. Adding a new subform or row adds a duplicate of the subform object currently selected in the Alternative Subforms list. If no subform or row is selected, then a new unnamed subform is added to the choice subform set or choice section.

Delete

Removes the currently selected subform or row object from the current choice subform set or choice section.

Move Up

Moves the currently selected Alternative Subforms entry up in the list.

Move Down

Moves the currently selected Alternative Subforms entry down in the list.

Name

Specifies the name of a subform object within the choice subform set or the name of a row within the choice section.

Binding

Specifies all available data binding options for the current data connection:

[name_of_data_node]

Binds the subform or row to this data node entry from within the data connection.

<None>

Disables data binding using a user-defined data node from the data connection. The subform or row is still bound to a like-named data node from the data connection, if any exists.

NOTE: You can also type a user-defined binding directly into the Binding field.

Expression

Specifies a relational scripting expression, such as equality and inequality expressions, to indicate the conditions that must occur for the named subform or row to appear in the form layout.

If no expression is specified, the conditional break is considered to have a value of true.

Language

Specifies which scripting language you want to use for the current calculation or script. There are two options:

FormCalc

A native Adobe calculation language ideal for shorter scripts.

JavaScript

A powerful and flexible scripting language for more complex scripts.

RELATED LINKS:

Using subforms

Using subform sets

Using tables

Section properties in the Section tab

Edit Dictionary dialog box

Use this dialog box to view the list of words available in your custom dictionary (My Custom Dictionary) under individual or all languages, change the spelling of any of the listed words, as well as add and delete words to and from individual or all languages.

To display this dialog box, select Tools > Edit Dictionary.

Language

Lists the available languages that you can select. The list of words associated with each language appears when the language is selected.

Word

Selected word from the list of words is displayed in this box, where you can correct or change the spelling. You can also type new words to add to My Custom Dictionary.

Add

Adds to My Custom Dictionary the new word that is typed in the Word box.

Delete

Deletes the selected word from the list of words associated with the selected language.

Done

Closes the Edit Dictionary dialog box.

Edit Fragment Identifier dialog box

Use this dialog box when you want to use a custom identifier instead of the default fragment name. In this dialog box, you can view or edit the fragment identifier.

To display this dialog box, select Custom from the Fragment Name list on the Subform tab of the Object palette.

Identify fragment by

Select the method you want to use to identify the fragment.

SOM expression

Type the SOM expression that identifies the fragment. It should uniquely identify the root node of the fragment content in the fragment file. For example, $template.form.#subform.MyFragment.

ID

Specifies the ID value that identifies the fragment. Most IDs are generated in Designer. The ID can be any valid XML ID string that is unique in the document that contains it.

Edit Hyperlink dialog box

Use this dialog box to edit URL and email hyperlinks.

To display this dialog box, select Insert > Hyperlink.

URL

Select to create a URL hyperlink.

Email

Select to create an email hyperlink.

Email address

Enter one or more email addresses.

Subject

Enter an applicable subject line (optional).

Encrypt Data and Submit Settings dialog box

Encrypt Data (Encrypt Data and Submit Settings dialog box)

Use this panel to select the form content to be encrypted before a user submits the form. To display this panel, perform one of the following actions:

- Select a Button object, on the Field tab select Submit, click the Submit tab, select Encrypt Submission, click Settings, and then click Encrypt Data.
- Select an Email Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Encrypt Data.
- Select an HTTP Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Encrypt Data.

Encrypt Data Only

Encrypts only the form data in the submission. This option is the default setting.

Encrypt Entire Submission

Encrypts the form data and any attachments, such as annotations and PDF documents.

RELATED LINKS:

Using buttons

About submitting data using a button

Encryption Algorithm (Encrypt Data and Submit Settings dialog box)

Use this panel to specify which encryption algorithm to use to encrypt form content.

To display this panel, perform one of the following actions:

- Select a Button object, on the Field tab select Submit, click the Submit tab, select Encrypt Submission, click Settings, and then click Encryption Algorithm.
- Select an Email Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Encryption Algorithm.
- Select an HTTP Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Encryption Algorithm.

Encryption Algorithm

Lists available encryption algorithms.

Require The Encryption Party To Use The Specified Encryption Algorithm

Indicates that the encryption party must use the specified encryption algorithm.

RELATED LINKS:

Using buttons

About submitting data using a button

Encryption Certificates (Encrypt Data and Submit Settings dialog box)

Use this panel to specify encryption certificates and subject distinguished names, which identify the parties that encrypt the document.

To display this dialog box, perform one of the following actions:

- Select a Button object, on the Field tab select Submit, click the Submit tab, select Encrypt Submission, click Settings, and then click Encryption Certificates.
- Select an Email Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Encryption Certificates.
- Select an HTTP Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Encryption Certificates.

Specify The Encryption Certificates

Lists the available encryption certificates.

Add

Opens the Select User Certificate dialog box, where you can browse for and add encryption certificates.

Remove

Removes the selected certificates from the list.

Details

Opens the Certificate Viewer, where you can view the details about the selected certificate. The details that are listed vary according to the installed Certificate Authority (CA) certificates.

Restrict Encryption To The Specified Certificates

Allows author to specify that the form can be encrypted only for the public encryption certificates selected in the table.

Specify The Subject Distinguished Name

Specify a subject Distinguished Name (DN) for the encryption certificate. A subject distinguished name in an encryption certificate has several attributes. An example of a distinguished name is O = My Company; OU = My Department; CN = My Name; and C = My Country. The attributes in this distinguished name are O for Organization; OU for Organizational Unit; CN for Common Name; C for Country.

Add

Adds a DN to the list.

Delete

Deletes the selected DN from the list.

Move Up

Moves the selected DN up the list.

Move Down

Moves the selected DN down the list.

Sort Ascending

Organizes the listed DN entries in an ascending order.

Sort Descending

Organizes the listed DN entries in an ascending order.

Restrict Encryption To Certificates With The Specified Subject Distinguished Name

Allows the author to specify that the content can be encrypted only for the public encryption certificates in the table that meet the criteria.

RELATED LINKS:

Using buttons

About submitting data using a button

Issuers and Policies (Encrypt Data and Submit Settings dialog box)

Use this panel to specify certificate issuers, as well as certificate policies and associated object identifiers (OIDs). To display this dialog box, perform one of the following actions:

- Select a Button object, on the Field tab select Submit, click the Submit tab, select Encrypt Submission, click Settings, and then click Issuers and Policies.
- Select an Email Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Issuers and Policies.
- Select an HTTP Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Issuers and Policies.

Certificate Issuers

Lists the certificates that identify the issuers who can provide encrypting parties with encryption certificates:

Add

Opens the Select Issuer Certificate dialog box, where you can browse for and add issuer certificates.

Remove

Removes the selected issuer certificates from the list.

Details

Opens the Certificate Viewer, where you can view the details about the selected issuer certificate. The details that are listed vary according to the installed Certificate Authority (CA) certificates.

Restrict Encryption To Certificates From The Specified Issuer

Allows the author to specify that the content can be encrypted only for public encryption certificates in the table that meet the criteria.

Certificate Policies And Associated Object Identifiers

Lists Object Identifiers (OIDs) representing certificate policies in a recipient encryption certificate that are acceptable for the context XML encryption.

Add

Adds an object identifier.

Delete

Deletes the selected object identifier.

Restrict Encryption To Certificates That Conform To The Specified Policies

Allows the author to specify that the content can be encrypted only for public encryption certificates in the table that meet the criteria.

RELATED LINKS:

Using buttons

About submitting data using a button

Key Usage (Encrypt Data and Submit Settings dialog box)

Use this panel to specify which key usage parameters an encryption certificate requires or excludes to be acceptable for XML encryption.

To display this dialog box, perform one of the following actions:

- Select a Button object, on the Field tab select Submit, click the Submit tab, select Encrypt Submission, click Settings, and then click Key Usage.
- Select an Email Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Key Usage.
- Select an HTTP Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Key Usage.

  The key usage settings represent the key usage parameters of a certificate. Each key usage check box enables or disables the associated drop-down list. In every drop-down list, you can select either Require Key Usage or Exclude Key Usage.

Require Key Usage

Indicates that the encryption certificate requires the specified key usage parameter.

Exclude Key Usage

Indicates that the encryption certificate excludes the specified key usage parameter.

Require The Signing Certificate To Satisfy The Key Usage Settings

Allows the author to specify that the content can be encrypted only for public encryption certificates that meet the criteria indicated in each key usage setting list.

RELATED LINKS:

Using buttons

About submitting data using a button

Extract Style Sheet dialog box

Use this dialog box to extract the styles from the Internal style sheet to an external style sheet file (XFS).To display this dialog box, click the menu on the Internal style sheet panel, and select Extract Styles To A New Style Sheet.

Save In

Specifies the location where you want to locate a file or folder.

File Name

Displays the name of the file you select.

Save As Type

Indicates Designer Style Sheet (*.xfs) as the default file type option for saving style sheet files.

Replace Internal Style Sheet With Reference To New Style Sheet

Replaces the internal style sheet with a reference to the new external style sheet.

File Import Options

Adobe Output Designer form (File Import Options dialog box)

Use this dialog box to set options for importing Adobe Output Designer form files.

To display this dialog box, select File > Open and then select an Output Designer form file to open.

Generate a trace log of the conversion process

Creates a temporary log file in system's temporary folder that records information about the conversion process, and can be read using any ASCII text editor. Select one of these options to specify where to locate the log file:

- Do not log: No log file is generated.
- \TEMP\ConvertIFD.log: Generates a log file, called ConvertIFD.log, in system's temporary folder.

- \TEMP\inputfile.log: Generates a log file in system's temporary folder using the same name as the input file. For example, if the input file is called ABC.ifd, the log file is called ABC.log. Creates a log file for each file imported.

Use Output Designer PDF configuration

Uses an Output Designer configuration file (pdf.ics) and a compiled version of the configuration file (pdf.icf) to generate a PDF form file.

Sync like named fields as global

Synchronizes field objects with matching names to use global data binding. Applies a global value to all field objects in the form with the same name.

Import fields only

Removes all un-named subforms from the hierarchy.

Microsoft InfoPath (File Import Options dialog box)

Use this dialog box to set options for importing Microsoft® InfoPath files.

To display this dialog box, select File > Open and then select an InfoPath file to open.

Prompt to select view during conversion

Prompts you for a view. If the InfoPath file has only one view, a prompt will not display even if you select this option.

View

Specifies the name of the view to convert. To convert the default view, leave the box blank. If the Prompt To Select View During Conversion option is selected, this option is ignored.

Import print settings

Imports headers and footers, page orientation, and page margins of the InfoPath form. If you select the Import Print Settings option, the page size and orientation are taken from the InfoPath form and you do not have to select the page size and orientation.

Page Size

Specifies the page size for the converted form. To convert the default page size, select Default.

Orientation

Specifies the page orientation for the converted form.

Generate a log file

When any option other than Do Not Log is selected, Digital Signatures creates a log file (*.log) using the name and location that you select. The log file records information about the conversion process and can be read using any ASCII text editor.

Embed images in XDP

When selected, converts images in the InfoPath file and embeds them in the converted form. If you leave the option deselected, Digital Signatures does not convert the images but inserts place-holders for them.

RELATED LINKS:

Importing InfoPath files

How Designer converts Microsoft InfoPath objects

Microsoft Word (File Import Options dialog box)

Use this dialog box to set options for importing Microsoft® Word Document files.

To display this dialog box, select File > Open and then select a Word file to open.

Password to open file

If the Word file requires that you enter a password to open the document, type the password into this box.

Password to unprotect file

If the Word file is protected (users can enter data only into designated areas) and requires that you enter a password to remove the protection, type the password into this box.

Ignore missing Fonts

Digital Signatures detects whether the Word file contains fonts that are not available on your computer. To deal with this issue, you can either install the missing fonts or set up appropriate font-substitution rules in Word. When this option is selected, Digital Signatures substitutes the missing fonts with an available font and all font-substitution error messages are suppressed.

Convert Images

When selected, converts images and embeds them in the converted form. If you leave the option deselected, Digital Signatures does not convert the images but inserts placeholders for them.

Generate Log File

When any option other than Do Not Log is selected, Digital Signatures creates a log file (*.log) in the location that you select. The log file records information about the conversion process and can be read using any ASCII text editor. The log file name includes the name of the form file.

RELATED LINKS:

Importing Word files

How Designer converts Word objects and settings

About reducing Word conversion problems

XForms Model (File Import Options dialog box)

Use this dialog box to set options for importing XForms Model files.

To display this dialog box, select File > Open and then select an XForms Model file to open.

Embed Schemas Inline

Embeds schema within the output XDP file if a schema is set for the XForms document.

Set Working Directory

Indicates where Designer checks for relative file path locations that may be included in the XForms document, such as data XML files or XML schema files. If you do not specify this folder, the default working directory will be the same location as the XForm that is being imported. This is not the location where log files or temporary instance data files are created.

Generate log file

Specifies where to locate a generated log file that contains conversion output messages. You can select one of these options:

- Do Not Log: no log file is generated.
- \TEMP\ConvertXForms.log: Generates a log file named ConvertXForms.log in the Temp folder in Documents and Settings\[username]\Local Settings\Temp (Windows XP) or Users\[username]\AppData\Local\Temp (Windows Vista). Each file is appended with a unique number, such as ConvertXF38512.log.

Form Properties dialog box

Info (Form Properties dialog box)

Use this dialog box to capture and view general information about the form design. For example, the File option displays the file name and path to the location of the form design.

If you are working with a PDF document that contains PDF artwork, the information in the Info tab is displayed as read-only and can only be edited in Acrobat.

To display this dialog box, select File > Form Properties and click Info.

File

Displays the file name and path to its location. The name is created when you save the form design.

Title

Specifies the name of the form design.

Description

Specifies a description of the form design.

Author

Specifies the creator of the form design.

Contact

Specifies additional contact information.

Department

Specifies the department in your organization that owns the form design.

Creation Date

Specifies the date the form design was created.

Version

Specifies the current version of the form design.

Version Date

Specifies the date of the current version of the form design.

Custom Properties

Specifies custom properties for the form (for example, a copyright URL).

RELATED LINKS:

Adding details about the form

Defaults (Form Properties dialog box)

Use this dialog box to specify a number of standard settings that Designer will use for each form design. If necessary, you can override these settings for an individual form design.

To display this dialog box, select File > Form Properties and click Defaults.

Choose Target Version

Specifies the default Acrobat and Adobe Reader target version in which the form will be run. Designer validates the form design as you work and displays a warning message on the Warnings tab in the Report palette if a feature that you add fails the validation for the selected Acrobat and Adobe Reader target version. You must also ensure that you have the version of Acrobat installed that matches the target version that you select; otherwise, you may get an error or a warning message when you click the Preview PDF tab. The target version you select is also displayed at the bottom of the Save As dialog box when you save a form.

Form Locale

Specifies the default language and country, or region for the form design, which defines the format and punctuation of date, time, date/time, numeric, and currency values displayed in the form. Locale options are organized first by language and then by country or region. The form locale also defines the language used by the spell-check feature. For forms that have print output for multiple locales, select the Viewer's system locale option.

Default Language

Specifies the default scripting language to use for the current form design. You can choose JavaScript or FormCalc. Keep in mind that the scripting language that is initially displayed matches the scripting language option you select as the default for new forms in the Workspace panel in the Options dialog box. To set the scripting language for new forms, use the Default Language For New Forms list in the Workspace panel in the Options dialog box. See *Workspace (Options dialog box)*.

Default Run At

Specifies the default location for running scripts on the template. These options are available:

Client

Indicates that scripts should be run on the client. This option is the default.

Server

Indicates that scripts should be run on the server. The results are then returned to the browser.

Client And Server

Indicates that scripts can run on both the client and the server.

RELATED LINKS:

Report palette

Run-time (Form Properties dialog box)

Use this dialog box to specify run-time scripting options for preserving scripting changes, and to control the availability of the JavaScript Debugger and the JavaScript Console.

To display this dialog box, select File > Form Properties and click Run-time.

Allow automatic saving of document changes to temporary file

Enables or disables AutoSave for the form in Acrobat or Adobe Reader X. Specifies that form content is periodically and automatically saved according to the Acrobat or Adobe Reader Save setting. Does not run preSave scripts. May reduce performance when saving the content of a large form

NOTE: Uers can save changes to a temporary file when working in Acrobat Reader, only if the appropriate permission are applied to the form in Acrobat Reader DC extensions.

Preserve scripting changes to form when saved

Specifies how scripting changes will be preserved in a form when it is saved

Automatically

Specifies that scripting changes in a form will be preserved automatically when the form is saved. If the form is certified, scripting changes will not be saved. If the form is uncertified, certification of the form will not be possible.

Manually

Specifies that scripting changes in a form will be preserved through scripting when the form is saved. You must use the delta script object to do this. The Manual option is useful for certified forms.

Enforce strict scoping rules in JavaScript

Applies strict scoping rules to JavaScript. Applying strict scoping rules enhances the performance of forms and makes script more portable. Performance enhancements apply to forms targeted for Acrobat 8.1 and later.

You should not apply strict scoping rules in forms that are targeted for versions of Acrobat earlier than 8.1. When you select strict scoping and then choose a target version earlier that Acrobat 8.1, the script may behave differently or not work.

Java Script Debugger

Sets the availability of the JavaScript debugger. Available only for forms targeting Acrobat X and Adobe Reader X.

Enable JavaScript debugging

Allows the user to enable the JavaScript debugger.

Disable JavaScript debugging

Allows the user to disable the JavaScript debugger.

Availability of JavaScript debugging is determined by PDF Viewer

Allows the user to assign JavaScript debugging to the PDF Viewer (default).

JavaScript Console

Sets the availability of the JavaScript console. Available only for forms targeting Acrobat X and Adobe Reader X.

Enable JavaScript console

Allows the user to enable the JavaScript console.

Disable JavaScript console

Allows the user to disable the JavaScript console.

Availability of JavaScript console is determined by PDF Viewer

Allows the user to assign the availability of the JavaScript console to the PDF Viewer (default).

For more information see “Debugging Calculations and Scripts” in *Scripting Basics*.

Formatting (Form Properties dialog box)

Use this dialog box to set hyphenation, and widow and orphan control options for the current form only.

To display this dialog box, select File > Form Properties > Formatting.

Words with at Least <x> letters

Specifies the minimum number of letters that a word must contain to be hyphenated. For example, if you enter a value of 7, only words that have seven or more letters are hyphenated. The minimum value is 2. The maximum value is 99.

After First <x> letters

Specifies the minimum number of letters in a word that must appear on a line before the hyphen. For example, if you enter a value of 2, at least the first two letters of a word appear on a line before the word is hyphenated. The minimum value is 1. The maximum value is 99. The value should be less than the value in the Words With At Least <x> Letters box.

Before Last <x> letters

Specifies the minimum number of letters in a word that must appear on the next line after the hyphen. For example, if you enter the value of 4, at least the last four letters of a word appear on the next line. The minimum value is 1. The maximum value is 99. The value should be less than the value in the Words With At Least <x> Letters box.

Hyphenate Capitalized Words

Hyphenates words that begin with a capital (uppercase) letter, such as the first word of a sentence.

Hyphenate Words in ALL CAPS

Hyphenates words that have all capital letters, such as acronyms.

Allow Hyphenation in Text and Field Captions

Allows hyphenation in text objects and in the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields.

Allow Hyphenation in Text Field Values

Allows hyphenation in the value area of text field objects (default text and text the form filler enters).

Hyphenate All

Hyphenates the text in text objects, the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, and the value area of text field objects (default text and text the form filler enters). This option is available when you select Allow Hyphenation in Text and Field Captions, or Allow Hyphenation in Text Field Values, or both.

Remove All Hyphenation

Removes all hyphenation from the text in text objects, the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields, and the value area of text field objects (default text and text the form filler enters).

Widow and Orphan Control

Controls widow and orphan lines in text and text field objects. A widow is the last line of a paragraph that appears by itself at the top of the next page. An orphan is the first line of a paragraph that appears by itself at the bottom of a page. The widow and orphan control ensures that a minimum of two lines from the paragraph are kept together.

RELATED LINKS:

Hyphenate text

Considerations for setting hyphenation

To set hyphenation for a new form

To control widow and orphan lines for a form

To remove widow and orphan control from a form

Default Fonts (Form Properties dialog box)

Use this dialog to inspect and modify the default font properties for the current form. To display this dialog box, click File > Form Properties and select Default Fonts from the list on the left.

Default Caption Font Properties

Sets the default font properties for captions in the current form.

Typeface

Sets the font type.

Size

Sets the size of the font.

Style

Set the style of the font

Apply Properties to Existing Captions

Applies the selected typeface, size, and style properties to the captions in the current form.

Default Value Font Properties

Sets the default font properties for values in the current form.

The font properties for values are the same as the default font properties for captions.

Reset to Default Properties

Restores the default font properties for captions and values in the current form.

RELATED LINKS:

Default Fonts (Options dialog box)

Caption Formatting (Style Editor dialog box)

Form Validation (Form Properties dialog box)

Use this dialog box to control how Acrobat displays validation message boxes, highlights failed or mandatory fields that contain invalid data or no data, and sets the focus on the first field that fails to validate when the user submits the form.

To display this dialog box, select File > Form Properties > Form Validation.

Show Dialog Messages

Displays options that allow you to configure how Acrobat displays validation message boxes.

Configure how Acrobat displays validation message boxes

Enables options that allow you to specify how Acrobat displays validation message boxes.

Show every message in its own message box, one after the other

Displays each new validation error message after the user closes the current message.

Combine the messages of all the failed fields into one message box

Combines validation errors into a single list.

Show the first failed field's message and suppress any other messages

Displays only the first validation error message.

Don't show any messages boxes at all

Suppresses all validation messages.

Color Failed Fields

Displays options that allow you to choose the border and background colors that are used to highlight fields that fail to validate.

Color fields that fail their validations

Applies the selected border and background colors to fields that fail to validate. This option also enables the Border Color and Background Color options.

Border Color

Opens color palette. Selected color is applied to the borders of fields that fail to validate.

Background Color

Opens color palette. Selected color is applied to the background of fields that fail to validate.

Color Mandatory Fields

Displays options that allow you to choose the border and background colors that are used to highlight mandatory fields that fail to validate. This option also enables the Border Color and Background Color options.

Color mandatory fields that are not filled-in

Applies the selected border and background colors to mandatory fields that fail to validate. This option also enables the Border Color and Background Color options.

Border Color

Opens color palette. Selected color is applied to the borders of mandatory fields that fail to validate.

Background Color

Opens color palette. Selected color is applied to the background of mandatory fields that fail to validate.

Set Focus

Set focus to the first field that failed the validation

Selects the first field that fails to validate.

RELATED LINKS:

Displaying validation errors in Acrobat

To validate user input

Server (Form Properties dialog box)

Use this dialog box to specify server and print caching properties for a form design. The print caching options improve the server-side processing and performance of the specified print output. However, selecting these options may increase the file size of form designs and degrade the client-side performance. Selecting multiple Print options will significantly increase the file size of a form design.

To display this dialog box, select File > Form Properties and click Server.

RELATED LINKS:

Creating Forms

Creating Forms for Optimal Performance

Variables (Form Properties dialog box)

Use this dialog box to specify string variables based on the form design. The variables are saved with the current form design.

To display this dialog box, select File > Form Properties and click Variables.

Variables

Displays all the variables in the current form design.

Add

Click this button to define a new variable. You must specify a name in the Variables list and the variable's value in the large edit box to the right.

Each variable in a form design must have a unique name. Variable names are not case-sensitive.

The variable name should not contain any spaces. As soon as you press Enter, Designer automatically positions the cursor in the large edit box to the right.

If a variable does not have a value, Designer does not save it with the form design beyond the current session.

Delete

Click this button to delete the variable currently selected in the Variables list.

For more information, see “variables” in the Designer *Scripting Reference*.

PDF Security (Form Properties dialog box)

Use this dialog box to limit access to the PDF form. You do this by specifying passwords and by restricting certain features such as printing and modifying the PDF form.

To display this dialog box, select File > Form Properties and click PDF Security.

Require a password to open the document

Indicates that the PDF form prompts the user for a password to open the form. You are prompted to set the password when you save the form design in PDF.

Clear Passwords

Removes the passwords required to open the PDF form and print and edit the document. You are prompted to set the password when you save the form design in PDF.

Use a password to restrict printing and editing of the document and its security settings

Indicates that the end user must provide the correct password in order to print and edit the PDF form. The end user is prompted to set the password when saving the form design in PDF format.

Printing Allowed

Indicates whether the user can print the form. If you want to allow printing, select the printing resolution from the list. Options include:

- Low Resolution (150 dpi)
- High Resolution

Changes Allowed

Indicates whether the user can make changes to the PDF form. If you want to allow changes, select the type of change that the user is allowed to make. Options include:

- Inserting, deleting, and rotating pages
- Filling in form fields and signing
- Commenting, filling in form fields, and signing
- Any except extracting pages

Enable copying of text, images, and other content

Indicates that the user can copy text, images, and other contents.

Enable text access for screen reader devices for users with vision impairment

Indicates that the user can access the document using a screen reader device.

Enable plaintext metadata

Indicates that the information entered in the Info tab will be saved with the PDF form.

RELATED LINKS:

Setting Security

Using PDF security options

PDF Print Options (Form Properties dialog box)

Use this dialog box to specify settings for printing a PDF form in the form design instead of placing the effort on users to properly configure the print job. You can also specify that the Print dialog box appears when the PDF form opens, allowing users to print the form immediately.

To display this dialog box, select File > Form Properties and click PDF Print Options.

Use these print settings for printing this PDF form

Includes options for printing the PDF form:

NOTE: Setting the following options is recommended only for form designs saved as Acrobat 8 (Static) PDF forms or Acrobat 8 (Dynamic) XML forms.

Number of Copies

Specifies the number of copies that will be printed.

Duplex Mode

Specifies the setting for duplex printing. The selected printer must support duplex printing if you select a duplex option.

Page Scaling

Specifies print settings in the form design instead of placing the effort on the user:

Use Acrobat/Reader Setting

Prints using the page scaling options selected in the Acrobat/Reader Print dialog box.

No Page Scaling

Sets page scaling to none.

Prevent User from Changing

Prevents users from changing page scaling.

Select Paper Source by Page Size

Selects the paper source based on the paper size.

Automatically print the form when it is opened

Includes options for printing the PDF form when it opens:

Print to the user's default printer

Prints the PDF form using the user's default printer, which can be a local or network printer.

Print to

Specifies the printer on which the PDF form prints.

RELATED LINKS:

Printing forms

Compatibility (Form Properties dialog box)

Use this dialog box to update forms that were created in Designer version 6 to version 7. Updating older forms will result in the shifting of some text, which may require manual adjustment. In addition, script handling of enter and exit events on drop-down list boxes is affected. If the older form will be viewed primarily with Acrobat 7.0.5 or Adobe Reader 7.0.5 or later, use this Compatibility dialog box to update the form to version 7.

To display this dialog box, select File > Form Properties and click Compatibility.

If your form was created in version 6, you will see buttons you can use to update to version 7.

Revert Rendering Model

Forces all versions of Acrobat to use the previous form rendering model that was available in earlier versions of Designer.

By default, Acrobat now uses a new form rendering model (Direct Rendering) for rendering PDF forms. The new form rendering model provides better performance than the previous form rendering model. However, the new form rendering model can cause differences in the appearance and performance of legacy forms that use the previous rendering model.

Revert Text Formatting or Update Text Formatting

If you open a form from an earlier version of Designer, the text formatting rules are different.

To use the current text formatting rules, click Update Text Formatting. Updating older forms will result in the shifting of some text, which may require manual adjustment.

To use the version 6 text formatting rules, click Revert Text Formatting.

Revert Event Model or Update Event Model

If you open a form from an earlier version of Designer, the behavior of the enter events on the form is different. In version 7, the first time a user clicks in a drop-down list, an enter event is sent with

the reenter property set to false. If the user clicks in the field again or presses the Enter key, another enter event is sent with the reenter property set to true.

To use the current event handling rules, click Update Event Model. If you want the enter event to occur each time a user clicks a drop-down list, update the event model.

To use the version 6 event handling rules, click Revert Event Model. For example, if you have a form that was created in Designer version 6, and you want to download data from the server only once on the first enter, you may want to keep the old event model or revert to the old event model.

Preview (Form Properties dialog box)

Use this dialog box to specify different options for previewing the form design.

To display this dialog box, select File > Form Properties and click Preview.

Preview Type

Specifies the type of form that you want to preview when using the Preview PDF tab:

Interactive Form

Select this option to view a form that contains fields for interactive data capture. When you preview the form design, the resulting form opens as an interactive PDF.

Print Form (One-sided)

Select this option to preview forms that will be printed on one side of the paper. When you preview the form design, all objects are non-interactive.

Print Form (Two-sided)

Select this option to preview forms that will be printed on both sides of the paper. When you preview the form design, all objects are non-interactive.

Generate Preview Data

Select this option to generate sample form data. When you preview the form design, it will be merged with the sample data. You can also indicate how many times to repeat a repeating subform in the previewed form design.

Data File

Specifies the data file to use when previewing how a form design will be merged with data. The data file must be in standard XML format. You can type the path to the data file in the Data File box or browse to the file.

Preview Adobe XML Form As

Specifies the preview format for forms saved as Adobe XML Form File (XDP). You can select one of the following options:

Static PDF Form

Sets the default preview format to Static PDF. The form design can contain only static elements. If your form design contains dynamic elements, they will be unavailable in the form. For example, if you have a form that contains a table in which the number of columns are fixed but the number of rows in the table will change depending on how much information is in the data source, only one row of data is shown in the Preview PDF tab if the minimum number of occurrences is set to 1.

Dynamic XML Form

Sets the default preview format to Dynamic PDF. The form design can contain dynamic elements. For example, if you have a form that contains a table in which the number of columns are fixed but the number of rows in the table will change depending on how much information is in the data source, all the rows of data are shown in the Preview PDF tab. You can also test adding a row to the table, which you could not do in a static PDF form.

RELATED LINKS:

To preview and test forms in the Preview PDF tab

To set preview options for an interactive form

To set preview options for a non-interactive form

Save Options (Form Properties dialog box)

Use this dialog box to specify different options to apply when saving a form design as a PDF.

NOTE: The options in this dialog box only apply when saving a form design as a PDF file. They do not apply to PDF and XDP form designs being rendered through Forms.

To display this dialog box, select File > Form Properties and click Save Options.

Generate Accessibility Information (Tags) For Acrobat

Embeds accessibility information within the saved PDF file. This option is only available when publishing in PDF.

Generate Log File When Saving

Creates a log file of the saved file. The log file is placed in the same directory as the saved form. This option is only available when publishing in PDF.

Embed Fonts

Embeds the form fonts in the saved form design. This option is only available when publishing in PDF.

RELATED LINKS:

Saving forms

Saving forms for Acrobat and Adobe Reader

Previewing and testing forms

Data Binding (Form Properties dialog box)

Use this dialog box to specify options for binding data in a form design.

To display this dialog box, select File > Form Properties and click Data Binding.

Allow binding to data not defined by the default data connection

Select this option to allow binding to data that is not defined in the default data connection (either an XML schema or Sample XML data). Data in fields bound to data nodes not defined by the data connection is saved when this option is enabled:

XML namespace

Select this option to save the undefined data in or export the undefined data to a separate name-space. The namespace is added to the data description. If you do not specify a namespace, the namespace information is inherited from the parents of the nodes that contain undefined data.

XML namespace prefix

(optional) Lets you specify a namespace prefix.

NOTE: Do not use the prefix dd. It is reserved for the data description namespace to use.

If you specify a namespace or a prefix that is invalid in XML, a warning appears next to the relevant text box.

Add data nodes inside the filtered data hierarchy

Specifies that the undefined data is bound relative to a single filtered root when the Data View is being filtered.

RELATED LINKS:

Binding to data not defined by the default data connection

Schema filtering

Fragment Information dialog box

Use this dialog box to view or edit information about a fragment. When you use this dialog box with a fragment source file, you can view and edit the information. When you use this dialog box with a fragment reference, you can only view the information.

To display this dialog box, perform one of the following actions:

- Select the fragment on the form and then select Edit > Fragments > Fragment Info.
- Select the fragment on the form and, in the Object palette, click Fragment Info.
- Right-click the fragment and select Fragments > Fragment Info.
- Click More under the details for the fragment in the Fragment Library palette.
- Select the fragment in the Fragment library, and then select Fragment Info in the Fragment Library Panel menu.

Name

The name of the fragment. The fragment must have a name.

Description

The description for the fragment.

Location

The file name and location of the fragment.

Import Options dialog box

Use this dialog box to set options for importing PDF files that were not created with Designer.

This dialog box is not available when Designer is integrated with Workbench.

To display this dialog box in the stand-alone version of Designer, select File > Open, select a PDF file, and click Open. If the New Form Assistant appears, click the Do Not Use Assistant link.

Create an Interactive Form with Fixed Pages

Preserves the appearance of PDF documents created in Acrobat and other applications available in the Open dialog box. Converts Acrobat form fields to the corresponding Designer library objects. After you import the content of a PDF document as artwork, you can place additional library objects on top of the artwork as needed. Use this option if you have a PDF document that was created in Acrobat or some other application that you want to use as the basis (background) for creating an interactive form with fixed pages in Designer

Create an Interactive Form with a Flowable Layout

Imports PDF documents so that the contents can be edited in Designer. Any file attachments and annotations are removed. Keeps text together in fewer objects. Use this option if you want to create an interactive form with a flowable layout, make substantial text changes to the imported PDF form, or use Designer to make ongoing updates to form. This option also supports word-wrapping, but line breaks may not be preserved. Some touchup may be required to restore the appearance of the original document.

Custom Options for Importing PDF

Includes options for importing text:

Place content In

Specifies where the content will be placed in the form design, either in a page or a master page.

Join text as

Specifies how the text will be converted into objects.

Tolerance for joining text

A high tolerance setting provides a higher degree of possible variations and will join more pieces of text together into an object. A high setting may combine pieces of text that were meant to be separate objects. A low setting will keep more text as separate objects.

Tolerance for joining text lines

A high tolerance setting provides a higher degree of possible variations and will join more lines of text together into an object. A high setting may combine lines of text that were meant to be separate objects. A low setting will keep more text as separate lines of text.

Tolerance for joining paragraphs

A high tolerance setting provides a higher degree of possible variations and will join more paragraphs together. A high setting may combine paragraphs that were meant to be separate. A low setting will keep more paragraphs as separate objects. A low setting may miss combining text that should be together in a paragraph.

Embed images

Converts images in the PDF file and embeds them in the converted form. You can select one of the supported formats. If you leave the option deselected, Designer does not convert the images but inserts placeholders for them.

Format

Lists the image file formats you can select.

Display a summary report when conversion issues are found

If conversion issues are found, related information is displayed in a dialog box before the conversion operation is completed. You can cancel the operation if required.

Generate a log file

Creates a temporary log file in system's temporary folder that records information about the conversion process, and can be read using any ASCII text editor. Select one of these options to specify where to locate the log file:

- Do not log: No log file is generated.
- \TEMP\ConvertPDF.log: Generates a log file, called ConvertPDF.log, in system's temporary folder.
- \TEMP\inputfile.log: Generates a log file in system's temporary folder using the same name as the input file. For example, if the input file is called ABC.pdf, the log file is called ABC.log. Creates a log file for each file imported.

Page Range

Includes options for indicating the number of pages to import.

All

Imports all the pages in the PDF.

Pages

Imports an individual page or a range of pages. For example, type 2,3,6 for individual pages or 2-5 for a range of pages. Individual pages and page ranges can be combined: 1,3,6-9.

RELATED LINKS:

Importing PDF files

How Designer converts PDF objects

About reducing PDF conversion problems

Importing PDF documents asartwork

Insert Fragment dialog box

Use this dialog box to select the fragment you want to insert into a form design.

Script fragments are inserted in the variables child under the root subform to ensure that they are the most visible among the existing scripts in a form design.

To display this dialog box, select Insert > Fragment.

File Name

The name of the file that contains the fragment.

Files of type

The type of file that contains the fragment. The default is XDP.

Select a fragment

Select the fragment you want to insert from the list. The list includes all fragments in the specified file.

Fragment Info

Displays the description associated with the selected fragment.

Fragment Preview

Displays a preview of the selected fragment.

Insert Hyperlink dialog box

Use this dialog box to create URL and email hyperlinks.

To display this dialog box, select Insert > Hyperlink

URL

Select to create a URL hyperlink.

Email

Select to create an email hyperlink.

Email address

Enter one or more email addresses.

Subject

Enter an applicable subject line (optional).

RELATED LINKS:

Hyperlinks

To insert a hyperlink

Insert Table dialog box

Use this dialog box to create a simple table or to create a table using the Table Assistant.

To display this dialog box, select Insert > Standard > Table, or drag the standard Table object from the Standard category in the Object Library palette into the Layout Editor. You can also open the Insert Table dialog box by selecting Table > Insert Table.

Create Simple Table

Creates a simple table using the selected options.

Columns

Adds the specified number of columns to a simple table.

Body Rows

Adds the specified number of body rows to a simple table.

Include Header Row in Table

Adds a header row to a simple table.

Include Footer Row in Table

Adds a footer row to a simple table.

Create Table Using Assistant

Opens the Table Assistant, which you can use to create a simple table or a table that adjusts, depending on the number of rows of data.

Don’t Show Again

Stops the Insert Table dialog box from displaying when you select Insert > Standard > Table or when you drag a Table object onto the Layout Editor. To start the Insert Table displaying again, select Tools > Options > Wizards and Tips > Show When Creating Tables.

RELATED LINKS:

Body Layout (Table Assistant dialog box)

To create a table using the Table Assistant

To create a simple table

Keyboard Shortcuts dialog box

Use this dialog box to create and manage keyboard shortcuts for the commands in Designer.

To display this dialog box, select Tools > Keyboard Shortcuts.

Set

Lists the (Default) and user-defined sets of keyboard shortcuts that you can use, modify and save, or delete. After any changes are made to a selected set, the word (Modified) is appended to the set name.

Save As

Opens the Save As dialog box, letting you name and save the selected keyboard shortcut set. You can rename any modified set to suit your needs. You cannot replace the (Default) keyboard shortcut set.

Delete

Deletes the selected set of keyboard shortcuts. You cannot delete the (Default) keyboard shortcut set.

Product Area

Lists the product areas (including menus and palettes) for which you can create one or more customized keyboard shortcuts.

Commands

Displays all available commands when All Commands is selected in the Product Area list but displays only the relevant commands when a specific menu or palette is selected. Commands are organized into product areas such as File, Edit, View, and Palettes.

Current Shortcuts

Lists the shortcut keys that are assigned to the command that is selected in the Commands list.

Remove

Removes the selected shortcut from the Current Shortcuts list. This option is unavailable when the command selected in the Commands list does not have any shortcut keys assigned to it.

New Shortcut

Box in which you type the new shortcut keys you want to assign to the selected command.

Assign

Assigns the new shortcut keys to the selected command.

Description

Provides a short description of the selected command.

RELATED LINKS:

Customizing keyboard shortcuts

Missing Fonts dialog box

Use this dialog box to select a substitute font for the fonts that are not available on your computer. After you select the substitute fonts or accept the default, the document will display using those fonts.

This dialog box displays when you open a form containing fonts that are not on your computer.

Font Substitution

Each missing font is displayed.

Change Substitute Font

If you do not want to use a default font, click this button to select a different one.

Permanently Replace Unavailable Fonts

By selecting this option, the available substitute fonts that you specify will permanently replace missing fonts.

Show This Only If Fonts Do Not Have A Substitution Defined

Shows the Missing Fonts dialog box if one or more fonts do not have a substitution defined.

Change Font Substitutions dialog box

Use this dialog box to accept the proposed font substitution or change it. It displays the missing font and the proposed substitute font.

To display this dialog box, select Tools > Options. Select Document Handling from the list on the left, and then click Modify Font Substitutions.

Font Substitution

Each missing font is displayed and the proposed substitute font displays beside it.

Change Substitute Font

If you do not like the proposed substitute font, click this button to select a different one.

Network Connection Not Found dialog box

Use the Network Connection Not Found dialog box to create a connection to the specified URL.

The Network Connection Not Found dialog box appears when Designer cannot detect an established network connection to the specified base URL. The folder that Designer is attempting to detect is displayed in the box. Click OK to continue and create a connection to the folder.

Copy

Copies the folder listed in the box.

New Data Connection dialog box

Authentication (New Data Connection dialog box)

Use this dialog box to apply HTTP/HTTPS or message-level (SOAP) or both types of authentication to the WSDL data connection you are creating, as well as to select the types of credentials to accept for verification.

To display this dialog box, select File > New Data Connection, and then create a WSDL data connection.

Requires HTTP/HTTPS Authentication

Indicates that an HTTP/HTTPS (transport layer) message requires client authentication to access the WSDL document and to connect to a secure web service. The authentication credentials for the transport layer are sent as part of the HTTP/HTTPS protocol header.

Accepts User Name and Password

Provides basic client authentication, where the user is prompted with a login dialog box. Passwords are not hashed (clear text) before they are sent to the server.

Accepts User Name and Password Digest

Provides an enhanced form of basic client authentication, where the user is prompted with a login dialog box. However, passwords are hashed before they are sent to the server.

Accepts Client Certificate

Provides digital ID client verification, where you are prompted with the Select Digital dialog box listing the available digital IDs, and select the correct digital ID to send to the server for authentication.

Requires Message-Level Authentication

Indicates that the message level (SOAP) requires client authentication that uses a security token (embedded in the SOAP message header) to connect to a secure web service. The credentials for the message layer (SOAP) are used to create a security User Token.

Accepts User Name and Password

Provides basic client authentication, where the user is prompted with a login dialog box, and the user name and password combination are not encrypted before they are sent to the server.

RELATED LINKS:

Working with Data Sources

Importing digital IDs

To create a data connection using a WSDL file

ADO Properties (New Data Connection dialog box)

Use this dialog box to specify the properties for determining characteristics of the database connection.

User

Specifies the user for authentication when connecting to the database.

Password

Specifies the password for authentication when connecting to the database.

Connect Timeout

Indicates how long to wait while establishing a connection before terminating the attempt and generating an error.

Query Timeout

Indicates how long to wait for the database to return the results of an SQL query before terminating the attempt and generating an error.

Delayed Open

When selected, Designer does not automatically connect the data source when loading the form. Select this option when the form requires input from the user before it tries to connect, for example, when it requires a user ID and password.

Cursor Type

Indicates the type of cursor used when opening a Record set object.

Cursor Location

Indicates the cursor location. You can select a client or server-side location.

Lock Type

Indicates the type of locking to use when the form is opened.

BOF Action

Indicates what action will be taken when moving to a record that moves beyond the first beginning of file (BOF) record of the data source. For example, stay on the invalid record, move back to the last good record, or wrap around.

EOF Action

Indicates what action will be taken when moving to a record that moves beyond the last end of file (EOF) record of the data source. For example, stay on the invalid record, move back to the last good record, or wrap around.

RELATED LINKS:

Working with Data Sources

To create a data connection to an OLE database

OLEDB Connection (New Data Connection/Connection Properties dialog box)

Use this dialog box to provide a description of the data connection between OLEDB and a form design.

To display this dialog box, select File > New Data Connection. Select OLEDB and click Next.

Connection String

Specifies the name of the connection string.

Table

Specifies the table.

Stored Procedure

Specifies the stored procedure.

SQL Query

Specifies the SQL query.

RELATED LINKS:

Working with Data Sources

To create a data connection to an OLE database

New Connection (New Data Connection dialog box)

Use this dialog box to define a connection between a data source and a form design.

To display this dialog box, select File > New Data Connection.

Name New Connection

Specifies the name that you assign to the data connection.

Get Data Description From

Shows the list of available data sources to choose from.

NOTE: The Adobe Data Model option is unavailable with Adobe static PDF forms that contain PDF artwork. See *Importing PDF documents as artwork*

RELATED LINKS:

Working with Data Sources

To create a data connection to an XML schema

To create a data connection to an OLE database

To create a data connection using a WSDL file

Sample XML Data (New Data Connection/Connection Properties dialog box)

Use this dialog box to provide a description of the data connection between the sample XML data and a form design.

To display this dialog box, select File > New Data Connection. Select Sample XML Data and click Next.

Select Sample XML Data File

Specifies the location and name of the XML file.

Transform Incoming Data

Transforms incoming data using the XLST file that you specify.

XLST File

Specifies the XLST file that is used to transform incoming data.

Transform Outgoing Data

Transforms outgoing data using the XLST file that you specify.

XSLT File

Specifies the XLST file that is used to transform outgoing data.

RELATED LINKS:

Working with Data Sources

To create a data connection to an XML schema

Operation (New Data Connection dialog box)

Use this dialog box to select a WSDL operation.

To display this dialog box, select File > New Data Connection, and then create a WSDL data connection.

Select An Operation

Lists the operations that you can select to create a WSDL data connection.

RELATED LINKS:

Working with Data Sources

To create a data connection using a WSDL file

WSDL File (New Data Connection dialog box)

Use this dialog box to specify the location and name of the Web Service Description Language (WSDL) file to use to create a data connection.

To display this dialog box, select File > New Data Connection. Select WSDL File and click Next.

WSDL File

Specifies the location and name of the WSDL file.

RELATED LINKS:

Working with Data Sources

To create a data connection using a WSDL file

Adobe Data Model (New Data Connection dialog box)

Use this dialog box to specify the location and name of the Adobe Data Model (FML) file to use to create a data connection. A JVM error can occur if a user tries to create a FML data connection in a form. To overcome this issue a machine restart is required, for Designer to read the value of `WB_JRE_HOME` variable.

To display this dialog box, select File > New Data Connection. Select Adobe Data Model and click Next.

Select Adobe Data Model

Specifies the location and name of the Adobe Data Model (FML) file.

RELATED LINKS:

Working with Data Sources

To create a data connection to an Adobe Data Model

XML Schema (New Data Connection/Connection Properties dialog box)

Use this dialog box to provide a description of the data connection between an XML schema and a form design.

To display this dialog box, select File > New Data Connection. Select XML Schema and click Next.

Select XML Schema File

Specifies the location and name of the XML schema file.

Use XML Data Root Element Name

Specifies the data root element that you want displayed in the Data View palette.

Embed XML Schema

Embeds the XML schema in the form design.

Transform Incoming Data

Transforms incoming data using the XLST file that you specify.

XLST File

Specifies the XLST file that is used to transform incoming data.

Transform Outgoing Data

Transforms outgoing data using the XLST file that you specify.

XSLT File

Specifies the XLST file that is used to transform outgoing data.

RELATED LINKS:

Working with Data Sources

To create a data connection to an XML schema

New dialog box

This dialog box is available only in the stand-alone version of Designer. Use this dialog box to create a new form, create a new form from a template, or open an existing sample form. You also have the option to explore the sample form templates.

To display this dialog box, with the New Form Assistant disabled, select File > New or click New Form or New From Template in the Welcome screen.

Blank tab

Displays the available template sizes.

Forms tab

Displays the available sample templates.

Forms - Vanilla tab

Displays templates that have the same content as the ones on the Forms tab, but use an alternative, "vanilla" layout.

Forms - Beige tab

Displays templates that have the same content as the ones on the Forms tab, but use an alternative, "beige" layout.

Forms - Blue tab

Displays templates that have the same content as the ones on the Forms tab, but use an alternative, "blue" layout.

Other tab

Displays templates that you have added to the template manager.

Set Selected As Default

Sets the selected template as the default template for creating new form designs.

Create

Specifies whether a new form or new template is created.

Open dialog box

Use this dialog box to open an existing form design or template. You can open documents from a variety of formats. Once opened in Designer, the documents are converted to the XDP file format.

To display this dialog box, select File > Open.

Look in

Specifies the location where you want to locate a file or folder.

File name

Displays the file name. The name is created when you save the template.

Files of type

Lists the available file types that you can open.

RELATED LINKS:

Opening forms

Options dialog box

Document Handling (Options dialog box)

Use this dialog box to modify the default file type for new forms, the recently used file list, compression and backup options, image file browser options, and font substitution options.

To display this dialog box, select Tools > Options and select Document Handling from the list on the left.

Default File Type For New Forms

Specifies the default file type for new forms. This setting affects the Save As dialog box default and the file format used in the Preview PDF tab if the form has never been saved.

Static PDF Form (*.pdf)

Saves forms as static PDF, based on the Acrobat and Adobe Reader target version specified. Static PDF forms render once and are displayed on the client in the Acrobat or Adobe Reader target version. They are not rerendered in response to user interaction. The PDF form may have been designed with a flowable layout, but when the static PDF form is created, its layout is fixed and the resulting PDF form will not rerender on the client. Static PDF forms can be interactive or non-interactive.

Dynamic XML Form (*.pdf)

Saves forms as dynamic PDF, based on the Acrobat and Adobe Reader target version. The form design can contain dynamic elements. Dynamic PDF forms render on the client in Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of an object are possible in Adobe Reader because Adobe Reader has enough infor-

mation to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If the end user clicks a button that adds a new row to a table, the form is rerendered in Adobe Reader.

Adobe XML Form File (.xdp)

Sets the default file type for new forms to the native XML-based file format created by Designer. Use this option if you will be using Forms. Notice that the XDP Preview Format setting in the Preview tab (Form Properties dialog box) controls the PDF used for Preview PDF for this file type.

Recently Used File List

Controls the number of files shown in the recently used file list. Type the number of file names to display. The maximum is 10.

Save PDF Files Compressed

Compresses PDF files when saved.

Create Backup Copy on Save

Creates a backup file when you save a form design. The backup copy is saved in the same location as the form design. _BAK is appended to the file name before the extension. The backup copy preserves the previous version of a file when you save, in case you need to restore the previous version.

Create Translation IDs when Saving

Creates identifiers (XLIFF) for the translatable text in the form design. XLIFF is an XML-based format for exchanging localization data. See *Prepare for translation* for more information.

Show Default File Browser

Specifies the file browser that you want to use as a default. You can load an external COM component (such as a file browser component) that looks in a specialized repository. If you have any such components registered on your computer, Designer detects them and displays them in this list.

Modify Font Substitutions

Defines font substitutions for documents that are missing fonts.

Show Missing Fonts dialog box only if fonts do not have a substitution defined

Displays the Missing Fonts dialog box if one or more fonts do not have a substitution defined.

Show Warning dialog box when Report palette is hidden

Displays the Target Version Warning dialog box when the Report palette is hidden.

RELATED LINKS:

Previewing and testing forms

Prepare for translation

Selecting the Acrobat and Adobe Reader target version

Workspace (Options dialog box)

Use this dialog to show or hide tool tips about objects, set Script Editor options, and set source view options.

To display this dialog box, select Tools > Options and select Workspace from the list on the left.

Display Position And Size Tool Tips While Editing

Displays the position and size tool tips.

Display Object Name Tool Tips While Pointing

Displays the tool tip for the object when your pointer is over it.

Display Keep Drawing Tool

Enables the Keep Drawing command on the View menu and adds the Keep Drawing button to the Standard toolbar.

Make Fixed-Size Text Objects Auto-Fit While Editing

Expands fixed-size objects to fit the amount of text. The object will grow horizontally until it reaches the right edge of the parent or the user types a carriage return, at which point the object will grow vertically.

FormCalc Syntax Formatting

Displays the font and color settings of FormCalc syntax in the Script Editor.

JavaScript Syntax Formatting

Displays the font and color settings of JavaScript syntax in the Script Editor.

Show Statement Completion Options

Displays the statement completion options in the Script Editor.

Add Statement Completion Method Signatures

Automatically appends the parameter signature to the script.

Show Line Numbers

Displays the line number in the Script Editor.

Default Language for New Forms

Sets the default scripting language for new forms. To set the scripting language for an individual form, use the Default Language list on the Defaults panel in the Forms Properties dialog box. See *Defaults (Form Properties dialog box)*.

XML Source Syntax Formatting

Displays the font and color settings of the XML syntax on the XML Source page.

Show Line Numbers

Displays the line number on the XML Source page.

Set Current Designer As Default

Sets Designer as the default version. The default version is the registered version of Designer. If you have an earlier version on your system and you start it, the previous version will self-register and becomes the default version.

Designer should check whether it is the default Designer on startup

Checks whether Designer is the default version when you start it. If it is not the default version, you can set it as the default.

RELATED LINKS:

To customize the tool tips

Syntax Formatting dialog box

Spelling (Options dialog box)

Use this dialog box to select options to check spelling while you type, control the type of text Designer checks for spelling errors, add and remove custom dictionaries, and open the Edit Dictionary dialog box to edit your Custom Dictionary.

To display this dialog box, select Tools > Options and select Spelling from the list on the left.

Check Spelling While Typing

Enables spell checking while you type. When this option is selected, a wavy colored line appears under misspelled words anywhere in Designer.

Wavy Underline Color

Opens a color palette where users can select an alternate color for the wavy line that appears under misspelled words. The default color is red.

Spell Check Options

Lists the types of text in a form that Designer can check the spelling of when you use the Check Spelling dialog box. All options are selected by default.

Perform Spell Check On

Lists the areas you can select to spell check.

Installed Dictionaries

Lists the dictionaries that are available for spell checking. Designer uses the selected dictionaries when spell checking a form. The first dictionary listed is always your (current user) custom dictionary (My Custom Dictionary). This dictionary is always selected for use and cannot be removed from the list.

Add a Dictionary

Opens the Custom Dictionary File dialog box from which you can locate and open the custom dictionary files (*.clam) that you want to add to the Dictionaries list.

Remove a Dictionary

Removes the selected dictionary from the list of installed dictionaries. Does not delete the actual dictionary file (*.clam) from your computer's file system.

Edit the Custom Dictionary

Opens the Edit Dictionary dialog box from which you can view the list of words available in My Custom Dictionary under individual or all languages, change the spelling of any of the listed words, as well as add words to and delete words from individual or all languages. You cannot edit the contents of the other installed dictionaries.

Formatting (Options dialog box)

Use this dialog box to set hyphenation, and widow and orphan control options for all new forms.

To display this dialog box, select Tools > Options > Formatting.

Use Style Sheet

Specifies the location of the Designer Style Sheet file (XFS) to use as the default style sheet for new forms.

Widow and Orphan Control

Controls widow and orphan lines. A widow is the last line of a paragraph that appears by itself at the top of the next page. An orphan is the first line of a paragraph that appears by itself at the bottom of a page.

Words with at Least <x> letters

Specifies the minimum number of letters that a word must contain to be hyphenated. For example, if you enter a value of 7, only words that have seven or more letters are hyphenated. The minimum value is 2. The maximum value is 99.

After First <x> letters

Specifies the minimum number of letters in a word that must appear on a line before the hyphen. For example, if you enter a value of 2, at least the first two letters of a word appear on a line before the word is hyphenated. The minimum value is 1. The maximum value is 99. The value should be less than the value in the Words With At Least <x> Letters box.

Before Last <x> letters

Specifies the minimum number of letters in a word that must appear on the next line after the hyphen. For example, if you enter the value of 4, at least the last four letters of a word appear on the next line. The minimum value is 1. The maximum value is 99. The value should be less than the value in the Words With At Least <x> Letters box.

Hyphenate Capitalized Words

Hyphenates words that begin with a capital (uppercase) letter, such as the first word of a sentence.

Hyphenate Words in ALL CAPS

Hyphenates words that have all capital letters, such as acronyms.

Allow Hyphenation in Text and Field Captions

Allows hyphenation in text objects, and in the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields.

Allow Hyphenation in Text Field Values

Allows hyphenation in the value area of text field objects (default text and text entered by the form filler).

Hyphenate New Items

Hyphenates the text in new objects that are added to the form. Hyphenates the text in text objects and in the caption area of objects such as text fields, decimal fields, numeric fields, and signature fields.

RELATED LINKS:

Hyphenate text

Considerations for setting hyphenation

Paragraph properties in the Paragraph palette

Formatting (Form Properties dialog box)

Default Fonts (Options dialog box)

Use this dialog to inspect and modify the default font properties for new forms.

To display this dialog box, click Tools > Options and select Default Fonts from the list on the left.

Default Caption Font Properties for New Forms

Sets the default font properties for captions in new forms.

Typeface

Sets the font type.

Size

Sets the size of the font.

Style

Sets the style of the font.

Default Value Font Properties for New Forms

Sets the default font properties for values in new forms.

The font properties for values are the same as the default font properties for captions.

Restore Defaults

Restores the default font properties for captions and values in new forms.

RELATED LINKS:

Default Fonts (Form Properties dialog box)

Caption Formatting (Style Editor dialog box)

Wizards and Tips (Options dialog box)

Use this dialog box to modify options for showing and hiding tips in dialog boxes, warnings, and the Welcome dialog box.

To display this dialog box, select Tools > Options and select Wizards and Tips from the list on the left.

Reset All Tips

Displays tip dialog boxes. These dialog boxes provide hints based on your actions in Designer. If you have selected Don't Show This Tip Again, but you want to display the tip dialog boxes again, click Reset All Tip Dialogs.

Reset All Warnings

Displays warning dialog boxes when an error occurs.

Show Welcome Dialog Box

Displays the Welcome dialog box when Designer starts.

Show In-place Warning Markers

Displays warning markers, such as the yellow triangle or white "X" in the red circle, that appear when there is a problem with an object.

When Value Area Is Too Small

Displays a warning marker when the value area is too small to display the longest value allowed by the field. For example, if you size the field to 3 inches wide by 0.5 inches high and you limit the length of the field to a maximum of 50 characters, then at run time, the value area may be too small to display all the characters in the field. You must adjust the width of the value area to accommodate the field value.

Designer checks the field to evaluate whether, at run time, the value area will be large enough to display the value. Designer shows the warning marker when these conditions are met:

- The field uses a fixed-width or monospace font (such as Courier or LetterGothic). If you use a proportional font (such as Times New Roman), Designer cannot evaluate the field because many i characters, for example, can fit in the space of one W character. Using a fixed-width font makes the size of the value area more predictable for Designer.

NOTE: Ensure that Designer recognizes the font as fixed-width or monospace by testing the font with a field that you set to be too small in order to see whether the warning marker appears.

- The field has a fixed dimension set in the Width and Height boxes of the Layout palette.
- The field has a length limit set to a number of characters or digits. The character limit may be set in the Field tab of the Object palette.
- The field does not have a data pattern or display pattern set for it.

NOTE: If the field is a cell within a table, the warning marker will not display.

Show Text Overflow Indicators

Displays a red text overflow indicator button on an object when the value or caption area is too small to display the contents. When you hover the pointer over the text overflow indicator button, the area that is too small is highlighted in red. You can click the button to automatically expand the affected area. Value areas expand toward the right, up to the edge of a container (subform or page), and then expand downward. Caption areas expand downward. Select this option when importing a form into Designer from another application to make the process of correctly sizing objects containing text a little easier. Notice that when the Expand to Fit options in the Layout palette are selected, text overflow indicators are not displayed.

Show in Tables

Displays a gray marker in a table when a text object cannot be expanded.

Show When an Object Cannot be Expanded

Displays a gray marker when an object cannot be expanded.

Show When Creating New Documents

Shows the New Form Assistant when creating a new form design.

The New Form Assistant is always shown when creating new documents in Designer with Workbench, and deselecting this option has no effect.

Show When Importing Documents

Shows the New Form Assistant when importing a file.

The New Form Assistant is always shown when importing documents in Designer with Workbench, and deselecting this option has no effect.

Show When Importing PDF Documents From Acrobat

Shows the New Form Assistant when opening a PDF file from Acrobat. This option does not apply when you open a PDF file directly in Designer.

Show When Creating Tables

Shows the Insert Table dialog box when creating a table using the Table object from the Object Library palette. From this dialog box you can use the Table Assistant.

Bidirectional Text (Options dialog box)

Use this dialog box to specify logical or visual cursor movement during design time. The default setting is Logical.

To display this dialog box, select Tools > Options and select Bidirectional Text from the list on the left.

Logical

Moves the insertion point in bidirectional text according to the direction of the text encountered. For example, when using the arrow keys to move through Arabic text and then English numbers in the same caption, the right arrow key moves from right-to-left through the Arabic text and then starts at the left-most character in the English number and progresses from left-to-right. The Logical option applies to dialogs boxes, palettes, Script Editor, and XML Source.

Visual

Moves the insertion point in bidirectional text by moving to the next visually adjacent character. For example, when using the arrow keys to move from right-to-left through Arabic text and then English numbers in the same caption, the left arrow key moves from right-to-left, regardless of the direction of the text. The Visual option only changes the cursor movement in the Design View and Master Pages. It does not apply it to dialogs boxes, palettes, Script Editor, and XML Source.

RELATED LINKS:

To set up an object for Arabic and Hebrew

To set up an object for Thai and Vietnamese

Data Binding (Options dialog box)

Use this dialog box to turn active labels on and off, set the default color for active labels, and set the default binding for new subforms.

To display this dialog box, select Tools > Options and select Data Binding from the list on the left.

Show Dynamic Properties

Turns the active labels for dynamic properties on or off. Active labels indicate which properties can be dynamic. They are identified by a green underline that you can click and dynamically bind the property to a data source.

Use Custom Label Color

Sets the default color for active labels.

Custom Label Color

Opens the color palette in which you select the color for active labels.

Default Binding for New Subforms

Sets the default binding for new subforms.

Use Name

Sets data merging and saving options. Data values are merged and stored implicitly according to Adobe data-merging rules.

No Data Binding

Clears data merging and saving options for new subforms. A new subform will not be involved in any data-merging operations by default.

Warnings

Warns about missing information.

Warn About Missing Data Connections

Displays a warning when you open a form and Designer cannot locate the sample data file that is specified in the Connection Properties dialog box. Warning is shown for all data connection types, except OLEDB. See *Connection Properties dialog box*.

Warn About Missing Preview Data

Displays a warning when you click the Preview PDF tab and Designer cannot locate the data file that is specified in Data File box on the Preview tab in the Form Properties dialog box. (See *Form Properties dialog box*).

Automatic Data Connection Updates

Warns when the source file for a data connection has changed

Show Update Dialog when Source File Changes

Displays a warning dialog box when the file that is the source of a data connection has changed and enables you to update the connection.

RELATED LINKS:

To set up dynamic properties

International (Options dialog box)

Use this dialog box to select the application language for the application. You must restart Designer to apply the language change.

To display this dialog box, select Tools > Options and select International from the list on the left.

Application Language

Lists the available applications languages.

Select Font

Specifies whether Designer uses the default font or one that supports extended characters (for example, GB18030 and HKSCS). The options available depend on which fonts are installed.

NOTE: HKSCS - 2001 support for Microsoft Windows 2000 and Windows XP. Hong Kong Supplementary Character Set (HKSCS) rendering is supported by Designer. If the correct HKSCS font is selected for the object, HKSCS characters should be displayed correctly in the form design as well as in the Design view and Preview PDF tab. However, HKSCS-extended characters may not be displayed correctly in the user interface, such as in the Hierarchy view and Object palette.

Display units of measurement in default application locale

Displays the units of measurement according to the application locale.

Tab Order (Options dialog box)

Use this dialog box to exclude the text and image objects from the tabbing order and to view tabbing order using visual aids.

To display this dialog box, select Tools > Options and select Tab Order from the list on the left.

Only Show Tab Order for Fields

Displays tabbing order only for the input or display fields. Text fields and images are excluded from the tabbing order.

Display Additional Visual Aids for Tab Order

When the pointer hovers over a field, displays blue arrows indicating the tabbing order for the two preceding and the two following fields.

RELATED LINKS:

Excluding text and image objects from the tabbing order

To view the tabbing order using visual aids

Security (Options dialog box)

Use this dialog box to specify whether the login dialog box is always displayed when users log in and to access the Digital IDs dialog box.

To display this dialog box, select Tools > Options and select Security from the list on the left.

Digital IDs

Opens the Digital IDs dialog box.

RELATED LINKS:

Digital IDs dialog box

Importing digital IDs

Patterns dialog box

Display (Patterns dialog box)

Use this dialog box to define the pattern for displaying formatted values (date, time, numerical, and textual data) in a form. The display pattern must be compatible with the option selected in the Data Format list on the Data panel. For example, if the data format is set to Date, the displayed pattern must represent a date value.

NOTE: At design and run time, formatted values in the object display in the locale-sensitive format.

To display this dialog box, select the object you want to create a pattern for and, in the Object palette, click the Field tab, click Patterns, and click the Display tab.

Select Type

Specifies the pattern used to display data. Options for dates, times, dates/times, and numeric data vary, depending on the option selected in the Locale list. To select multiple display patterns, hold down the CTRL or SHIFT key while selecting options.

NOTE: To display all date, time, and date/time formats in the Select Type list, you must first select Date and Time in the Data Format list of the Binding tab in the Object palette.

Locale

Specifies the language and country, or region used to define the format and punctuation of date, time, date/time, numeric, and currency values displayed in the selected object. Locale options are organized first by language and then by country or region. The locale option that is selected by default, matches the option that is selected in the Form Locale list on the Defaults panel in the Form Properties dialog box. You can select a specific language, country, or region from the list, or you can select one of these options:

Default Locale

Uses the Default Locale specified on the Defaults panel in the Form Properties dialog box.

Viewer's System Locale

Uses the system locale of the user's computer.

Notice that the options available in the Select Type list can differ depending on the locale selected.

Sample

Specifies the value of the formatted object against the selected pattern. If no value is specified, the Sample box remains blank or displays the value produced by the Allow Empty option if a value is included in the pattern.

Pattern

Displays the pattern associated with the option selected in the Select Type list, as well as the Allow Empty box and Allow Zero box. You can also type custom patterns in this box. You cannot mix patterns from the Select Type list with custom patterns.

Allow Empty

Inserts a null value symbol into the pattern. Lets you define how the object processes an empty value whether it is a pattern clause or text.

Allow Zero

Inserts a zero symbol into the pattern and defines how an object processes a zero value.

RELATED LINKS:

Edit (Patterns dialog box)

Validation (Patterns dialog box)

Data (Patterns dialog box)

Edit (Patterns dialog box)

Use this dialog box to define the pattern for the syntax of user input (date, time, numerical, and textual data). The edit pattern must be compatible with the option selected in the Data Format list on the Data tab. For example, if the data format is set to Date, the pattern must represent a date value. If the user's input does not conform to the Edit Pattern, the data is rejected and the field value is not formatted.

NOTE: At design and run time, formatted values in the object appear in the locale-sensitive format.

To display this dialog box, select the object you want to create a pattern for and, in the Object palette, click the Field tab, click Patterns, and click the Edit tab. The options on this tab are the same as the options on the Display tab. For information about the options available, see *Display (Patterns dialog box)*.

NOTE: This option is not available when the Type option in the Value tab of the Object palette is set to Protected, Calculated - Read Only or Read Only.

RELATED LINKS:

Validation (Patterns dialog box)

Data (Patterns dialog box)

Validation (Patterns dialog box)

Use this dialog box to define a validation pattern for validating user input (date, time, numerical, and textual data). The pattern must match the syntax of the user input. Only this tab is available with barcodes, drop-down list, and list box; the Display, Edit, and Data tabs are not available.

To display this dialog box, select the object you want to create a pattern for and, in the Object palette, click the Value tab, click Validation Patterns, and click the Validation tab. The options on this tab are the same as the options on the Display tab. For information about the options available, see *Display (Patterns dialog box)*.

NOTE: This option is not available when the Type option is set to Protected, Calculated - Read Only or Read Only.

RELATED LINKS:

Edit (Patterns dialog box)

Data (Patterns dialog box)

Form Validation (Form Properties dialog box)

Data (Patterns dialog box)

Use this dialog box to define a pattern for storing and retrieving bound data or for saving data when the form is not bound to a data source. See *To specify a data pattern*.

To display this dialog box, select the object you want to create a pattern for and, in the Object palette, click the Field tab, click Patterns, and click the Data tab. The options on this tab are the same as the options on the Display tab. For information about the options available, see *Display (Patterns dialog box)*.

RELATED LINKS:

Edit (Patterns dialog box)

Print dialog box

Prints the selected form design on the specified printer. If the option Print Form with Data is not selected, Designer prints the Design View or the Master Pages view, which ever is currently open.

The options that display in the Printer and Print range sections of the dialog box depend on the printer that your computer is connected to. For more information about your printer's options, see the printer documentation.

Print Form with Data

Selecting this option allows you to print a final version of the current form design to the selected printer. You can optionally choose to print the form design with a specified data file. Designer merges the form design with the data file to produce a complete document on the selected printer.

Data File

Allows you to specify the data file to use when printing a form design with data. The data file must be in standard XML format. You can type the path to the data file in the Data File box or browse to the file.

Generate Data File

Select this option to generate the sample form data to use when printing a form design. When you print the form design, it will be merged with the sample data. You can also indicate how many times to repeat a repeating subform in the printed form design.

Two-Sided Print

Select this option to print forms that will be printed on both sides of the paper. If the printer does not support two-sided printing, the Single-Sided Print options appear instead.

Configuration File

Allows you to specify the device configuration file that will define the format of the output. The drop-down list displays several configuration files installed with Designer, as well as any that you add to the Designer installed folder. The list shows the name of the XDC file in square brackets, preceded by a more descriptive name (for example, PCL - Black & White [hppcl5e].

Publish To Repository dialog box

This dialog box is available only in the stand-alone version of Designer. Use this dialog box to put a form design and its linked files in a shared or web folder where it can be accessed by many users or by other applications. If the form contains a link to an external file when the form design and files are published, the list in the form design is modified to reflect the new location of the file.

To display this dialog box, click File > Publish To Repository.

Save in

Specifies the location where you want to publish the form.

File name

Displays the file name.

Save as type

Lists the file types that you can publish.

Adobe Static PDF Form (*.pdf)

Saves forms as static PDF, based on the Acrobat and Adobe Reader target version specified. Static PDF forms render once and are displayed on the client in the Acrobat or Adobe Reader target version. They are not rerendered in response to user interaction. The PDF form may have been designed with a flowable layout, but when the static PDF form is created, its layout is fixed and the resulting PDF form will not rerender on the client. Static PDF forms can be interactive or non-interactive.

Adobe Dynamic XML Form (*.pdf)

Saves forms as dynamic PDF, based on the Acrobat and Adobe Reader target version. The form design can contain dynamic elements. Dynamic PDF forms render on the client in Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of an object are possible in Adobe Reader because Adobe Reader has enough information to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If the end user clicks a button that adds a new row to a table, the form is rerendered in Adobe Reader.

Adobe XML Form (*.xdp)

Publishes forms using the native XML-based file format created by Designer. Use this option if you will be using Forms.

Designer Template (*.tds)

Publishes the form as a template (.tds) file. It can contain components and settings, such as fonts, page layout, formatting, and scripts. Use it as a starting point for a new form.

RELATED LINKS:

To publish a form to a repository

Remove Style dialog box

Use this dialog box to remove styles from objects on the Design View tab.

To display this dialog box, on the Design View tab, select the object you want to remove a style from, right-click, and select Styles > Remove Style.

Keep current style properties after removing the reference

Removes reference to style sheet. Embeds object's style properties in form. Style properties remain unchanged. Appearance of object does not change.

Reset object properties to match the standard objects in the Object library

Removes reference to style sheet. Restores object's style properties to default settings. Appearance of object reverts to standard look.

Do not reset object properties

Does not remove style properties. Appearance of object changes to XML Forms Architecture (XFA) settings.

RELATED LINKS:

To remove a style from an object

Styles

Report Warnings Options dialog box

Use this dialog box to specify the warnings to display on the Warnings tab in the Report palette.

To display this dialog box, click the Report palette Warnings tab menu button > Warning Options.

Target Version

Shows target version warnings in the Warnings tab.

Format Pattern

Shows picture clause warnings in the Warnings tab.

Data Binding

Shows data binding warnings in the Warnings tab.

Locale Setting

Shows locale setting warnings in the Warnings tab.

Script Syntax Checking

Shows script syntax checking warnings in the Warnings tab.

Miscellaneous

Shows miscellaneous warnings in the Warnings tab.

Actions

Shows warnings related to actions in the Warnings tab.

RELATED LINKS:

Addressing warning messages in the Report palette

Report palette

Save As dialog box

Use this dialog box to save a form designs or template using its current name and location, or save a copy using a different name or location.

To display this dialog box, select File > Save As.

Save in

Specifies the location where you want to locate a file or folder.

File name

Displays the file name. The name is created when you save the form design.

Save as type

Lists the available file types that you can save.

Adobe Static PDF Form (*.pdf)

Saves forms as static PDF, based on the Acrobat and Adobe Reader target version specified. Static PDF forms render once and are displayed on the client in the Acrobat or Adobe Reader target version. They are not rerendered in response to user interaction. The PDF form may have been designed with a flowable layout, but when the static PDF form is created, its layout is fixed and the resulting PDF form will not rerender on the client. Static PDF forms can be interactive or non-interactive.

Adobe Dynamic XML Form (*.pdf)

Saves forms as dynamic PDF, based on the Acrobat and Adobe Reader target version. The form design can contain dynamic elements. Dynamic PDF forms render on the client in Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of an object are possible in Adobe Reader because Adobe Reader has enough information to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If an end user clicks a button that adds a new row to a table, the form is rerendered in Adobe Reader.

Adobe XML Form (*.xdp)

Saves forms in the native XML-based file format created by Designer. Use this option if you will be using Forms.

Designer Template (*.tds

Saves the basic structure for a form as a template. It can contain components and settings, such as fonts, page layout, formatting, and scripts. Use it as a starting point for a new form.

RELATED LINKS:

To save a form design

To save a copy of a form design

Select Digital IDs dialog box

Use this dialog box to select the digital ID used to access the Web Service Description Language (WSDL) document that you are using to create a data connection, or to add or remove digital IDs to and from the Windows Certificate Store and the Designer Digital IDs File store.

To display this dialog box, select File > New Data Connection, and create a WSDL data connection.

Digital IDs

Lists the digital IDs that are available in the Windows Certificate Store and the Designer Digital ID Files store.

Add Digital ID

Opens the Add Digital ID dialog box, where you specify the location of the digital file (*.p12 or *.pfx) you want to add to the list, and enter the password required to access the file. To remove a digital file from the list, use the Digital IDs dialog box (Tools > Options > Security > Digital IDs). See *Digital IDs dialog box*.

Refresh

Updates the Digital IDs list to display the IDs that are currently available.

View

Opens the Certificate Viewer dialog box, which displays general and detailed information about the selected certificate (digital ID).

RELATED LINKS:

Security (Options dialog box)

Importing digital IDs

To create a data connection using a WSDL file

Signature Settings dialog box

Signature Handler (Signature Settings/Sign Data and Submit Settings dialog box)

Use this dialog box to define an optional signature handler for a document signature or a data signature. A signature handler is a software module that implements the public-key cryptography process and controls access to the encrypted document or data. The Adobe.PPKLite (Adobe default security) signature handler supports public-key encryption.

NOTE: Adobe.PPKLite is the only signature handler that is designed to analyze and process the information that you enter in the Signature Settings and Sign Data and Submit Settings dialog boxes. Third-party signature handlers may not process this information.

To display this dialog box, perform one of the following actions:

- Select a signature field, click the Signature tab in the Object palette, click Settings, and then click Signature Handler.
- Select a submit button, click the Submit tab in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Signature Handler.
- Select an email or HTTP submit button and, in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Signature Handler.

Name

Lists Adobe.PPKLite as the default security handler. You can enter the name of another third-party signature handler if needed.

Require the signing party to use the specified signature handler

Forces Acrobat to use the signature handler (also known as a filter) that is specified in the Name list box to create the digital signature. If the signing party does not have the specified signature handler available on their computer, Acrobat will cancel the signing operation.

When deciding whether to require the signing party to use the specified signature handler (which they may or may not have available), it is helpful to understand what the signing party can and cannot do in Acrobat and Adobe Reader, depending on the circumstances.

The following table describes the various situations that can arise when you specify a signature handler in the Name list.

Required	Available to signing party	Behavior in Acrobat or Adobe Reader No
No	No	Acrobat or Adobe Reader attempts to use the default installed signing method, but allows the signing party to select any other installed signature handler.

Required	Available to signing party	Behavior in Acrobat or Adobe Reader No
No	Yes	Acrobat or Adobe Reader lets the signing party use only the signature handler specified in the Name list. The signing party cannot use any other signature handler.
Yes	No	Acrobat or Adobe Reader requires the signing party to use the signature handler specified in the Name list. The signing party cannot sign until they install the specified signature handler.
Yes	Yes	Acrobat or Adobe Reader requires the signing party to use the signature handler specified in the Name list.

Specify the subFilter

Specifies the syntax of the contents of the signature dictionary and allows inter-operability between handlers. The valid subFiltersfor digital signatures that Acrobat recognizes are adbe.x509.rsa_sha1,adbe.pkcs7.detached, and adbe.pkcs7.sha1.

Require the signing party to use the specified subFilter

Forces Acrobat or Adobe Reader to use one of the specified subFilters.The first name in the subFilter list that matches an encoding method supported by the signature handler is used as the encoding method for signing. If you specify a subFilter, and the flags property indicates that this entry is a required constraint, the first matching encoding is used when signing; otherwise, signing will not take place. If the flags property indicates that this is an optional constraint, the first matching encoding is used if it is available. If it is not available, the system default encoding is used.

When deciding whether to require the signing party to use the specifiedsubFilter (which they may or may not have available), it is helpful to understand what the signing party can and cannot do in Acrobat and Adobe Reader, depending on the circumstances.

The following table describes the various situations that can arise with subFilters.

NOTE: The Adobe.PPKLite signature handler analyzes and processes the subFilter information that you enter in the Signature Settings and the SignData and Submit Settings dialog boxes, not Acrobat. As a result, these four situations arise only if you select the Adobe.PPKLitesignature handler option. Third-party signature handlers may not process this information.

Required	Available to signing party	Behavior in Acrobat or AdobeReader
No	No	The Adobe.PPKLite signature handler requires the signing party to use one of the subFilters specified in the subFilter list. If you specify but do not require a subFilter, the signing party can use another subFilter. If you specify a signature handler other than Adobe.PPKLite it may or may not respect this setting.
No	Yes	The Adobe.PPKLite signature handler lets the signing party use only one of the subFilters specified in the subFilter list. The signing party cannot use any other subFilter. If you specify a signature handler other than Adobe.PPKLite it may or may not respect this setting.
Yes	No	The Adobe.PPKLite signature handler requires the signing party to use one of the subFilters specified in the subFilter list. The signing party cannot sign until they select one of the specified subFilters. If you specify a signature handler other than Adobe.PPKLite it may or may not respect this setting.
Yes	Yes	The Adobe.PPKLite signature handler requires the signing party to use one of the subFilters specified in the subFilter list. If you specify a signature handler other than Adobe.PPKLite it may or may not respect this setting.

Specify the Digest Method

Specify the acceptable digest method element values. The valid string values are MD5, SHA1, SHA256, SHA384, SHA512, and RIPEMD160.

Require the signing party to use the specified digest method

Forces Acrobat or Adobe Reader to use the specified digest method.

When deciding whether to require the signing party to use the digest method (which they may or may not have available), it is helpful to understand what the signing party can and cannot do in Acrobat and Adobe Reader, depending on the circumstances.

The following table describes the various situations that can arise with digest methods.

NOTE: The Adobe.PPKLite signature handler analyzes and processes the digest method information that you enter in the Signature Settings and the Sign Data and Submit Settings dialog boxes, not Acrobat. As a result, these four situations arise only if you select the Adobe.PPKLite signature handler. Third-party signature handlers may not process this information.

Required	Available to signing party	Behavior in Acrobat or AdobeReader
No	No	The Adobe.PPKLite signature handler uses the system default digest method.If you specify a signature handler other than the Adobe.PPKLitesignature handler, it may or may not respect this setting.
No	Yes	The Adobe.PPKLite signature handler requires the signing party to use one of the digest methods specified in the Digest Method list. The signing party cannot use another digest method. If you specify a signature handler other than the Adobe.PPKLitesignature handler, it may or may not respect this setting.
Yes	No	TheAdobe.PPKLite signature handler requires the signing partyto use one of the digest methods specified in the Digest Method list. If you specify a signature handler other than theAdobe.PPKLite signaturehandler it may or may not respect this setting.
Yes	Yes	The Adobe.PPKLite signature handler requires the signing party to use one of the digest methods specified in the Digest Method list. If you specify a signature handler other than the Adobe.PPKLitesignature handler it may or may not respect this setting.

RELATED LINKS:

Using signature fields

Email submit buttons

Overview of HTTP submit buttons

About submitting data using a button

To add a signature field

Signing Certificates (Signature Settings/Sign Data and Submit Settings dialog box)

Use this dialog box to specify certificates that identify signing parties who can sign the document or data.

To display this dialog box, perform one of the following actions:

- Select a signature field, click the Signature tab in the Object palette, click Settings, and then click Signing Certificates.
- Select a submit button, click the Submit tab in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Signing Certificates.
- Select an email or HTTP submit button and, in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Signing Certificates.

Add

Opens the Select User Certificate dialog box, where you can browse for and add signing certificates. A certificate file has a file name extension such as .p7C, .p7b, or .cer.

Remove

Removes the selected certificates from the list.

Details

Opens the Certificate Viewer, where you can view the details about the selected certificate. The details that are listed vary according to the installed Certificate Authority (CA) certificates.

Restrict signing to parties associated with the specified certificates

Allows only the signing parties that are identified by the listed certificates to sign the document or data. However, if the selected certificate is available, the signer must use it even you do not select this option

Specify the Subject Distinguished Name

Specify a subject Distinguished Name (DN) for the signing certificate. A subject distinguished name in a signing certificate has several attributes. An example of a distinguished name is O = My Company; OU = My Department; CN = My Name; and C = My Country. The attributes in this distinguished name are O for Organization; OU for Organizational Unit; CN for Common Name; C for Country.

Require the signing certificate to meet the specified subject distinguished name

Forces the signing certificate to meet the specified subject distinguished name.

When deciding whether to require the signing certificate to meet the specified subject distinguished name, it is helpful to understand what the signing party can and cannot do in Acrobat and Adobe Reader, depending on the circumstances.

The following table describes the various situations that can arise with signing certificates that may or not meet the specified subject distinguished name.

NOTE: The Adobe.PPKLite signature handler analyzes and processes the signing certificate information that you enter in the Signature Settings and the Sign Data and Submit Settings dialog boxes, not Acrobat. As a result, these four situations arise only if you select the Adobe.PPKLite signature handler. Third-party signature handlers may not process this information.

Required	Available to signing party	Behavior in Acrobat or Adobe Reader
No	No	The Adobe.PPKLite signature handler requires the signing party to use a digital ID with a subject distinguished name that matches one of the subject distinguished names in the Subject Distinguished Name list. If you do not specify a subject distinguished name, the signing party can use another digital ID. If you specify a signature handler other than the Adobe.PPKLite signature handler it may or may not respect this setting.
No	Yes	The Adobe.PPKLite signature handler requires the signing party to use only a digital ID with a subject distinguished name that matches one of the subject distinguished names in the Subject Distinguished Name list. The signing party cannot use another digital ID. If you specify a signature handler other than theAdobe.PPKLite signature handler it may or may not respect this setting.

Required	Available to signing party	Behavior in Acrobat or Adobe Reader
Yes	No	The Adobe.PPKLite signature handler requires the signing party to use a digital ID with a subject distinguished name that matches one of the subject distinguished names in the Subject Distinguished Name list. The signing party must obtain the required digital ID before they sign. If you specify a signature handler other than the Adobe.PPKLite signature handler it may or may not respect this setting.
Yes	Yes	The Adobe.PPKLite signature handler requires the signing party to use a digital ID with a subject distinguished name that matches one of the subject distinguished names in the Subject Distinguished Name list. If you specify a signature handler other than the Adobe.PPKLite signature handler it may or may not respect this setting.

RELATED LINKS:

Using signature fields

Email submit buttons

Overview of HTTP submit buttons

About submitting data using a button

To add a signature field

To add a data signature to a submit button

Issuers and Policies (Signature Settings/Sign Data and Submit Settings dialog box)

Use this dialog box to specify certificate issuers, a certificate enrollment server, and certificate policies.

To display this dialog box, perform one of the following actions:

- Select a signature field, click the Signature tab in the Object palette, click Settings, and then click Issuers and Policies.
- Select a submit button, click the Submit tab in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Issuers and Policies.
- Select an email or HTTP submit button and, in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Issuers and Policies.

Certificate Issuers

Lists the certificates that identify the issuers who can provide signing parties with signing certificates:

Add

Opens the Select Issuer Certificate dialog box, where you can browse for and add issuer certificates. A certificate file has a file name extension such as .p7c, .p7b, or .cer.

Remove

Removes the selected certificates from the list.

Details

Opens the Certificate Viewer, where you can view the details about the selected certificate. The details that are listed vary according to the installed Certificate Authority (CA) certificates.

Restrict signing to certificates from the specified issuers

Acrobat rejects the signature if the signing party does not have a certificate from a specified issuer.

Certificate Enrollment Server URL

Specifies the URL for a web page where people can enroll for a certificate. This web page appears when a signing party does not have an available signing certificate.

Certificate Policies And Associated Object Identifiers

Lists the object identifiers (OIDs) that are associated with the certificate policies that restrict the certificates that can be used to sign the document or data. When you set an OID, you must also specify a certificate issuer so that Acrobat recognizes the entry.

Add

Adds an object identifier.

Delete

Deletes the selected object identifier.

Restrict signing to certificates that conform to the specified policies

Acrobat rejects the signature if the signing certificate does not conform to the specified policies.

When deciding whether to restrict signing to certificates that conform to the specified policies, it is helpful to understand what the signing party can and cannot do in Acrobat and Adobe Reader, depending on the circumstances.

The following table describes the various situations that can arise with signing certificates that may or may not conform to the specified policies.

NOTE: The Adobe.PPKLite signature handler analyzes and processes the object identifier information that you enter in the Signature Settings and the Sign Data and Submit Settings dialog boxes, not Acrobat. As

a result, these four situations arise only if you select the Adobe.PPKLite signature handler. Third-party signature handlers may not process this information.

Required	Available to signing party	Behavior in Acrobat or Adobe Reader
No	No	The Adobe.PPKLite signature handler lets the signing party use any digital ID for signing regardless of the certificate policy.
No	Yes	The Adobe.PPKLite signature handler only lets the signing party use a digital ID with the specified certificate policy. The signing party cannot select a digital ID for signing that does not contain the matching certificate policy.
Yes	No	The Adobe.PPKLite signature handler requires the signing party to use a digital ID with the specified certificate policy. The signing party must obtain a digital ID with the specified certificate policy before they can proceed with the signing.
Yes	Yes	The Adobe.PPKLite signature handler requires the signing party to use a digital ID with the specified certificate policy.

RELATED LINKS:

Using signature fields

Email submit buttons

Overview of HTTP submit buttons

About submitting data using a button

To add a signature field

To add a data signature to a submit button

Signing Reasons (Signature Settings/Sign Data and Submit Settings dialog box)

Use this dialog box to specify reasons for the document signature.

To display this dialog box, select a signature field, click the Signature tab in the Object palette, click Settings, and then click Signing Reasons.

Specify The Signing Reasons

The specified reasons replace the default reasons in Acrobat. The signing party can select one of the reasons when signing a form.

Add

Adds a signing reason to the list.

Delete

Removes the selected signing reasons from the list.

Move Up

Moves the selected reason up in the list.

Move Down

Moves the selected reason down in the list.

Sort Ascending

Sorts the list in alphabetical order, from top to bottom.

Sort Descending

Sorts the list in alphabetical order, from bottom to top.

Require the signing party to select a reason from this list

Forces the signing party to select a reason from the list when signing the form. If the Signing Reason list is not required, the signing party can use any other signing reasons that may be displayed in the Certify Document or Sign Document dialog box, or they can enter another reason. If the Signing Reason list is required, the signing party must select one of the supplied signing reasons; they cannot add their own reason. If the Signing Reason list is required but no signing reasons are added to the list, Acrobat 8.0 and later disables the Reasons list.

RELATED LINKS:

Using signature fields

To add a signature field

Key Usage (Signature Settings/Sign Data and Submit Settings dialog box)

Use this panel to specify which key usage parameters an encryption certificate requires or excludes to be acceptable for XML encryption.

To display this dialog box, perform one of the following actions:

- Select a Button object, on the Field tab select Submit, click the Submit tab, select Encrypt Submission, click Settings, and then click Key Usage.
- Select an Email Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Key Usage.

- Select an HTTP Submit Button object, on the Field tab select Encrypt Submission, click Settings, and then click Key Usage.

  The key usage settings represent the key usage parameters of a certificate. Each key usage check box enables or disables the associated drop-down list. In every drop-down list, you can select either Require Key Usage or Exclude Key Usage.

Require Key Usage

Indicates that the encryption certificate requires the specified key usage parameter.

Exclude Key Usage

Indicates that the encryption certificate excludes the specified key usage parameter.

Require The Signing Certificate To Satisfy The Key Usage Settings

Allows the author to specify that the content can be encrypted only for public encryption certificates that meet the criteria indicated in each key usage setting list.

RELATED LINKS:

Using buttons

About submitting data using a button

Signature Info (Signature Settings/Sign Data and Submit Settings dialog box)

Use this dialog box to specify a Certificate Revocation List (CRL) or an Online Certificate Status Protocol (OCSP) response and to specify a time-stamp server.

To display this dialog box, perform one of the following actions:

- Select a signature field, click the Signature tab in the Object palette, click Settings, and then click Signature Info.
- Select a submit button, click the Submit tab in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Signature Info.
- Select an email or HTTP submit button and, in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Signature Info.

Time Stamp Server URL

This option appears only for signature fields. Specifies the URL for the time-stamp server. Time-stamps specify the date and time when a document or data was signed. Using an external server to append a timestamp to a digital signature removes any doubt about when the document was signed. Acrobat or Adobe Reader selects a time-stamp server in the following order of priority:

1) Uses the time-stamp server that is set in the Security Settings console
2) Uses the time-stamp server that is listed as an extension in the digital ID
3) Uses the specified time-stamp server first

Require the signing party to use the specified time stamp server

This option appears only for signature fields. Forces Acrobat to append a timestamp to a digital signature when the document is signed. If you select this option, Acrobat or Adobe Reader selects a time-stamp server; however, if a valid timestamp is not returned, the signing will fail and the signature will not be created. If you do not select this option, Acrobat or Adobe Reader selects a time-stamp server; however, if a valid time-stamp server is not returned, the signing will still continue.

Include the CRL or OCSP response when signing

Includes the CRL or OCSP response in the signature when the document or data is signed. The signing party must have access to the Internet to retrieve the CRL or OCSP response from the appropriate server.

Require in order for signing to succeed

Requires the CRL or OCSP response for signing to succeed.

Signing Server URL

Points to a web service that holds the digital credentials that a signing party uses to sign a document or data.

Restrict signing to specified URL server

Forces Acrobat to search for the credentials on the specified server. If this option is not selected, the server is only suggested; Acrobat can use other available credentials if the specified server is not available. If the server is available, it must use credentials even if they are not required. If you do not require a specific roaming credential server, Acrobat uses any digital ID regardless of its location.

RELATED LINKS:

Using signature fields

Email submit buttons

Overview of HTTP submit buttons

About submitting data using a button

To add a signature field

To add a data signature to a submit button

Sign Data and Submit Settings dialog box

About the Sign Data and Submit Settings dialog box

This dialog box contains the same panels as the Signature Settings dialog box, with the exception that it contains the Sign Data panel instead of the Signing Reasons panel.

For information about the panels in this dialog box, see *Signature Settings dialog box*.

Sign Data (Sign Data and Submit Settings dialog box)

Use this dialog box to select the data that is included in a data signature associated with a submit button. The signature can include either the default data set or all the data.

To display this dialog box, perform one of the following actions:

- Select a submit button, click the Submit tab in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Sign Data.
- Select an email or HTTP submit button and, in the Object palette, ensure that Sign Submission is selected, click Settings, and then click Sign Data.

Sign Data Only

Creates a signature that covers only the form data in the submission. This option is the default setting.

Sign Entire Submission

Creates a signature that covers the form data and the included attachments, such as the form design, document signature, or annotations.

RELATED LINKS:

Email submit buttons

Overview of HTTP submit buttons

About submitting data using a button

To add a data signature to a submit button

Specify Substitute Font dialog box

Use this dialog box to select a substitute font for the missing font.

To display this dialog box, select Tools > Options. Select Document Handling from the list on the left, and then click Modify Font Substitutions. Click Change Substitute Font.

Missing Font

Specifies the name of the font that is missing from your computer.

Available Fonts

Lists the fonts that you can choose as the substitute for the missing font.

Font Style

Lists the font styles for the selected available font.

Spreadsheet Paste Options dialog box

Use this dialog box to paste cells from a Microsoft Excel spreadsheet into a form, or use the spreadsheet as the basis to create a new PDF form.

To display this dialog box, select Edit > Paste. The Spreadsheet Paste Options dialog box is displayed only when you paste the content from an Excel spreadsheet into a form.

Paste cells as a table

Creates a table object in the form with the same dimensions, number of cells, and data content as the selection in the spreadsheet. Designer also preserves some of the original formatting of the spreadsheet cells, including the text, fonts, colors, background colors, alignment, and borders. If the selected cells in the spreadsheet contain column headers, Designer maintains the column header formatting in a header row at the top of the table.

Paste cells as input fields

Creates one text field object for each column in the spreadsheet. Text field objects are positioned in the form from left to right and wrapped onto additional lines if needed. The caption and binding name of each text field object corresponds to the column header text in the spreadsheet.

RELATED LINKS:

Copying spreadsheet data from Microsoft Excel

Creating a form from the content of a Microsoft Excel spreadsheet

Style Editor dialog box

General (Style Editor dialog box)

Use this dialog to edit the styles in the internal style sheet panel.

To display this dialog box, right click a style in the Internal style sheet panel, select Edit Style, and then select General from the list on the left. The General panel is displayed by default.

Style Name

Name of the style. Provide a unique name for each style within a single style sheet. The style name cannot contain spaces.

NOTE: You can override the individual formatting options for a style by using the different library palettes. When you override a formatting option, an asterisk appears next to the option and next to style name in the style toolbar.

Type

Lists the available object types you can use for the style.

The rest of the options on the General panel change depending on the object type you select. The options that are available on the General panel for a particular object type correspond to the options that are available on the Field tab in the Object palette. See *Properties in the Object palette*.

RELATED LINKS:

Styles

Editing a style

Caption Text Formatting (Style Editor dialog box)

Use this dialog to format the text in field captions. To display this dialog box, right click a style in the Internal style sheet panel, select Edit Style, and then select Caption Text Formatting from the list on the left.

Inherit Properties From

Specifies the style from which to inherit formatting properties. Lists the available styles for the same object type or the Common Style type. Do Not Inherit is the default option. When you select this option, you can still override individual formatting properties. When you save the style, only those overrides which differ from the referenced style are preserved.

Font and Font Size

Sets the typeface and typeface size.

Style

Sets the typeface style.

Baseline Shift

Sets the amount of space to move a character (or group of characters) up or down relative to the baseline. Positive numbers shift characters up, and negative numbers shift characters down.

Letter Spacing

Sets the amount of space to leave between the letters in a word or between a group of words.

Vertical Scale

Sets the degree (percentage) by which to increase or decrease the vertical size of text.

Horizontal Scale

Sets the degree (percentage) by which to increase or decrease the horizontal size of text.

Auto Kern

Reduces the amount of space between letters within a word or group of words.

Alignment

Sets horizontal and vertical alignment.

Align Left

Left-aligns the caption or value.

Align Center

Center-aligns the caption or value.

Align Right

Right-aligns the caption or value.

Justify

Justifies the caption or value. Full justification is applied to all of the lines except the last line in a multiple-line caption or value (single-line paragraphs cannot be justified).

Align Top

Aligns to the top of the area reserved for the caption and value.

Align Middle

Aligns to the middle of the area reserved for the caption and value.

Align Bottom

Aligns to the bottom of the area reserved for the caption and value.

Indents

Sets the indentation of the paragraph.

Left

Sets the left indention.

Right

Sets the right indentation.

First

Sets the first line indentation. None indents all lines by the same amount. First line indents the first line only. Hanging indents all line except the first.

By

Sets the amount of the first line or hanging indentation.

Spacing

Sets the amount of space above and below the paragraph.

Above

Sets the amount of vertical space above the paragraph.

Below

Sets the amount of vertical space below the paragraph.

Line Spacing

Sets the amount of space between lines. Single makes the line height equivalent to the height of the tallest character in the line. 1.5 Lines makes the line height one and a half times the height of

the tallest character in the line. Double Line makes the line height twice the height of the tallest character in the line.

RELATED LINKS:

Styles

Formatting captions

Value Text Formatting (Style Editor dialog box)

Use this dialog to format the text in field values. To display this dialog box, right click a style in the Internal style sheet panel, select Edit Style, and then select Value Text Formatting from the list on the left.

The options on the Value Text Formatting panel are essentially the same as the options on the Caption Text Formatting panel. For definitions of the various options, see *Caption Text Formatting (Style Editor dialog box)*.

RELATED LINKS:

Styles

Formatting field values and using patterns.

Border (Style Editor dialog box)

Use this dialog to format border properties. To display this dialog box, right click a style in the Internal style sheet panel, select Edit Style, and select Borders from the list on the left.

Inherit Properties From

Specifies the style from which to inherit formatting properties. Lists the available styles for the same object type or the Common Style type. Do Not Inherit is the default option. When you select this option, you can still override individual formatting properties. When you save the style, only those overrides which differ from the referenced style are preserved. Note that if you choose to inherit border properties, the other options on the Border panel are unavailable and individual options cannot be overridden.

Edges

Sets border properties for all or individual edges.

Edit Together

Applies the same border to all edges.

Edit Individually

Applies a border to each edge separately. This option does not apply to three-dimensional borders.

Sides

Sets the style, width, and color of the borders.

Corner

Applies a style to border corners.

Omit Border Around Page Breaks

Omits the selected object's bottom border on the first page and top border on the second page.

Style

Sets the background fill style and color.

RELATED LINKS:

Styles

Formatting objects

Layout (Style Editor dialog box)

Use this dialog to format layout properties. To display this dialog box, right click a style in the Internal style sheet panel, select Edit Style, and select Layout from the list on the left.

Size

Enables the Width, Height, and Expand to Fit options

Width and Expand to Fit

Sets the minimum overall width of the object and enables expansion in that direction if necessary.

Height and Expand to Fit

Sets the minimum overall height of the object and enables expansion in that direction if necessary.

Anchor

Sets the object's anchor point.

Rotate

Removes object rotation or rotates object by 90, 180, or 270 degrees.

Content Alignment in a Flowed Container

Sets the alignment of content in a flowed container like a subform.

Margins

Sets the amount of white space surrounding an object.

Caption

Sets the position of a caption and the amount of white space reserved between the caption and value field.

RELATED LINKS:

Styles

Formatting objects

Syntax Formatting dialog box

Use the FormCalc, JavaScript, and XML Source Syntax Formatting dialog boxes to define the font and color settings for the FormCalc syntax, JavaScript syntax, and the XML source syntax.

To display Arabic, Hebrew, Thai, and Vietnamese characters in the Script Editor or XML Source Tab, you must change the font settings that Designer uses in the Script Editor and XML Source tab. Otherwise, Designer displays rectangles where the language-specific characters should be. For example, set the font for Arabic to Adobe Arabic, for Hebrew to Adobe Hebrew, for Thai to Adobe Thai, and for Vietnamese to Myriad Pro or Minion Pro.

To display this dialog box, select Tools > Options. Select Workspace from the list on the left and then click one of the Syntax Formatting buttons.

Font

Sets the font.

Size

Sets the font size.

Color Settings

Sets the category that you want to modify.

Use Custom Text Color

Sets the color of the characters.

Use Custom Background Color

Sets the color of the background.

Sample

Displays an example of what you set.

Restore Defaults

Resets the font and color settings to their defaults.

Table Assistant dialog box

Body Layout (Table Assistant dialog box)

Use this dialog box to create a table that has a fixed number of body rows and columns or to create a table that has a varying number of body rows and a fixed number of columns.

To display this dialog box, select Insert > Standard > Table > Create Table Using Assistant > OK.

Body Has Fixed Dimensions

Creates a table that has the selected (fixed) number of columns and body rows.

Number of Columns

Adds the specified number of columns (20 maximum).

Number of Body Rows

Adds the specified number of body rows (50 maximum).

Body Rows Vary Depending On Data

Creates a table in which the number of rows varies depending on the number of rows in the data.

RELATED LINKS:

To create a table using the Table Assistant

Footer Row (Table Assistant dialog box)

Use this dialog box to exclude a footer row from the table, to include a footer row in the table, and to make the footer row repeat on every page.

To display this dialog box, select Insert > Standard > Table > Create Table Using Assistant > OK > Footer Row.

No Footer Row

Excludes a footer row from the table.

Has Footer Row

Includes a footer row in the table.

Repeat Footer Row On Each Page

Repeats a footer row on each page. This option is available only when the parent subform is set to Flowed in the Subform tab in the Object Palette.

RELATED LINKS:

To create a table using the Table Assistant

Header Row (Table Assistant dialog box)

Use this dialog box to exclude a header row from the table, to include a header row in the table, and to make the header row repeat on every page.

To display this dialog box, select Insert > Standard > Table > Create Table Using Assistant > OK > Header Row.

No Header Row

Excludes a header row from the table.

Has Header Row

Includes a header row in the table.

Repeat Header Row On Each Page

Repeats a header row on each page. This option is available only when the parent subform is set to Flowed in the Subform tab in the Object Palette.

RELATED LINKS:

To create a table using the Table Assistant

Row Shading (Table Assistant dialog box)

Use this dialog box to add alternating colored rows.

To display this dialog box, select Insert > Standard > Table > Create Table Using Assistant > OK > Row Shading.

Alternating Row Colors

Adds color to alternate body rows using the specified number of rows and selected colors.

First

Specifies the number of rows to which the first color is applied.

Color Picker

Opens the color palette in which you select the color for the first row or set of rows.

Next

Specifies the number of rows to which the second color is added.

Color Picker

Opens the color palette in which you select the color for the second row or set of rows.

RELATED LINKS:

To create a table using the Table Assistant

Sections (Table Assistant dialog box)

Use this dialog box to add, remove, name, and order sections (grouping of rows that remain a unit) in the table.

To display this dialog box, select Insert > Standard > Table > Create Table Using Assistant > OK > Sections.

Has Body Rows And No Sections

Creates a table that contains the number of body rows indicated in the Body Layout dialog box; excludes sections.

Has Sections of Body Rows

Creates a table that contains the number of body rows indicated in the Body Layout dialog box and the sections indicated in the Sections list.

Sections

Lists the sections added to the table layout. Section1 is a placeholder section name, which you can rename by double-clicking the name.

Add

Adds a section and opens the Table Section dialog box, which you use to name the section.

Delete

Deletes the selected section.

Up Arrow

Moves the selected section up.

Down Arrow

Move the selected section down.

Section Has Header Row

Adds a header row to the selected section.

Section Has Footer Row

Adds a footer row to the selected section.

Section is Optional

Prevents the selected section from displaying if no data is available for that section.

RELATED LINKS:

To create a table using the Table Assistant

Template Manager dialog box

Use the Template Manager to preview templates and select a default template for creating form designs. The default template appears when the Template Manager opens.

To display this dialog box, select Tools > Template Manager.

Blank tab

Displays the available template sizes.

Forms tab

Displays the available sample templates.

Forms - Vanilla tab

Displays templates that have the same content as the ones on the Forms tab, but use an alternative, "vanilla" layout.

Forms - Beige tab

Displays templates that have the same content as the ones on the Forms tab, but use an alternative, "beige" layout.

Forms - Blue tab

Displays templates that have the same content as the ones on the Forms tab, but use an alternative, "blue" layout.

Other tab

Displays templates that you have added to the template manager.

Set Selected As Default

Sets the selected template as the default template for creating new form designs.

RELATED LINKS:

Creating and managing templates

To add a template to theTemplate Manager

To add a tab to the TemplateManager

Template Options dialog box

Use this dialog box to specify how Designer processes the template you import.

To display this dialog box, open a template file (TDS) that has not been added to the Template Manager.

Copy this template into the Template Manager

Adds the template file to the Template Manager.

Create a new form based on this template

Creates an untitled PDF form based on the template; you can name and save this form. If the template contains placeholders, the New Form Assistant opens, enabling you to change the placeholder information displayed in the various fields.

Edit this template

Opens the template file for editing.

23. Best Practices for Creating Adobe Forms Designs

Designer can help you create complex forms and deliver delightful digital experiences. However, it requires concerted effort to create and publish efficient forms on at a faster pace. The Best Practices for Creating Adobe Form Designs document provides guidelines and recommendations that forms developers can benefit from while working with designer. It discusses best practices right from selecting form content till up a forms development project to performance improvements. These best practices collectively contribute to the overall performance of forms.

24. Glossary

This glossary contains terminology definitions that are specific to Designer documentation. These terms may have different meanings in other contexts, but they have restricted meanings in this documentation.

24.1. A

accessible forms

Forms that users with disabilities or vision impairments can fill using screen readers and other assistive technologies.

Acrobat form

A PDF document created in Acrobat that contains one or more form fields. Acrobat forms may also contain non-form content.

Adobe certified document

A document that is signed with a specific Adobe root certificate. The Adobe root certificate is distributed on a hardware security module. An Adobe certified document provides a strong guarantee as to the authenticity and immutability of the document. See also *certificate*.

application

A set of related and generally interdependent files that make up a self-contained application that Forms can run.

24.2. B

binding data

The action of linking an object to the appropriate field within a data file or database.

boilerplate

Static or fixed objects, including text or graphics, that provide the visual components of a form design. Unlike data fields, the information in the boilerplate never changes.

boilerplate object

A static object on a form, such as text, a line, a circle, or a rectangle.

24.3. C

canonical format

A standard way of writing a formula. For example, two formulas, such as 9 + x and x + 9, are said to be equivalent because they mean the same thing; however, the second one is in "canonical form" because it is written in the usual way, with the highest power of x first. Usually, there are fixed rules you can use to decide whether something is in canonical form. Things in canonical form are easier to compare.

This table describes the different field types and their canonical formats. Note that values surrounded by square brackets are optional. Currency and grouping symbols are not valid in canonical format.

Field type	Description	Canonical format	Example
Date fields	An ISO-8601/XFA date string	YYYY[-MM[-DD]] YYYY[MM[DD]]	2005-07-04 20050704
Date/time fields	An ISO-8601/XFA date time, which is the concatenation of a valid ISO-8601/XFA date string and a valid ISO-8601/XFA time string with the letter 'T' as a separator between the date and time string		2004-07-04T10:11:12+05:00 20040704T101112
Numeric fields	A sequence of ASCII digits consisting of an integral part, a decimal point, a fractional part, and optionally, an e (or E) and a signed exponent part		12 1.234 .12 1e-2 1.2E3

Field type	Description	Canonical format	Example
Text fields	Any sequence of Unicode characters without using spaces		A1B2C3 Display Pattern: A9A 9A9 Formatted Value: A1B 2C3
Time fields	An ISO-8601/XFA time string	HH[:MM[:SS[.FFF][Z]]] HH[:MM[:SS[.FFF][+HH[:MM]]]] HH[:MM[:SS[.FFF][-HH[:MM]]]] HH[MM[SS[.FFF][Z]]] HH[MM[SS[.FFF][+HH[MM]]]] HH[MM[SS[.FFF][-HH[MM]]]]	10:11:12.123+05:00 101112+0500 10:11:12Z 10:11 10

certificate

A public key that corresponds to a credential (private key) used in encryption and signing operations. See also *Adobe certified document*.

client

The requesting program or person in a client/server relationship. A web browser is an example of a client application.

credential

A private key used to establish identity in decryption and signing operations.

24.4. F

fixed layout

A form whose layout remains exactly as it was designed when being filled with data. The layout does not change to accommodate the amount of incoming data.

flowable layout

A form, form design, or document layout that can expand or shrink to accommodate the amount of data being merged or entered by a user.

form

An electronic document that captures and delivers data. A person may add data to an interactive form, or a server process may merge a form design with data to produce a form.

form authors

Designer users who are capable of creating simple interactive and non-interactive forms that may contain basic calculations and scripts.

FormCalc

A calculation language similar to that used in common spreadsheet software that facilitates form design without requiring a knowledge of traditional scripting techniques or languages.

form design

The design-time version of a form that a form author or form developer creates in Designer.

form developers

Designer users who are capable of creating complex form-based applications for use in different environments. See also *form authors*.

form object

A form element, such as a button or text field, that you can place on a form. An object has its own set of properties and events.

24.5. I

interactive form

A form that a person can interact with and complete electronically both online (for example, PDF, guide, HTML, or Flex application) and offline (for example, PDF).

24.6. P

PDF form

Any PDF file that contains one or more form fields. A PDF form can be created in Acrobat or Designer.

permissions

Security settings that restrict users from opening, editing, or printing a PDF document. Permissions cannot be changed unless the user has the permissions password.

You can set permissions in Designer, Acrobat, or Forms .

prepopulated form

A form that appears to the end user with some or all fields automatically populated with data.

24.7. R

render

An action whereby Forms or Output retrieves a form design, possibly merges it with data, and displays it in PDF or HTML format in a browser.

restricted document

A PDF document with password security restrictions that prevent the document from being opened, printed, saved, or edited.

rights-enabled document

A PDF document that includes security extensions that enable Adobe Reader users to save the form with data, add comments, and sign documents.

run time

The time when an application or server process retrieves a form design, possibly merges it with data, and presents it to a user for viewing or filling.

24.8. S

script fragment

A piece of FormCalc or JavaScript code that can be reused across multiple XDP files.

security handler

A software module that implements various aspects of the encryption process and controls access to the contents of the encrypted document. It contains all security logic, such as how to sign and verify signatures.

static form

See *fixed layout*.

subform

An object that can act as a container for form objects and other subforms. A subform helps to position form objects relative to each other and provide structure in form designs that have a flowable layout. A subform can also provide a reference point, when binding data to a form, by restricting the scope for a field so that it matches that of the corresponding data node.

24.9. T

tagged PDF form

A form that includes a logical structure and a set of defined relationships and dependencies among the various elements, as well as additional information that permits reflow.

trusted identity

Recognized by the associated certificate. Only the specified trusted identities can sign a form.

24.10. U

usage rights

Rights that extend the functionality of Adobe Reader and enable users to save forms with data, add comments, and sign documents.

24.11. X

XML Forms Architecture

Represents the underlying technology beneath the Adobe XML forms solution. It enables the construction of robust and flexible form-based applications for use on either the client or the server.

XML form

A form based on the XML Forms Architecture. XML forms are created in Designer and can have the file name extension .xdp or .pdf.