

OpenText™ Embedded Output Transformation Engine

User Guide

This document provides information about the features and functionality of Output Transformation Engine. The information provided is intended for use by administrators and users.

VDTOTS240200-UTE-EN-2

OpenText™ Embedded Output Transformation Engine User Guide

VDTOTS240200-UTE-EN-2

Rev.: 2024-May-03

This documentation has been created for OpenText™ Embedded Output Transformation Engine CE 24.2.

It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111

Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440

Fax: +1-519-888-0677

Support: <https://support.opentext.com>

For more information, visit <https://www.opentext.com>

© 2024 Open Text

Patents may cover this product, see <https://www.opentext.com/patents>.

Disclaimer

No Warranties and Limitation of Liability

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Open Text Corporation and its affiliates accept no responsibility and offer no warranty whether expressed or implied, for the accuracy of this publication.

Table of Contents

1	Introduction	23
1.1	OpenText Output Transformation Server Engine	23
1.2	OpenText Output Transformation Designer	23
1.2.1	Rapid Application Development Features	24
1.3	Output Transformation Engine Terminology	24
1.4	Recommended Server Configuration	25
1.4.1	Scaling Recommendation	25
1.4.2	Memory Allocation Recommendations	25
2	System Configuration	27
2.1	System Configuration Parameters	28
2.1.1	Logging Parameters	35
2.1.1.1	SystemLog	35
2.1.1.2	JobLog	36
2.1.1.3	Filtering Messages	37
2.1.2	Job Manager Parameters	39
2.1.2.1	DefaultReportListener	39
2.1.2.2	MaxConcurrentJobs	39
2.1.2.3	CheckQueuedJobs	39
2.1.2.3.1	Benefit	39
2.1.2.3.2	Recommended Configuration	40
2.1.2.3.3	Notes	40
2.1.2.4	Verbose Job Report	41
2.1.2.5	Enable Job Report	41
2.1.2.6	Default Job Report Listener	41
2.1.2.7	Application Config Path	42
2.1.2.8	FdlcuTables	42
2.1.2.9	Console Messages	42
2.1.2.10	CheckHeapUsage	42
2.1.2.10.1	Benefit	43
2.1.2.10.2	Recommended Configuration	43
2.1.2.11	JVM Garbage Collection	43
2.1.2.12	JobProfiling	43
2.1.2.12.1	Benefit	44
2.1.3	JobStats Parameters	44
2.1.3.1	Enabled	44
2.1.3.2	CreateStatsFile	44
2.1.3.3	FdStatsFile	44
2.1.3.4	EnableFileHeader	44
2.1.3.5	EnableColumnHeader	44

2.1.3.6	DecimalFormat	45
2.1.3.7	DateTimeFormat	45
2.1.3.8	MillisecondFormat	45
2.1.3.9	MegabyteFormat	45
2.1.3.10	EnabledStats	45
2.1.4	Job Override Parameters	48
2.2	Using the Output Transformation System Manager	49
2.2.1	Opening a System Configuration File for Viewing or Editing	50
2.2.2	Finding the Location of a System Configuration File	51
2.2.3	Setting the Active System Configuration File	51
2.2.4	Restarting the System	52
2.3	System Variables	52
2.4	Reports and Log Files	55
2.4.1	System Log File	55
2.4.2	Debug Log	55
2.4.3	Error Log	56
2.4.4	Log File	56
2.4.5	Installer Log	56
2.4.6	Report Log	56
2.4.7	Managing Log Files	57
2.4.7.1	Report Manager	57
2.4.7.2	Log Message Options	57
2.5	Dynamically Renaming Reports	57
2.6	Tuning for Performance	58
2.6.1	Factors Affecting Performance	58
2.6.2	Performance Tuning Assumptions	60
2.6.3	Recommended Project Configurations	60
2.6.4	Dynamic Environment Memory Requirements Planning	62
2.6.4.1	Other Factors Influencing Memory Usage	63
2.6.5	Interpreting the FdStats File	64
2.7	Output Transformation Engine Configuration from the Classpath	66
2.7.1	Configuring the System Configuration File with a Classpath Argument	67
2.7.2	Modifying the Batch File with a Classpath Argument	67
2.7.3	Implementing a License File from a Classpath Argument	68
2.8	System Properties	69
2.8.1	Setting a Separate Work Directory	69
2.9	Running JavaScript	70
2.9.1	Maintaining JavaScript Security	70
2.9.2	JavaScript library compatibility	71
2.10	Packaged Embedded Output Transformation Engine File	71

3	Getting Started with Projects	73
3.1	Getting Started with Projects and Components	73
3.1.1	Fine Tuning a Project	74
3.1.2	Project Components	74
3.1.3	Creating a New Project	75
3.1.4	Adding Components to a Project	75
3.1.5	Viewing Component Resources	76
3.1.6	Editing a Component	77
3.1.7	Removing a Component	77
3.1.8	Removing a Component Connection	78
3.1.9	Special characters	78
3.2	Projects Overview	79
3.2.1	Project Checklist	79
3.2.2	Fine Tuning a Project	80
3.2.2.1	Creating a New Environment	80
3.2.2.1.1	Unmounting a File System	83
3.2.2.2	Opening a Project	83
3.2.2.3	Transformation Projects Tab	83
3.2.2.3.1	Project	84
3.2.2.3.2	Parsers	85
3.2.2.3.3	Generators	85
3.2.2.3.4	XFT	85
3.2.2.3.5	Tool Components	85
3.2.2.3.6	IO	86
3.2.2.3.7	XIF	86
3.2.2.3.8	MFCT	86
3.2.2.3.9	Font Environment Map	87
3.2.2.3.10	Results	87
3.2.2.4	Project Properties Tab	87
3.2.2.5	Page View Tab	90
3.2.2.5.1	Page View Toolbar	92
3.2.3	Wizards	93
3.2.3.1	Available Wizards	94
3.2.4	New Component Wizard	95
3.2.5	Switching Projects in Output Transformation Designer	98
3.3	Resources	98
3.3.1	Resources Tab	98
3.3.1.1	Resources Toolbar	100
3.3.2	File Paths	100
3.3.2.1	File Definitions	100
3.3.2.1.1	File Definition Lists (or FDef Lists)	101

3.3.2.1.2	Fdefs Using Variables	101
3.3.2.1.3	Order of Evaluation	102
3.3.2.1.4	Using Seekable Buffers	102
3.3.2.2	Select Resource Path/File Window	102
3.3.3	Resource Collection (IORES)	104
3.4	Project Workflow	105
3.4.1	Project Tab	106
3.4.2	Project Tab Workflow Area	107
3.4.2.1	Producers and Consumers	108
3.4.3	Project Tab Sidebar Menus	108
3.4.3.1	Parser Sidebar Menu	108
3.4.3.1.1	Adding a Parser to Your Project	109
3.4.3.2	Component Sidebar Menu	110
3.4.3.2.1	Adding a Component to Your Project	111
3.4.3.3	Generator Sidebar Menu	111
3.4.3.3.1	Adding a Generator to Your Project	112
3.5	I/O Utility	112
3.5.1	I/O Handlers	112
3.5.1.1	I/O Handler Factory Lists	112
3.5.1.2	The I/O Utility	113
3.5.1.2.1	System I/O Handlers	114
3.5.1.2.2	Application I/O Handlers	115
3.5.2	Quick Configuration of Predefined I/O Handlers	117
3.5.3	IORES Parameters	117
3.5.3.1	Examples of the IORES Parameter	118
3.5.4	TempPath Job Variable	118
3.5.5	I/O Directives	118
3.5.5.1	{NOCRLF}	119
3.5.5.2	{CRLF}	119
3.5.5.3	{AFP}	120
3.5.5.4	{AFPCRLF}	120
3.5.5.5	{AFPLINE}	120
3.5.5.6	{AFPLINE-CRLF}	120
3.5.5.7	{AFPLINE-LF}	120
3.5.5.8	{AFPNO5A}	121
3.5.5.9	{CRLF-DOS}	121
3.5.5.10	{CRLF-UNIX}	121
3.5.5.11	{Fnnn,nnn}	121
3.5.5.12	{LF***}...	121
3.5.5.13	{RDW}	122
3.5.5.14	{BDW}	122
3.5.5.15	{MVSFTP}	122

3.5.5.16	{METABAR}	122
3.5.5.17	{METABARR}	123
3.5.5.18	{METABAR2}	123
3.5.5.19	{METASPUR}	123
3.5.5.20	{LF2NM}	123
3.5.5.21	{CRLF-MVS}	124
3.6	XIF Options	124
3.6.1	XIF Options Parameters	124
3.6.1.1	XifFragmentJoining	124
3.6.1.2	XifLineJoining	127
3.6.1.3	XifShadeJoining	127
3.6.1.4	XifVectorOptions	127
3.6.1.5	XifDrawingOptions	128
3.6.1.6	XifRasterOptions	128
3.6.1.7	XifTextOptions	128
3.6.1.8	XifAcroFormOptions	129
3.6.1.9	XifImageOptions	129
4	Parsers	131
4.1	AFP Parser	132
4.1.1	AFP Parser Parameters	133
4.1.1.1	Setting Page Range Parameters	134
4.1.1.2	ContentFilter	134
4.1.1.3	SkipInlineResOptions	136
4.1.1.4	ExternalResources	136
4.1.1.5	UnresolvedResourceOption	137
4.1.1.6	OptimizedAfpOptions	137
4.1.2	AFP Parser Wizard	138
4.1.2.1	Launching the AFP Parser Wizard	138
4.1.2.1.1	Identify Your Parser	138
4.1.2.1.2	Select AFP Form and Page Definition	139
4.1.2.1.3	Select AFP Record Format	140
4.1.2.1.4	Select Additional Settings	142
4.1.3	Parsing AFP Data Utilizing OpenType Fonts	143
4.2	Image Parser	144
4.2.1	Image Parser Parameters	145
4.2.1.1	Setting Page Range Parameters	146
4.2.2	Image Parser Wizard	146
4.2.2.1	Launching the Image Parser Wizard	146
4.2.2.1.1	Identify Your Image Parser	147
4.2.2.1.2	Select Image Input Details	147
4.2.2.1.3	Select Property Settings	148

4.3	Linedata Parser	148
4.3.1	Linedata Parser Parameters	149
4.3.2	Using OpenType Fonts with the Linedata Parser	151
4.3.2.1	Step 1: Importing the TrueType Font into the Font Environment Map	152
4.3.2.2	Step 2: Adding the TrueType Font as an OpenType Font in the Linedata Parser Properties	153
4.3.2.3	Step 3: Adding the New OpenType Font in the MFCT and Mapping it to an Output Font	153
4.3.3	Using Double-Byte Characters with the Linedata Parser	154
4.4	Metacode Parser	155
4.4.1	Metacode Parser Parameters	156
4.4.2	Metacode Parser Wizard	157
4.4.2.1	Launching the Metacode Parser Wizard	157
4.4.2.1.1	Identify Your Parser	157
4.4.2.1.2	Select Metacode Input Details	158
4.4.2.1.3	Select Property Settings	158
4.5	PCL Parser	159
4.5.1	Supported Formats	159
4.5.2	PCL Parser Parameters	160
4.5.2.1	Setting Page Range Parameters	160
4.5.3	PCL Parser Wizard	161
4.5.3.1	Launching the PCL Parser Wizard	161
4.5.3.1.1	Identify Your Parser	161
4.5.3.1.2	Select PCL Input Details	162
4.5.3.1.3	Set Page Attributes	162
4.6	PDF Parser	163
4.6.1	Supported Formats	163
4.6.2	PDF Parser Parameters	164
4.6.2.1	Setting Page Range Parameters	165
4.6.2.2	PreserveTags	165
4.6.2.3	PreserveInteractiveForm	166
4.6.2.4	ProcessDigitalSignatures	166
4.6.3	PDF Parser Wizard	166
4.6.3.1	Launching the PDF Parser Wizard	166
4.6.3.1.1	Identify Your Parser	167
4.6.3.1.2	Select PDF Input Details	167
4.6.3.1.3	Select Property Settings	168
4.7	TIFF Parser	168
4.7.1	TIFF Parser Parameters	170
4.7.1.1	Setting Page Range Parameters	170
4.7.2	TIFF Parser Wizard	170
4.7.2.1	Launching the TIFF Parser Wizard	171

4.7.2.1.1	Identify Your Parser	171
4.7.2.1.2	Select TIFF Input Details	172
4.7.2.1.3	Select Property Settings	172
5	Generators	173
5.1	Accessible PDF Generator	174
5.1.1	Using the Accessible PDF Generator	174
5.1.2	Defining Alternate Text	176
5.1.2.1	Defining and Packaging Alternate Text	176
5.1.2.2	Editing an Existing Alternate Text Package	178
5.1.2.3	Testing the Output for Accessibility	179
5.1.3	Preserving Invisible Text in PDF Documents	179
5.2	AFP Generator	180
5.2.1	AFP Generator Parameters	180
5.2.1.1	AcifOptions Parameter	181
5.2.1.2	DefineFeed Parameter	182
5.2.1.3	GocaOptions Parameter	184
5.2.1.4	ImageOptions Parameter	186
5.2.2	AFP Generator Wizard	189
5.2.2.1	Launching the AFP Generator Wizard	189
5.2.2.1.1	Identify Your Generator	189
5.2.2.1.2	Select AFP Output Details	190
5.2.2.1.3	Select Property Settings	190
5.3	Data Generator	191
5.4	Image Generator	191
5.4.1	Image Generator Parameters	191
5.4.2	Image Generator Wizard	192
5.4.2.1	Launching the Image Generator Wizard	192
5.4.2.1.1	Identify Your Generator	192
5.4.2.1.2	Select Image Output Details	193
5.4.2.1.3	Select Property Settings	193
5.5	Metacode Generator	194
5.5.1	Applications	194
5.5.2	Generated Formats	194
5.5.3	Supported Features	194
5.5.4	Limitations	195
5.5.5	Metacode Generator Parameters	195
5.5.6	Metacode Generator Wizard	196
5.5.6.1	Launching the Metacode Generator Wizard	196
5.5.6.1.1	Identify Your Generator	196
5.5.6.1.2	Select Metacode Output Details	197
5.5.6.1.3	Specify JSL/DJDE Attributes	197

5.5.6.1.4	Specify Printer Attributes	198
5.5.6.1.5	Specify Data Element Options	199
5.6	PCL Generator	199
5.6.1	Applications	199
5.6.2	Supported Features	199
5.6.3	Limitations	200
5.6.4	PCL Generator Parameters	200
5.6.5	PCL Generator Wizard	200
5.6.5.1	Launching the PCL Generator Wizard	200
5.6.5.1.1	Identify Your Generator	201
5.6.5.1.2	Select PCL Output Details	201
5.6.5.1.3	Select Property Settings	202
5.7	PDF Generator	203
5.7.1	PDF Output	204
5.7.2	PDF File Linearization	204
5.7.3	Substituting Colors for Patterns in PDFs with Shade Mapping	205
5.7.3.1	Step 1: Setting the PDF Generator's ShadeMode Parameter	205
5.7.3.2	Step 2: Acquiring the Mapping Names for Unmapped Shades	205
5.7.3.3	Step 3: Configuring the Mapping Keys for Patterns	207
5.7.4	PDF Generator Parameters	208
5.7.4.1	AcrobatOpen Parameters	208
5.7.5	PDF Generator Wizard	211
5.7.5.1	Launching the PDF Generator Wizard	211
5.7.5.1.1	Identify Your Generator	211
5.7.5.1.2	Select PDF Output Details	212
5.7.5.1.3	Select PDF Features	212
5.7.5.1.4	Select Basic Document Options	213
5.7.5.1.5	Select Compression Options	214
5.7.5.1.6	Select Open View Options	214
5.7.5.1.7	Select Security and Encryption Options	216
5.7.5.1.8	Specify Authorship	217
5.8	PDF/A Generator	217
5.8.1	PDF/A Output	218
5.8.1.1	PDF/A Generator Parameters	219
5.8.1.2	PDF/A Generator Wizard	219
5.8.1.2.1	Launching the PDF/A Generator Wizard	219
5.8.1.2.2	Identify Your Generator	220
5.8.1.2.3	Select PDF/A Output Details	220
5.8.1.2.4	Select PDF/A Features	221
5.8.1.2.5	Select Basic Document Options	221
5.8.1.2.6	Select Compression Options	222
5.8.1.2.7	Select Open View Options	222

5.8.1.2.8	Specify Authorship	224
5.9	PostScript Generator	224
5.9.1	Applications	225
5.9.2	Supported Features	225
5.9.3	Limitations	225
5.9.4	PostScript Generator Parameters	225
5.9.5	PostScript Generator Wizard	226
5.9.5.1	Launching the PostScript Generator Wizard	226
5.9.5.1.1	Identify Your Generator	226
5.9.5.1.2	Select PostScript Output Details	227
5.9.5.1.3	Select Property Settings	227
5.10	TIFF Generator	228
5.10.1	TIFF Generator Parameters	228
5.10.2	TIFF Generator Wizard	229
5.10.2.1	Launching the TIFF Generator Wizard	229
5.10.2.1.1	Identify Your Generator	229
5.10.2.1.2	Select TIFF Output Details	230
5.10.2.1.3	Specify Image Settings	230
5.11	Web Generator	231
5.11.1	Web Generator Features	231
5.11.2	Known Limitations	232
5.11.3	Web Generator Parameters	232
5.11.4	Using the Image Replacement Feature	232
5.11.5	Web Generator Wizard	233
5.11.5.1	Identify Your Generator	234
5.11.5.2	Select Web Output Details	234
5.11.5.3	Select Property Settings	235
6	Components	237
6.1	Best Viewed Rotation Component	237
6.1.1	Best Viewed Rotation Wizard	237
6.2	Constant Print Component	238
6.2.1	Constant Print Requirements and Limitations	239
6.2.2	Constant Print Window	241
6.2.2.1	Constant Print Toolbar	243
6.2.3	Constant Print Control	244
6.2.3.1	Constant Print Operation Designer	244
6.2.3.1.1	Index Entries	245
6.2.3.1.2	Creating a New Match Definition	247
6.2.3.1.3	Editing a Match Definition	248
6.2.3.1.4	Removing a Match Definition	248
6.2.3.2	Constant Text	248

6.2.3.2.1	Constant Print - Text Window	249
6.2.3.2.2	Constant Print – Text Properties	250
6.2.3.3	Constant Image	250
6.2.3.3.1	Constant Print - Image Window	251
6.2.3.4	Constant Background Gradient	253
6.2.3.5	Constant Grid	255
6.2.3.6	Constant Form	256
6.2.3.7	Constant Project	257
6.2.3.8	Constant PDF	259
6.2.3.9	Color Picker Window	261
6.3	Document Breaking Component	262
6.3.1	Document Breaking Wizard	263
6.3.1.1	Document Breaking Tab	263
6.3.1.2	Document Naming Tab	264
6.4	Field Modifier Component	265
6.4.1	Field Modifier Window	265
6.4.1.1	Field Modifier Design Pane	266
6.4.1.2	Field Modifier Toolbar	267
6.4.2	Field Modifier Operations	268
6.4.2.1	Adding a Field Modifier Entry	269
6.4.2.2	Editing a Field Modifier Entry	271
6.4.2.3	Removing a Field Modifier Entry	271
6.4.2.4	Setting the Field Modifier Entry List Run Order	271
6.5	Index Writer Component	272
6.5.1	Index Writer Wizard	272
6.5.2	IBM CMOD Index Writer	272
6.5.2.1	Creating Stacked Output for CMOD Loading	273
6.5.3	XML Index Writer	278
6.5.4	FileNet Index Writer	278
6.5.4.1	FileNet Index Writer Parameters	279
6.5.4.2	Example FileNet Index File	280
6.5.5	Custom Index Writers	281
6.6	Indexer Component	281
6.6.1	Indexer Wizard	282
6.6.1.1	Indexing tab	283
6.6.1.2	Document Breaking tab	283
6.6.1.3	Output Document Naming tab	284
6.6.1.4	Document Merging tab	285
6.6.1.5	XFT Indexing tab	285
6.6.1.6	TLE Indexing tab	286
6.6.1.7	Comment Index tab	286
6.6.1.8	Custom Properties Index tab	288

6.6.1.9	Original Document Indexing tab	288
6.6.1.10	XFT Doc Index Filter tab	289
6.6.1.11	Comment Index tab	290
6.6.1.11.1	Adding and editing comment entries	291
6.6.2	Indexer parameters	293
6.7	Large Print Component	293
6.7.1	Getting Started with the Large Print Component	294
6.8	Page Markup Component	295
6.8.1	Configuring the Page Markup Component	295
6.8.1.1	Tagging Elements as Artifacts	300
6.8.1.2	Modifying Page Markup Properties	301
6.8.1.3	Modifying Document Properties	302
6.8.1.4	Modifying Section Properties	302
6.8.1.5	Modifying Field Properties	304
6.8.1.6	Page Markup Text Field Properties	304
6.8.1.7	Page Markup Object Field Properties	309
6.8.1.8	Page Markup Table Field Properties	313
6.8.2	Tagging Table of Contents Structures Using Page Markup for use with Accessible PDFs	318
6.8.3	Managing Alternate Text Providers	321
6.9	Replace Text Component	324
6.9.1	Replace Text Wizard	324
6.9.1.1	Adding or Editing a Replace Text Field Condition	325
6.9.1.2	Removing a Replace Text Field Condition	326
6.9.2	Field Validators	326
6.10	Script Component	326
6.10.1	Script Component Wizard	327
6.11	Simple Branch Component	327
6.12	Skip Page Component	327
6.12.1	Skip Page Wizard	328
6.12.1.1	Adding or Editing a Skip Page Condition	328
6.12.1.2	Removing a Skip Page Condition	329
6.12.2	Skip Page Classes	329
6.13	Table of Contents Component	330
6.13.1	Table of Contents Wizard	330
6.13.1.1	Table of Contents Wizard Controls	331
6.13.1.2	Table Properties Tab	332
6.13.1.3	Entry Properties Tab	333
6.13.2	Multiple Root Nodes	334
6.14	URL Link Component	335
6.14.1	URL Link Wizard	336
6.14.1.1	URL XFT Field Options tab	336

6.14.1.2	URL Index Entry Options tab	337
6.14.1.3	URL Extraction Settings tab	337
6.14.1.4	URL Link General tab	338
6.14.1.5	Adding or Editing URL links	339
6.15	XFT Field Technology Component	340
6.15.1	XFT Essentials	340
6.15.1.1	XFT Field Technology	341
6.15.1.2	XFT Documents	341
6.15.1.3	XFT Component Properties	342
6.15.1.3.1	Running projects using XFT1 vs. XFT2	344
6.15.1.3.2	Detecting and tagging special characters	344
6.15.1.4	Identifiers	345
6.15.1.5	Confidence Factor	345
6.15.1.6	Field Definitions	346
6.15.1.7	Field Definition Terminology	348
6.15.2	XFT Window	350
6.15.2.1	XFT Window Toolbar	351
6.15.2.2	DOM Viewer	355
6.15.3	Getting Started with XFT	356
6.15.3.1	Creating a Field Definition	356
6.15.3.2	Handling Field Definitions	357
6.15.3.3	Auto-Detecting and Splitting Sections, Fields, or Table Columns	359
6.15.3.3.1	Detecting Sections	359
6.15.3.3.2	Splitting Sections	361
6.15.3.3.3	Detecting Fields	363
6.15.3.3.4	Splitting Fields	365
6.15.3.3.5	Detecting Table Columns	367
6.15.3.3.6	Splitting Table Columns	368
6.15.3.4	Adding a Document	370
6.15.3.4.1	Document Properties General Tab	371
6.15.3.4.2	Document Properties Start Identification Tab	371
6.15.3.4.3	Document Properties End Identification Tab	373
6.15.3.4.4	Document Properties DOM Properties Tab	374
6.15.3.5	Adding a Section	375
6.15.3.5.1	Section Properties General Tab	376
6.15.3.5.2	Section Properties Start Identification Tab	377
6.15.3.5.3	Section Properties Positioning Tab	378
6.15.3.5.4	Section Properties End Identification Tab	381
6.15.3.5.5	Section Properties DOM Properties Tab	382
6.15.3.5.6	Section Properties Auto Detect Tab	384
6.15.3.5.7	Configuring Multiuse Locators for Sections	391
6.15.3.6	Adding a Text Field	393

6.15.3.6.1	Text Field Properties General Tab	393
6.15.3.6.2	Text Field Properties Identification Tab	394
6.15.3.6.3	Text Field Properties Positioning Tab	395
6.15.3.6.4	Text Field Properties Extraction Tab	398
6.15.3.6.5	Text Field Properties DOM Properties Tab	399
6.15.3.7	Adding an Object Field	401
6.15.3.7.1	Object Field Properties General Tab	402
6.15.3.7.2	Object Field Properties Identification Tab	403
6.15.3.7.3	Object Field Properties Positioning Tab	404
6.15.3.7.4	Object Field Properties Extraction Tab	406
6.15.3.7.5	Object Field Properties DOM Properties Tab	407
6.15.3.8	Adding a Form Field	409
6.15.3.8.1	Form Field Properties General Tab	409
6.15.3.8.2	Form Field Properties Identification Tab	411
6.15.3.8.3	Form Field Properties Positioning Tab	412
6.15.3.8.4	Form Field Properties Extraction Tab	414
6.15.3.8.5	Form Field Properties DOM Properties Tab	415
6.15.3.8.6	Form Field Properties Interactive Extraction tab	416
6.15.3.9	Adding a List Field	417
6.15.3.9.1	Tagging List Field Items and Accessible PDFs	417
6.15.3.9.2	List Field Properties General Tab	417
6.15.3.9.3	List Field Properties Start Identification Tab	418
6.15.3.9.4	List Field Properties Positioning Tab	419
6.15.3.9.5	List Field Properties End Identification Tab	422
6.15.3.9.6	List Field Properties Row Breakers Tab	423
6.15.3.9.7	Detecting Nested Lists	428
6.15.3.9.8	List Field Properties Extraction Tab	428
6.15.3.9.9	List Field Properties DOM Properties Tab	429
6.15.3.9.1	Configuring Multiuse Locators for List Fields	430
0		
6.15.3.10	Adding a Table	432
6.15.3.10.	Table Properties General Tab	432
1		
6.15.3.10.	Table Properties Start Identification Tab	433
2		
6.15.3.10.	Table Properties Positioning Tab	434
3		
6.15.3.10.	Table Properties End Identification Tab	437
4		
6.15.3.10.	Table Properties Columns Tab	438
5		
6.15.3.10.	Table Properties Rows Tab	439
6		

6.15.3.10. Table Properties Cells tab	445
7	
6.15.3.10. Table Properties Extraction Tab	447
8	
6.15.3.10. Table Properties DOM Properties Tab	447
9	
6.15.3.10. Configuring Multiuse Locators for Table Fields	449
10	
6.15.3.11 Adding a Column	451
6.15.3.11. Column Properties General Tab	451
1	
6.15.3.12 Adding a Cell	452
6.15.3.12. Cell Properties General Tab	453
1	
6.15.3.12. Cell Properties DOM Properties Tab	453
2	
6.15.3.12. Cell Properties Auto Detect Tab	455
3	
6.15.3.12. XFT Cell Best Practices	462
4	
6.15.3.13 Adding a Threshold	462
6.15.3.13. Threshold Properties General Tabs	463
1	
6.15.3.13. Font Based Thresholds Tab	464
2	
6.15.3.14 Adding a Container	465
6.15.3.14. Container Properties General Tab	466
1	
6.15.3.14. Container Properties Start Identification Tab	467
2	
6.15.3.14. Container Properties Positioning Tab	468
3	
6.15.3.14. Container Properties End Identification Tab	470
4	
6.15.3.14. Container Properties DOM Properties Tab	471
5	
6.15.3.14. Container Properties Auto Detect Tab	473
6	
6.15.3.15 Adding a Group	479
6.15.3.15. Adding a Text Field Group	480
1	
6.15.3.15. Adding a List Field Group	481
2	
6.15.3.15. Adding a Table Field Group	482
3	

6.15.3.15. Managing XFT Groups	484
4	
6.15.3.15. Group Properties	485
5	
6.15.3.16 Adding a Text Locator	486
6.15.3.16. Text Locator Properties General Tab	486
1	
6.15.3.16. Text Locator Properties Extraction Settings Tab	487
2	
6.15.3.17 Adding an Image Locator	487
6.15.3.17. Image Locator Properties General Tab	488
1	
6.15.3.17. Image Locator Properties Extraction Settings	488
2	
6.15.3.18 Adding a Graphic Locator	489
6.15.3.18. Graphic Locator Properties General Tab	489
1	
6.15.3.18. Graphic Locator Properties Graphic Object Models Tab	489
2	
6.15.3.19 Adding a Graphic Object Model	489
6.15.3.19. Graphic Object Model Properties General Tab	490
1	
6.15.3.19. Graphic Object Model Properties Extraction Settings Tab	491
2	
6.15.3.20 Using Graphic Object Models to Identify Content	491
6.15.3.21 XFT Covering Letter Example	492
6.15.4 XIF Text Element Properties	494
6.15.4.1 Font Information Tab	495
6.15.4.2 Character Information Tab	496
6.15.4.3 Positional Information Tab	497
6.16 XFT Writer Component	498
6.16.1 XFT Writer Wizard	499
6.16.1.1 Adding or Editing an Output File	499
6.16.1.2 Adding or Editing an Output Index File	500
7 Font Correlation	501
7.1 Output Transformation Engine Font Mapping	501
7.1.1 Font Correlation Overview	501
7.2 Master Font Correlation Table	502
7.2.1 MFCT Tips	503
7.2.2 Managing an MFCT within a Project	504
7.2.3 ICU Tables	505
7.2.3.1 Why you need ICU Capability in Output Transformation Engine	506
7.3 MFCT Window	506

7.3.1	Resizing a Master Font Correlation Table	507
7.3.2	Sorting a Master Font Correlation Table	507
7.3.3	MFCT Toolbar	508
7.3.4	MFCT Window Sidebar Menu	508
7.3.5	MFCT Fonts Sidebar Menu	509
7.3.5.1	AFP Fonts	510
7.3.5.1.1	AFP Typeface Input Fonts Table	510
7.3.5.1.2	AFP Charset Input Fonts Table	512
7.3.5.1.3	AFP OpenType Input Fonts Table	513
7.3.5.1.4	AFP Font Mapping Attributes	515
7.3.5.2	Metacode Fonts	516
7.3.5.2.1	Metacode Input Font Family Groups Table	516
7.3.5.2.2	Metacode Input Family Mappings Table	518
7.3.5.2.3	Metacode Input Font Mappings Table	519
7.3.5.2.4	Metacode Font Mapping Attributes	520
7.3.5.3	PCL Fonts	521
7.3.5.3.1	PCL Input Fonts Table	521
7.3.5.3.2	PCL XL Input Fonts Table	523
7.3.5.3.3	PCL Soft Input Fonts Table	525
7.3.5.3.4	PCL Font Mapping Attributes	527
7.3.5.3.5	PCL Typeface Values	529
7.3.5.4	PDF Fonts	533
7.3.5.4.1	PDF Input Font Mappings Table	533
7.3.5.4.2	PDF Font Mapping Attributes	535
7.3.5.5	Output Fonts	535
7.3.5.5.1	Output Fonts Table	536
7.3.5.5.2	Adding an Output Font	536
7.3.5.5.3	Editing an Output Font	537
7.3.5.5.4	Removing an Output Font	537
7.3.5.5.5	AFP Output Font Families	537
7.3.5.5.6	HTML Output Font Families	541
7.3.5.5.7	Output Font Mapping Attributes	543
7.3.6	MFCT Colors Sidebar Menu	548
7.3.6.1	Color Table	549
7.3.6.1.1	Adding a Color Table	550
7.3.6.1.2	Adding a Color to a Color Table	550
7.3.6.2	AFP Color Tables	551
7.3.6.2.1	Adding a Color to an AFP Color Table	552
7.3.6.3	Metacode Color Tables	552
7.3.6.3.1	Adding a Color to a Metacode Color Table	553
7.3.7	MFCT Translations Sidebar Menu	554
7.3.7.1	Translation Tables	554

7.3.7.1.1	Adding a New Translation Table	556
7.3.7.1.2	Removing a Translation Table	556
7.3.7.1.3	Adding a Translation	556
7.3.7.1.4	Editing a Translation	557
7.3.7.1.5	Removing a Translation	557
7.3.7.1.6	Unicode Browser	557
7.3.7.2	AFP Glyph Table	559
7.3.7.2.1	Adding a Glyph Mapping	559
7.3.7.2.2	Editing a Glyph Mapping	560
7.3.7.2.3	Removing a Glyph Mapping	560
7.3.8	Finding Unicode Values in MS Windows	560
7.3.9	Manage Fonts	562
7.3.10	Map Glyph Fonts	563
7.3.10.1	Glyph Mappings Controls	564
7.4	Font Environment Map	565
7.4.1	Catalog Manager	566
7.4.1.1	Importing a Folder	567
7.4.1.2	Using the Import Fonts Wizard	567
7.4.1.2.1	Choosing the fonts files and/or directories to import	567
7.4.1.2.2	Choosing import options in the Import Fonts Wizard	568
7.4.1.2.3	Removing font files and/or directories from the Import Fonts wizard ..	569
7.4.1.3	Catalog Table	569
7.4.2	Repository Manager	570
7.4.2.1	Adding a Repository	571
7.4.2.2	Editing a Repository	571
7.4.2.3	Removing a Repository	571
7.4.3	Font Convert Tool	572
8	Project Review	575
8.1	Reviewing Your Project	575
8.1.1	Reviewing Font Resources	575
8.2	The Input File	576
8.2.1	What is the input file?	576
8.2.1.1	Acceptable AFP Input	576
8.2.2	AFP Resources	577
8.2.2.1	Page Definition Resources	577
8.2.2.2	Form Definition Resources	577
8.2.2.3	Font Resources	578
8.2.2.4	Overlay Resources	578
8.2.2.5	Page Segment Resources	579
8.2.2.6	ACIF Resource Group	579
8.3	AFP Font Resources	579

8.3.1	Character Sets	580
8.3.2	Output Transformation Engine Character Sets	581
8.3.2.1	Character Set Attributes	581
8.3.2.2	Glyph Attributes	582
8.3.3	Codepage	582
8.3.3.1	Codepage Formats	583
8.3.3.2	Codepage Resources	583
8.3.4	Coded Font	583
8.4	AFP File Transfer Issues	584
8.4.1	AFP Resources and Fully Composed AFP Documents	584
8.4.2	For AFP Line Data and Mixed Mode Files	584
8.5	File Extension Recognition	584
8.5.1	Output Transformation Engine File Extensions	584
8.5.2	AFP File Extensions	585
8.5.3	PDF File Extensions	585
9	Running a Project	587
9.1	Processing Overview	587
9.1.1	Running the Project	587
9.1.2	Locating Your Output	587
9.2	Job Execution Options Window	588
9.2.1	Job Options Tab	588
9.2.2	Job Variables Tab	589
9.2.2.1	Adding New Job Variables	589
9.2.2.2	Removing a Job Variable	589
9.2.3	Sources Tab	589
9.3	Input Monitors	590
9.4	Running in Production	590
9.4.1	Transferring Configuration Files	590
9.4.1.1	Files from Development	590
9.4.1.2	UNIX Servers	591
9.4.1.3	IBM Mainframe (z/OS) Servers	591
9.4.1.3.1	Properly Transferring XML Files from Windows to z/OS	591
9.5	Command Line Operation	592
9.5.1	Usage	592
9.5.1.1	Specifying the ClassPath File	593
9.5.1.2	Specifying the Configuration File	593
9.5.1.3	Specifying the Project File	594
9.5.1.4	Optional Command Line Arguments	595
9.5.1.5	Command Line Example	598
9.5.2	Input Monitors	598
9.5.2.1	Input Monitor Setup	599

9.5.2.2	System Configuration Parameter File	599
9.5.2.3	Watch Directory Configuration File	600
9.5.2.3.1	Job Variable Support	602
9.5.2.4	Starting an Input Listener	604
9.5.2.5	Stopping an Input Listener	604
9.5.3	Resource Collection IORES	605
9.5.3.1	IORES Parameters	605
9.5.3.2	IORES I/O Directives	607
9.5.3.3	Creating a Resource Collection	607
9.5.3.4	IORES and Resource Paths	608
9.5.3.5	Running a Job with an IORES File	610
9.5.3.6	IORES Utilities	610
9.5.3.6.1	IORES Builder Utility	610
9.5.3.6.2	Zip2IoRes Utility	612
9.5.3.6.3	IoRes2Zip Utility	613
10	Reviewing the Output	615
10.1	The Output File	615
10.1.1	Reviewing Your Output File	615
10.2	Reviewing the Output tab	615
10.2.1	Job Stats tab	616
10.2.2	Log tab	617
10.2.3	Variables tab	618
10.2.4	Results tab	619
10.3	Messages	620
10.3.1	Message Format	620
10.3.2	Message Categories	621
10.3.3	Message Severity	622
10.3.3.1	AFP Parser Messages	622
10.3.3.2	Command Messages	625
10.3.3.3	Generator Messages	627
10.3.3.4	Input Listener Messages	628
10.3.3.5	Job Manager Messages	628
10.3.3.6	Job Messages	630
10.3.3.7	License Messages	632
10.3.3.8	MFCT Messages	633
10.3.3.9	Parameter Messages	634
10.3.3.10	Parser Messages	635
10.3.3.11	PDF Generator Messages	636
10.3.3.12	Resource Manager Messages	637
10.3.3.13	System Messages	638
10.3.3.14	TIFF Generator Messages	640

10.3.3.15	Table of Contents (TOC) Messages	640
10.3.3.16	XFT Messages	642
11	Troubleshooting	647
11.1	First Steps	647
11.2	Look for Possible Causes	648
11.3	Check the Report Log	648
11.4	Check the Parameters	649
11.5	Review Font Table	650
11.5.1	Is the font being mapped?	650
11.5.2	Is the font mapped twice by family and by font/character set?	650
11.5.3	Is the point size correct?	650
11.5.4	Is the font being rasterized?	650
11.5.5	Is a color table defined?	650
11.5.6	Are the colors defined correctly?	651
11.5.7	Are external filenames specified?	651
11.6	View the Diagnostic Log	651
11.7	Specific Problems and Solutions	651
11.7.1	Output File Does Not Exist	651
11.7.2	Output Text is Messy	652
11.7.3	Output Text is Unreadable	653
11.7.4	Windows Not Visible	654
11.7.5	Windows Display Problems	654
11.7.6	No Project Window	654
11.7.7	Error Occurred During Open	655
11.7.8	Data not Picked Up by Fields	656
11.8	Frequently Asked Questions	656

Chapter 1

Introduction

OpenText™ Embedded Output Transformation Engine repurposes documents from their printed formats into electronic formats, supporting ebusiness, archive and content management strategies, with no change to existing applications.

Output Transformation Engine and its powerful API, written in Java, are designed for rapid integration with e-business applications. The scalable component architecture of the application enables flexible implementation and high-volume performance. The application also addresses the needs for department and enterprise-wide document processing and e-content presentment solutions, and brings to market many innovative capabilities in response to and anticipating significant growth in our global customers' requirements.

Output Transformation Engine can be installed as a standalone product or with Output Transformation Server. If installed with Output Transformation Server, you will see a tabbed pane layout within the Output Transformation Designer interface that allows you to open multiple products simultaneously and switch from one resource to another.

1.1 OpenText Output Transformation Server Engine

Output Transformation Server allows the high performance transformation products, events, and process flows to be deployed to a JEE application server. Once deployed, the projects can be run either from the events themselves, or they can be run from a session bean, JMS queue, SOAP/Web service, or interactively from Output Transformation Designer. There is also a common API which allows projects to be called directly from Java.

1.2 OpenText Output Transformation Designer

Written in Java, Output Transformation Designer allows you to define and configure all of the elements you need to process, add value, transform, generate and repurpose documents to print and e-content formats. It also allows you to configure the products and deploy project files and resources to the application server.

For more information on using Output Transformation Designer, see *OpenText Output Transformation Designer - User Guide (VDTOTS-H-UTD)*.

1.2.1 Rapid Application Development Features

With Output Transformation Designer, you can easily:

- Define source (input) printstreams.
- Define destination (output) files.
- Use the “[Project Tab](#)” on [page 106](#) to connect components in the workflow.
- Modify component parameters.
- Define the location of supporting resources, for example: fonts, images, and overlays.
- Define I/O directives, for example, variable or fixed length record types.
- Correlate fonts in the input printstream to those in the output file format.
- Do rapid font correlation using font families.
- Use standard International Components for Unicode (ICU) tables to map AFP encodings into Unicode.
- Use auto-resource preview of overlays or page segments for fast verification.
- Use AWT (Abstract Windows Toolkit) font mapping using TrueType fonts.
- Use real time feedback from the engine on how fonts were correlated, for rapid proofing of font correlation.
- Identify URLs and Table of Contents, including bookmarks, for insertion in the output document.
- Identify regions in source documents from which to extract field data.

1.3 Output Transformation Engine Terminology

There are a few basic terms to be aware of within context of the product.

Application

An application is a subdirectory of a mounted file system which contains projects and the resource files specific to those projects.

Project

A project is a `.d2proj` file contained in the **Application** subdirectory. This is the project file containing the parser, generator(s), and component parameters. There can be any number of projects within an application.

Project names typically have the application name in it, such as `sample-afp2pd`, which is the project for converting the application “sample” from AFP into PDF.

For more information on working with projects, see “[Getting Started with Projects](#)” on [page 73](#).

Job

A job is a running project. Every time a project is run, the application's engine assigns a unique name for the job, consisting of the project name followed by a unique numeric identifier. For more information on engines, see [“Adding a Color Table” on page 550](#).

The job name may be something like `car-afp2pdf-0-1234567890`.

Each job will generate a report file using that job name, unless job reports have been disabled in System Configuration via the **JobManager** parameter.

For more information on System Configuration, see [“System Configuration Parameters” on page 28](#).

1.4 Recommended Server Configuration

For server requirements, see *OpenText Output Transformation Server - Installation guide (VDTOTS-IGD)*.

1.4.1 Scaling Recommendation

Where multiple processors (assume eight for the sake of argument) are required to support demand, we recommend using two quad-processor servers rather than four dual-processor servers in a load-balanced configuration.

1.4.2 Memory Allocation Recommendations

- Initial JVM heap size should be set to support more than the maximum load expected. Use the [“JobProfiling” on page 43](#) parameter to assess this prior to production.
- For example, if one job consumes 50MB and 5 concurrent jobs need to be processed, then the minimum memory required would be:
- $50 \times 5 = 250\text{MB}$ + memory for Output Transformation Engine overhead.
- Set the initial JVM heap size (`-Xms` command line parameter) to 4 times that calculation.
- Set the `<MaxHeapUsage>` parameter in the system configuration file (`.d2eSys`) to two times that calculation. This gives a buffer of two times the maximum expected heap usage to allow for unexpected variations in load and data type. If there is less memory available than indicated by the `<MaxHeapUsage>` parameter, jobs are intentionally failed.
- Use the `<MaxQueuedJobs>` parameter to limit the number of jobs that may be held in the queue when the system is busy running other jobs. There is some memory consumed in storing each queued job, so this can act as a safety valve to avoid `OutOfMemoryErrors` if the system is suddenly inundated with hundreds of requests. Additional jobs that are received after the `MaxQueuedJobs` limit has

been reached will be rejected with a return code of 28 (not enough resources available).

For more information on fine tuning the performance, see [“Tuning for Performance” on page 58](#).

Chapter 2

System Configuration

Each instance of Output Transformation Engine is started with its own default system configuration file (*.d2esys), but you can configure your own according to your project's requirements. This file is an XML format file and is used to view and edit the system configuration parameters, which set system-wide properties that affect all aspects of your Output Transformation Engine projects. The default system configuration file is named `ote-config.d2esys` and its default location is the `<install_home>\initialFiles\common\_configs\` folder.

The `.d2esys` file contains pointers to:

- Job variables
- Localized message file (default messages provided in English)
- Default report path
- ICU table (used in AFP font mapping)
- License file
- Job manager
- Resource manager
- Log options



Note: You cannot open or execute Output Transformation Engine projects without an active system configuration file. Multiple system configuration files can be stored in your file system, but only one file can be set as the active configuration at any given time. For more information on tasks related to managing your system configuration files, see [“Using the Output Transformation System Manager” on page 49](#).

When viewing system configuration files within the file structure in the File Systems window, you can determine the active file by the green arrow on top of the file icon. This can be particularly useful when you have multiple system configuration files stored within the same directory.

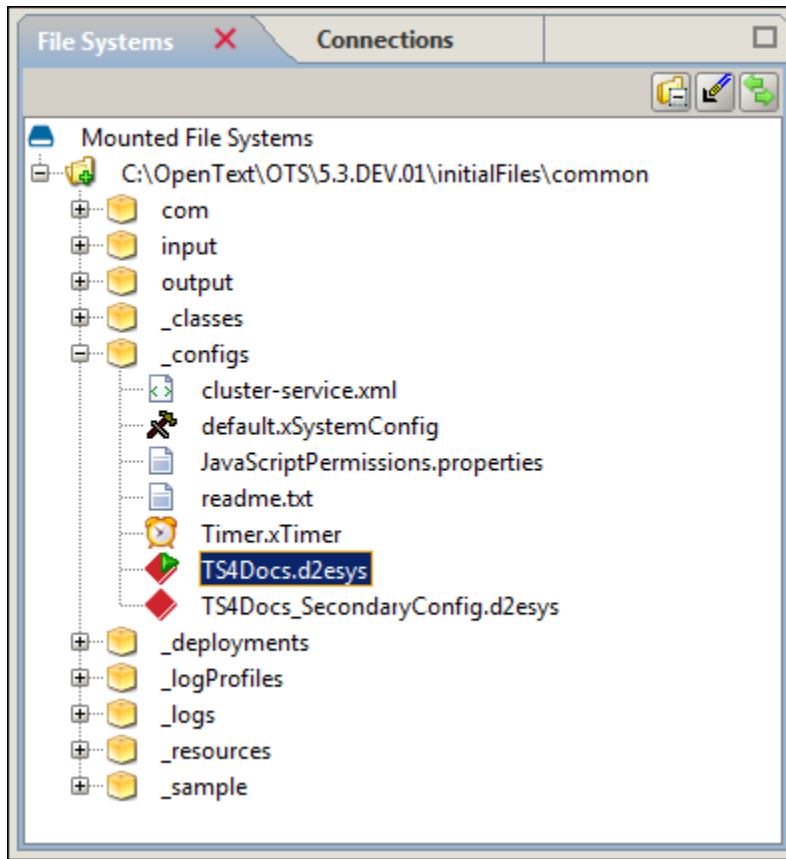


Figure 2-1: File Systems window showing active system configuration file

2.1 System Configuration Parameters

The system configuration file is used to view and edit the system configuration parameters. This group of parameters sets system-wide properties that affect all aspects of your Output Transformation Engine projects. Some of these values can be overridden on a per-project basis.

When you first install Output Transformation Engine, a default set of parameters, which is suitable for most projects, is used. It can be found in `<install_home>\initialFiles\common\_configs` and is named `ote-config.d2esys`. Double-clicking a system configuration file in the File Systems window opens the file in a new tab. For more information about the scope of this file and how to create multiple `.d2esys` files for specific uses, see [“System Configuration” on page 27](#). System-wide configuration modifications are made using this file.

The descriptions in this section explain the default settings for the parameters, as well as information about the parameters and how to best use the settings under normal conditions. Parameter descriptions with extensive secondary documentation contain only the default setting and a link to the secondary documentation.

Additionally, within Output Transformation Designer, when the system configuration file is opened in a tab, basic descriptions of each parameter are displayed in the parameter descriptions window when the parameter is selected.

Job Variables

System job variables are set in the `.d2esys` file. One variable, **d2eOutputPath**, is configured by default. For more information about system variables, see [“System Variables” on page 52](#).

Message Format

This parameter controls the default error message handling mechanism used by Output Transformation Engine at runtime. This parameter provides two options: **XenosFramework** and **Classic**.

The **Classic** option preserves the message reporting behavior and message format of Output Transformation Engine prior to version 3.0, while the **XenosFramework** option uses the new messaging system common to all Output Transformation Server products. This option is provided to provide legacy support to API users who apply custom message handling business logic around the older messaging system.

Use the **XenosFramework** option unless you require this legacy support and have not yet migrated to using the new message system in your API code. See [“Messages” on page 620](#) for more details.

Localized Message File

This parameter is only used when **MessageFormat** is set to **Classic**.

The default path to the message file is `{d2eSysPath}english.d2emsg`. This file contains all of the Output Transformation Engine messages in English. If you localize your system, these messages need to be translated into your system language and then the new file, for example, `french.d2emsg` would be placed in the same location as the original file and this entry would read, `{d2eSysPath}french.d2emsg`. Do not leave this field blank. Do not delete the `english.d2emsg` file.

See also [“Messages” on page 620](#).

Default Report Path

The default report path is `{d2eSysPath}../reports`. If you change the default reports directory, it must be changed here also.

For more information about report and log files, see [“Reports and Log Files” on page 55](#).

Logging

This section describes parameters that manage the system and job message logs produced by Output Transformation Engine. System logs contain messages that are

issued by system components such as the `JobManager` or `LicenseAuthenticator`, while Job logs incorporate messages that are issued by job components like parsers or generators. For more information on parameters that affect logging, see [“Logging Parameters” on page 35](#) and for further info about working with logs, see [“Reports and Log Files” on page 55](#).

Report Manager

The Report Manager section details how all reports for both the system and jobs will be compiled.

The **Sleep Interval** determines the number of milliseconds that reports will wait between intervals of calling the report listeners. The default value of 1000 means that the program will wait 1 second between loops of calling the report listeners (like `ReportWriter`, which writes reports to disk files). Increasing the frequency (choosing a lower value) may reduce overall performance as more CPU resources will be spent on processing the report messages.

Enable System Report is active (selected) by default. This means that a system report file (`system.log`) will be created when Output Transformation Engine is started. If cleared, no system report file will be created. For more information about report files, see [“Reports and Log Files” on page 55](#).

Log Message Options

Designates the format of log messages issued by Output Transformation Engine. The default option is **XenosFramework**, which is the universal format for all OpenText Output Transformation Suite products. If it is set to **Classic**, the engine will issue all existing messages in the original Output Transformation Engine style. For more information about log messages, see [“Messages” on page 620](#).

Input Monitor

Specifies settings to control different types of input monitors. For additional information about input monitors, see [“Input Monitors” on page 590](#).

InputSource

Contains the configuration string for an **InputMonitor** component. This is a comma delimited list where the first token will indicate the InputMonitor type to use, and the remaining tokens are the configuration parameters for that particular monitor. For more information about input monitors, see [“Input Monitors” on page 590](#).

SleepInterval

Indicates the number of milliseconds the **InputMonitor** will wait before it cycles through the sources to collect the external command requests. The default value is 500.

Job Manager

The Job Manager parameters control job handling. For more information about Job Manager parameters, see “[Job Manager Parameters](#)” on page 39 for descriptions.

JobStats

Contains settings related to the collection of job statistics. These metrics differ from the Logging parameters since they are geared more towards performance details, such as the number of documents processed, the length of time the job has been in a queue, and the number of pages generated, and are produced at a set interval for running jobs as opposed to Logging which registers more general messages that are issued by system or job components. Reviewing these job metrics can assist with general troubleshooting and analysis of performance bottlenecks. For more information about the associated parameters, see “[JobStats Parameters](#)” on page 44.

License Authentication

This field is automatically populated during installation. The Output Transformation Engine license is in the <install_home>\settings directory. The path in the system configuration is indicated relative to the {d2eSysPath}. Do not change or remove this entry, or the file to which it points.

Resource Manager

Use the resource manager to configure the overall resource management for the system. These settings can be overridden for individual projects, as required.

ResourceVersionController and RVC Attributes

The engine supports **Resource Version Control (RVC)** and user-written RVC components for AFP transformations. The purpose of the RVC is to allow users to store multiple versions of a resource in their archive system, such as a company logo in an AFP overlay or signature in a font, and still be able to retrieve the correct version of the resource from the cache when processing an archived AFP document.

For more information about the Resource Version Controller, see *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-H-PTE)*.

Cache Options

Specifies options for controlling how cached resources are stored in memory at runtime.

Hard Size

Indicates the maximum number of entries to protect from garbage collection. This parameter is not applicable to cache scheme types of **None**, **Soft**, or **Hard**.

Show Usage

If enabled, system cache statistics will be written to the system log file. For more information about log files, see [“Reports and Log Files” on page 55](#).

Scheme

Indicates the cache scheme arrangement in order of persistence. **None** has no persistence, while **Hard** has full persistence where all resources are kept in the cache for the full life cycle of the process. The soft cache options provide the ability for cached resources to be discarded when memory resources are running low. Therefore, using a soft cache offers a trade off by avoiding **OutOfMemoryErrors**, but potentially using more CPU time because some previously cached resources may need to be reloaded again.



Caution

If any of your projects include an AFP parser with **SkipInlineResOptions** enabled, you must set the Cache scheme to **Hard**.

CacheOverrideMethod

CacheOverrideMethod overrides the cache settings specified inside individual project files for all resources.

ForceJob causes all resources to be stored in the job cache.

ForceSys causes all resources to be stored in the system cache.

This is equivalent to manually setting them in the project (.d2epro) as follows:

```
<filedeflist name="FdAfpfonts">
<filedef directives="{afp}"location="{d2eAppPath}resources
\{0}.fnt"ioHandlerFactoryInstance="" cache="sys"/>
</filedeflist>
<filedeflist name="FdOverlays">
<filedef directives="{afp}"location="{d2eAppPath}resources
\{0}.ovl"ioHandlerFactoryInstance="" cache="sys"/>
</filedeflist>
```

Benefit

These parameters provide a convenient way of toggling the use of system caching. In most scenarios, users will find it better to configure the use of system caching at the global level rather than configuring the cache settings of individual resource types at the project level.

Also, sometimes introducing a project that is configured for batch environments (job-level caching) into a dynamic environment with the resource cache settings unchanged will have a negative impact on memory usage because system caching will be more efficient in most dynamic transform environments. This parameter ensures that an incorrectly configured project will not compromise the system in this manner.

Component Definition

This field is empty by default. When custom components are created by API users or our Professional Services group, the paths to the `components.xml` file should be entered here. This parameter also exists in the project file parameters, which allows you to specify your component definitions for just that project. Components declared here in system configuration file will be available to all projects.

I/O Handler Factory

The I/O Handler Factory enables the connection to custom system I/O handlers. This field is blank by default as the engine has its default way of performing I/O (using the local file system) and this will only be used for custom I/O handlers to access customized input or output streams such as a database. Create the custom I/O handler class and then enter the details in the appropriate fields of the I/O Handler Factory. For more information about I/O handlers and how to create custom I/O handlers, see [“I/O Handlers” on page 112](#) and *OpenText Embedded Output Transformation Engine - Developer’s Guide (VDTOTS-H-PTE)*.

TuningOverride

Not enabled by default, this parameter allows the user to force Output Transformation Engine to disable multithreading within jobs. This is highly recommended when running Output Transformation Engine on z/OS mainframe version 1.8 or higher, as the OS no longer supports multiple threads writing to the same data set. It is also useful in some dynamic transformation environments, where a large number of small concurrent jobs are being executed at the same time (for example, serving user requests via the web).

TuningOverride specifies whether to override the tuning parameters for the parsers and generators in all projects, and to instead use settings for **direct translation**. Direct translation means that all page queues are disabled and only one page is handled at any given time throughout the project flow. This puts the entire round trip per transaction in one thread.

This is equivalent to manually setting them in the project (`.d2epro.j`) as follows:

```
<tuning translatorCount="0"
iOThreadCount="0"
inputQueueSize="0"
outputQueueSize="0">
</tuning>
```

This applies to the parser and all generators in the project.

Benefit

By default, projects are configured to use one translator thread and one I/O thread for each parser and generator, resulting in a potential of four or more additional threads used per job. Parser and generator queues are set up to better handle batch, as opposed to dynamic loads.

In a dynamic environment, experience has shown that direct translation provides the best performance results. This is desirable because you want to have as few a number of threads as possible in order to have more threads available for more transactions. With `TuningOverride` enabled, project developers do not have to worry about changing these settings manually.

I/O Cache

The parameters in this section define the system level IOCACHE options. The IOCACHE provides a buffering system when reading or writing I/O streams to help optimize the number of reads or writes performed on the I/O stream. The main purpose is to prevent numerous reads/writes of single bytes or just a few bytes at a time. In those cases, processing such small amounts of data can dramatically impact performance because the overhead of performing an action on the stream is high and can be reduced by reducing the number of times a read or write action is preformed.

The IOCACHE is enabled by default.

The **cache size** can be any number between 0 and 200 Kb. The default value is 32 Kb. If you are observing I/O bottlenecks in your application, you can experiment with these values to find the optimal value for your environment.

If **Overrides is disabled**, system IOCACHE settings will apply to all jobs with no other IOCACHE settings.

If **Overrides is enabled**, the system IOCACHE set here will override any other job level or FileDef IOCACHE settings configured. Job level IOCACHE settings will override FileDef IOCACHE settings.

IntegrationOptions

Contains settings related to integrating Output Transformation Engine with other products.

InfoArchive

These options are related to the download process for Output Transformation for InfoArchive.

UseMultiThreadedCache

Designates whether to cache AFP resources using multiple threads when viewing or downloading the document. If set to true, AFP resources are cached using multiple threads and then these cached resources can be used later to improve performance. If set to false, the AFP resources are cached using a single thread.

2.1.1 Logging Parameters

This section describes parameters that control the types of logging that are used as well as organizational details such as the level of details the logs contain and where they are stored.



Note: Users who interface with log messages through the API to direct the messages to their own logging facility can still use the filtering options; the message log filters apply to both the default log files produced by the application and API subscribers, even if they have disabled the job logs in favor of their own logging service.

2.1.1.1 SystemLog

This set of parameters control system message logs.

- **SystemLogLevel.** Denotes the level of detail to include in the system message log. You can select from Minimal or Verbose. The **Verbose** logging level includes startup, licensing, and system configuration information. It also includes details about each job that runs, listing the start, end, and whether or not the job was successful. The **Minimal** logging level only contains licensing information, the job start/end times, and whether the job was successful.
- **FdSystemReport.** Specifies the location of the system log.
- **MessageFilter.** Contains settings related to the message filter for system logs.
 - **Enabled.** Indicates whether the message filter is enabled.
 - **FilterEntry.** Contains settings related to an instance of the message filter.
 - **MessageType.** Specifies the types of messages to be filtered. You can choose from **Any**, **None**, **Debug**, **Info**, **Warning**, **Error**, **Fatal**, **Config**, **Summary**, or **Exception**. The **None** option effectively disables the filter, which can be useful during testing runs when you want to see all messages and not lose your current filter configuration.
 - **CategoryName.** Designates which messages to filter based on the message source, such as “AfpParser”, using a regular expression. For example, in the message header “Info DocumentsCreated(PdfGenerator)”, the category name is “PdfGenerator”. If you want to filter any categories, use “.*”.
 - **MessageName.** Designates which messages to filter based on the name of the message, such as “ResourceNotLoaded”, using a regular expression. For example, in the message header “Info DocumentsCreated(PdfGenerator)”, the message name is “DocumentsCreated”. If you want to filter any message names, use “.*”.
 - **MessageText.** Designates which messages to filter based on the text in the message, using a regular expression. For example, setting the value as “.*Arial-Bold.*” filters any messages that contain the text “Arial-Bold” anywhere in the body of the message.

- **DebugLogging.** Contains settings related to the logging of debugging information to the system log. The debug logging can provide ongoing information about running jobs, contents of the resource cache, and the current state and stack traces of active threads.
 - **DebugStatusEnabled.** Indicates whether a debug status report is sent to the system log at an ongoing interval, as defined by `DebugStatusInterval`. It is highly recommended debug status reports be disabled in production environments since they cause the report log to increase in size indefinitely as reports continue to be written to the log, even when the application is idle. By default, this is disabled.
 - **DebugStatusInterval.** Specifies the number of seconds in between sending status reports to the system log. To use this parameter, you must also enable `DebugStatusEnabled`.
 - **IncludeAllThreads.** Designates that stack traces for all threads in the JVM are included in the debug status report. If disabled, then only application-related threads are included in the report.

2.1.1.2 JobLog

This set of parameters control job message logs:

- **Enabled.** Indicates whether job message logs are enabled.
- **JobLogLevel.** Indicates the level of detail to include in the system message log. You can select from `Minimal` or `Verbose`. The **Verbose** logging level includes startup, licensing, and system configuration information. It also includes details about each job that runs, listing the start/end times, full configuration settings, run details, and whether or not the job was successful. The **Minimal** logging level contains only licensing information, the job start/end times, whether the job was successful, and important messages such as if there were errors while running the job.
- **FdJobReports.** Specifies the location of the job log. Your value must contain the {0} replacement variable, which is replaced with the job report file name during runs.
- **MessageFilter.** Contains settings related to the message filter for job logs.
 - **Enabled.** Indicates whether the message filter is enabled.
 - **FilterEntry.** Contains settings related to an instance of the message filter.
 - **MessageType.** Specifies the types of messages to be filtered. You can choose from **Any**, **None**, **Debug**, **Info**, **Warning**, **Error**, **Fatal**, **Config**, **Summary**, or **Exception**. The `None` option effectively disables the filter, which can be useful during testing runs when you want to see all messages and not lose your current filter configuration.
 - **CategoryName.** Designates which messages to filter based on the message source, such as “`AfpParser`”, using a regular expression. For example, in the message header “`Info DocumentsCreated(PdfGenerator)`”, the

category name is "PdfGenerator". If you want to filter any categories, use ".".

- **MessageName.** Designates which messages to filter based on the name of the message, such as "ResourceNotLoaded", using a regular expression. For example, in the message header "Info DocumentsCreated(PdfGenerator)", the message name is "DocumentsCreated". If you want to filter any message names, use ".".
- **MessageText.** Designates which messages to filter based on the text in the message, using a regular expression. For example, setting the value as ".*Arial-Bold.*" filters any messages that contain the text "Arial-Bold" anywhere in the body of the message.
- **UseNewMfctFontMappingMessageFormat.** Signifies that the newer MFCT font mapping message format is used. In most circumstances, the regular MFCT font mapping message format is suitable as it provides data on the MFCT and automatic font mappings, while the newer messaging format provides much more detailed information on the MFCT. While most users can use this parameter, it is of particular interest to API users who have implemented business logic around the original message format to trigger a workflow when unmapped fonts are encountered.

2.1.1.3 Filtering Messages

The logging parameters in system configuration allow the suppression of messages from job and system level reports, which can be helpful for troubleshooting issues. To filter out messages, you must create a **FilterEntry** parameter set and ensure that the **MessageType**, **CategoryName**, **MessageName**, and **MessageText** child parameters are all specified, otherwise no messages will be filtered. See the following examples for sample FilterEntry schemes.

Example 1

If you want to filter out all license expiry messages like these as shown in the job report:

Warning	LicenseToExpire(Job)	Your license will expire in 20 day(s) (on December 31, 2018)

Then in your system configuration file, set up a **FilterEntry** parameter set with the following options:

```
<?xml version="1.0" encoding="UTF-8" ?>
<VisionService po-class="com.xenos.d2e.fwserver.parm.VisionServiceParm"
  localizedMessageFile="{d2eSysPath}../resources/english.d2emsg">
  .....
  <logging>
    <systemLog>
      .....
    </systemLog>
    <jobLog>
      .....
      <messageFilter enabled="true">
        <filterEntry messageType="Warning"
          categoryName="Job"
          messageName="LicenseToExpire"
```

```
messageText=".*license.*" />
</messageFilter>
</jobLog>
</VisionService>
```

This example returns all messages with the **Warning** type, from the **Job** category, with a **LicenseToExpire** message name, and containing the term **license** within the message's text. You could optionally broaden the range of returned messages by setting `messageText` to `".*"` which returns all messages that fulfill the requirements set out by the other three parameters.

Example 2

Message filters can be set on either the system or job log level and can even sort out messages from both at the same time. The following sample configuration filters all error messages from the system log and all warning messages from the job log.

```
<?xml version="1.0" encoding="UTF-8" ?>
<VisionService po-class="com.xenos.d2e.fwserver.parm.VisionServiceParm"
  localizedMessageFile="{d2eSysPath}../resources/english.d2emsg">
  .....
  <logging>
    <systemLog>
      .....
      <messageFilter enabled="true">
        <filterEntry messageType="Error"
          categoryName=".*"
          messageName=".*"
          messageText=".*" />
      </messageFilter>
    </systemLog>
    <jobLog>
      .....
      <messageFilter enabled="true">
        <filterEntry messageType="Warning"
          categoryName=".*"
          messageName=".*"
          messageText=".*" />
      </messageFilter>
    </jobLog>
  </logging>
  .....
</VisionService>
```

Related Links

- [“SystemLog” on page 35](#)
- [“JobLog” on page 36](#)

2.1.2 Job Manager Parameters

This section describes parameters that help control the overall way job submission is managed on your system. For more information about JobManager parameters, see [“System Configuration” on page 27](#).

2.1.2.1 DefaultReportListener

This is the file where all system processes will be logged. The default is: `{d2eSysPath}../reports/system.log`. By default, this file will be overwritten and started anew each time Output Transformation Engine is restarted. If you need to retain a record of the information contained within the system log file, rename `system.log` when Output Transformation Engine is not running. A new `system.log` file will be created on startup. Alternatively, you can include the `{0}` replacement variable syntax (e.g. `{d2eSysPath}../reports/system-{0}.log`) to have the system fill in a timestamp in the filename. This will result in a new system log being created each time the engine is started.

2.1.2.2 MaxConcurrentJobs

Typically, the most significant impact on performance is determined by the number of concurrent jobs.

We suggest you test with N concurrent jobs, where N is the number of CPUs in your server. Most production machines will easily be able to handle more and we will be pleased to assist you at that time in further performance tuning.

2.1.2.3 CheckQueuedJobs

CheckQueuedJobs defines options to control job submission based on the number of queued jobs. If enabled, new jobs will not be queued while the number of queued jobs is greater than or equal to the value of **MaxQueuedJobs**. Under heavy load, jobs will be intentionally failed if the queue is full. This option is disabled by default.

2.1.2.3.1 Benefit

If **CheckQueuedJobs** is not enabled (default), Output Transformation Engine will queue all submitted jobs. Memory is at a premium in large-scale implementations where printstreams are passed entirely in-memory (typically from a large ECM document or image archive), since each queued job requires memory for the input that is therefore not available for processing. If the system is being overloaded with job submissions, there will be a possibility of an `OutOfMemoryError` occurring, which could result in a system crash.

Limiting the number of queued jobs allows the job submitter application to quickly reclaim used memory if, based on the configuration settings, Output Transformation Engine decides that it will not be able to handle the job given the current load.

2.1.2.3.2 Recommended Configuration

The maximum number of queued jobs selected depends on your situation and the trade-off that you are willing to make between system stability and the number of failed jobs that will not produce output.

Taking this into account, set **MaxQueuedJobs** relative to the maximum number of concurrent jobs, set by the **MaxConcurrentJobs** parameter. There are three configuration scenarios:

- **MaxQueuedJobs** > **MaxConcurrentJobs**

The more **MaxQueuedJobs** exceeds **MaxConcurrentJobs**, there will be a greater chance of a job being available when a running job completes.

A reasonable configuration in a Dynamic Retrieval scenario would be **MaxQueuedJobs**=200 and **MaxConcurrentJobs**=10. The application that is submitting the job to then engine would have to include logic to deal with rejected jobs in the event of a spike in load.

- **MaxQueuedJobs** = **MaxConcurrentJobs**

There should always be a job available when a job completes. This is likely to be a better configuration for an environment with larger jobs where memory is a concern.

However, in a high load and small job size environment, a number of jobs would not get a chance to be processed.

- **MaxQueuedJobs** < **MaxConcurrentJobs**

In this situation, there may never be a job available when a running job completes. This would be a less than optimal configuration.

2.1.2.3.3 Notes

In realistic situations where jobs are submitted on average over a longer period of time, this parameter will work as designed. For performance reasons, Output Transformation Engine's job submission methods are only loosely synchronized. Due to this design, a large number of jobs being submitted concurrently may not be accurately controlled by the **MaxQueuedJobs** parameter.

If the user chooses to synchronize their own job submission code, and the submitter waits for the job to complete, multiple jobs will not be submitted at once and will instead be processed sequentially, negatively impacting performance, whereas if the submitter chooses not to wait for the job to complete, this leaves the handling of the output stream entirely with Output Transformation Engine.

2.1.2.4 Verbose Job Report

The Verbose Job Report parameter applies to both the system log file (`system.log`) and the job reports.

If enabled, the system log will include startup, licensing and system configuration information. It will also include entries for each job that runs, log listing the start, end and success/failure of the job. The job report will include full configuration and run details of the job.

If disabled, the only system startup information the system log will contain is licensing information. In addition, each job will write entries in the system log listing the start, end and success/failure of the job. For each job that runs, a job report file will be created but will only contain important entries, such as if there were errors while running the job.

When enabling or disabling the verbose job report, Output Transformation Engine must be restarted before the change takes effect.

2.1.2.5 Enable Job Report

If enabled, a new job report will be created every time a job runs. The report will be written to the folder specified at Default Report Path and will be named as `<job name><time stamp>.log`. In addition to a job report being generated, there will also be an entry made in the system log (`system.log`) listing the start, end and success/failure of the job.



Note: The time stamp is a 13-digit number representing the number of milliseconds since January 1, 1970. This is the standard convention for the value returned by Java's `System.currentTimeMillis()` method.

If disabled, no job report will be created for any job that is run. There will, however, be an entry in the system log listing the start, end and success/failure of the job.

When enabling or disabling the job report, you must restart the application before the change takes effect.

2.1.2.6 Default Job Report Listener

This field indicates the path to where the job reports will be written. The default, `"r;default"`, points to `<install_home>\initialFiles\common\_sample\d2eVision\reports`. When a job is run, a new job report will be created in this folder.

If you want to change the location of the Job Report files, enter a fully qualified path followed by `{0}.rpt`. For example, `C:\OpenText\OutputTransformation\3.2.0\initialFiles\common\_sample\d2eVision\reports\jobReports\{0}.rpt`.

When changing the location of the job report, Output Transformation Engine must be restarted before the change takes effect.

2.1.2.7 Application Config Path

This legacy parameter can be used to specify a directory containing project files which can be referenced by name (without a file extension) during job submission, rather than using the full path to the project file. Such project files should use the deprecated `<projectName>.xml` naming convention rather than the `<projectName>.d2eproj` convention currently used. This feature can be used in a variety of job submission methods, such as InputMonitors, d2eMain command line, and JobManager API simply by specifying the `<projectName>` string in place of the full project filename.



Note: Users are now encouraged to use the newer convention of keeping the project file in a directory tree with the associated input files and resources so they can take advantage of the `{d2eAppPath}` variable to keep their application folder portable.

2.1.2.8 FdlcuTables

This parameter specifies the location of the ICU encoding tables which are primarily used in bidirectional AFP – Unicode codepoint mappings. In most cases you will not have to change this parameter unless you have specific requirements in organizing your resource files on your production servers.

2.1.2.9 Console Messages

The default setting is disabled. During the configuration and testing phase, console messages will not be required as the setup is being done in the Output Transformation Designer GUI. You may wish to enable console messages for deployed servers in production when no GUI is available to capture messages sent to stdout.

2.1.2.10 CheckHeapUsage

CheckHeapUsage defines options to control job submission based on Java memory heap usage. If enabled, **MaxHeapPercent** defines the maximum percentage of the heap to use before failing new jobs (percentage of HeapSize or value specified by `-Xmx`). New jobs will not be queued while the percentage of the heap used is greater than or equal to this value. This helps prevent jobs that will not run soon from using all available memory.

2.1.2.10.1 Benefit

By default (i.e., if **CheckHeapUsage** is not enabled) Output Transformation Engine uses as much of the heap as it requires. Throttling the amount of memory that Output Transformation Engine can use (out of the total available) helps to prevent the JVM from getting into situations where the maximum heap size is reached and a subsequent garbage collection is unable to reclaim enough memory to continue, resulting in OutOfMemory errors. This prevents a single job from using all the available memory.

2.1.2.10.2 Recommended Configuration

The configuration chosen depends on your situation and the trade-off that is willing to be made between system stability and the number of failed jobs that will not produce output.



Note: For performance reasons, heap usage is checked only upon job submission.

2.1.2.11 JVM Garbage Collection

If enabled, a forced full system garbage collection (`system.gc()`) will run once the number of jobs specified at Job Count Interval has completed running. The Job Count Interval can be any number between 1 and 100. If

InvokeGcOnJobCompletion is not enabled (default), the number will be ignored and no forced garbage collection will not run.

If disabled (default), regular system garbage collection will run based on settings in the JVM configuration.

If you are running several very large jobs back-to-back, consider using this feature to free memory on a regular basis.

2.1.2.12 JobProfiling

The **JobProfiling** parameter can be used to help profile job heap usage and set an optimal configuration for your environment. If enabled, a number of jobs are run (set by the `JobCount` attribute) and heap usage statistics are recorded at a given interval (set by the `sleepInterval` attribute).

The resulting statistics are factored into a calculation that determines and sets the number of jobs Output Transformation Engine thinks can be handled concurrently:

- $\text{Max Heap Size} * \text{Max Heap Percent} - \text{Post-profiling Engine Size} = \text{Amount Available}$
- $\text{Amount Available} / \text{Maximum Job Size} = \text{Maximum Concurrent Jobs}$

This calculation takes into consideration the maximum heap percent and size set by the **“CheckHeapUsage”** on page 42 parameter, if enabled (for example, a lower value for **MaxHeapPercent** means a lower value for **MaxConcurrentJobs**).

A warning is logged to the system report when **MaxConcurrentJobs** has been overridden. (For more information about **MaxConcurrentJobs**, see [“MaxConcurrentJobs” on page 39.](#))

2.1.2.12.1 Benefit

This parameter provides a very useful diagnostic tool in determining the load the engine can handle in terms of concurrent jobs. Furthermore, it will override the existing setting to ensure that an optimal value is used.

2.1.3 JobStats Parameters

This section describes parameters that control job statistics collection such as the file’s appearance and allows you to configure which specific metrics to include in the report.

2.1.3.1 Enabled

Indicates whether the statistics collection mechanism is active. The collection of job statistics requires additional CPU and memory resources so statistics collection should be disabled when not required.

2.1.3.2 CreateStatsFile

Indicates that a statistics file, in CSV format, is created. If you only require Java API access to the job stats, then you can disable this.

2.1.3.3 FdStatsFile

Specifies the directory where the statistics file is written to.

2.1.3.4 EnableFileHeader

Denotes that a file header is added to the top of the statistics file, which lists environment information such as the software version.

2.1.3.5 EnableColumnHeader

Denotes that a CSV column header is added to the statistic file.

2.1.3.6 DecimalFormat

Designates the formatting of decimal numbers. For more information, see the `java.text.DecimalFormat` class.

2.1.3.7 DateTimeFormat

Designates the formatting of the date and time. For more information, see the `java.text.SimpleDateFormat` class.

2.1.3.8 MillisecondFormat

Designates the formatting of the number of milliseconds. For more information, see `java.util.Formatter` class.

2.1.3.9 MegabyteFormat

Designates the formatting of the number of megabytes. For more information, see `java.text.DecimalFormat` class.

2.1.3.10 EnabledStats

Contains a selection of specific statistic properties to log. Be aware that in order to capture some of these statistics, one or more options under **JobManager** > **JobMetrics** also need to be enabled; any prerequisite settings for a given statistic is listed along with its description.

- **JobID.** Indicates the job ID, which is an incrementing value starting from 1. The JobID is also part of the job name.
- **JobName.** Denotes the job name, which is presented with a `<ProjectName>_<JobID>_<Timestamp>` format.
- **StatusCode.** Indicates the job status code, which is a character that represents the status of the job when the job was terminated. The API specification declares the possible values for status codes as **H** (Held), **Q** (Queued), **R** (Running), **E** (Errored), **F** (Finished), **S** (Suspended), **N** (New Job), **A** (Aborted), or **D** (Denied). However, the expected values for statistics purposes only include those stats which apply to jobs that have terminated: **E**, **F**, **A**, or **D**.
- **CompletionCode.** Denotes the job completion or return codes. A job completion code can be one of the following values: **0** (Job successful), **4** (Job completed with warnings), **8** (Job completed with errors), **12** (Job completed with runtime exceptions), **16** (Job failed during initialization), **20** (Job aborted by request), **24** (Job denied, engine is shutting down), or **28** (Job denied, not enough heap/thread resources).
- **AppPath.** Indicates the application path, which is the path location of the directory containing the project file. This is equivalent to the `{d2eAppPath}` job variable.
- **ProjectName.** Indicates the project file name. This is equivalent to the `{d2eProjectName}` job variable.

- **ProjectFile.** Specifies the project file name including the full path location.
- **InputFileName.** Denotes the input file name. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **OutputFileName.** Indicates the output file name. Be aware that when there are multiple output files, only one of them is collected for job stats. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **DocsProcessed.** Specifies the number of documents processed. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **DocsPerMinute.** Specifies the number of documents processed per minute. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **InputFilePageCount.** Denotes the page count of the input file, which can be useful to compare to **PagesProcessed** to ensure that all known pages were processed. This value is only available for certain input formats, such as PDF, which provide easy access to the total number of pages in a document. Other stream-based formats, such as AFP, require full processing to calculate this value. Consequently, this statistic is not available for any stream-based format. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **StartPage.** Indicates the value of the start page parameter as defined by the project. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **StopPage.** Indicates the value of the stop page parameter as defined by the project. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **PagesParsed.** Specifies the total number of pages parsed by the parser during the job. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **PagesGenerated.** Specifies the total number of pages generated by the generators during the job. If your project contains multiple generators, this is the sum of pages generated by all generators. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **PagesProcessed.** Denotes the total number of pages processed. Note that this figure is based on the highest number of pages processed by any single component in the job. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **PagesPerMinute.** Denotes the number of pages processed per minute. Like the **PagesProcessed** statistic, this figure is based on the highest number of pages processed by any single component in the job. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **QueuedTimeStart.** Identifies the time when the job was added to the queue as it waits to run.
- **QueuedTimeEnd.** Identifies the time when the job was removed from the wait queue and enters the running state.
- **StartTime.** Identifies the time when the job started to run.
- **EndTime.** Identifies the time when the job finished running.

- **TotalQueuedTimeMS.** Identifies the total time, in milliseconds, the job remained in the wait queue.
- **TotalClockTimeMS.** Identifies the total time, in milliseconds, the job was running.
- **TotalCpuTimeMS.** Identifies the total consumed CPU time, in milliseconds, when the job was running. As a prerequisite, **CollectXdcComponentMetrics** must be enabled.
- **TotalUserTimeMS.** Identifies the total consumed CPU time in user time, in milliseconds, when the job was running. As a prerequisite, **CollectXdcComponentMetrics** and **CollectThreadDataMetrics** must be enabled.
- **TotalBlockedTimeMS.** Identifies the total blocked time, in milliseconds, when the job was running. As a prerequisite, **CollectXdcComponentMetrics** and **CollectThreadDataMetrics** must be enabled.
- **TotalWaitTimeMS.** Identifies the total wait time, in milliseconds, when the job was running. As a prerequisite, **CollectXdcComponentMetrics** and **CollectThreadDataMetrics** must be enabled.
- **MaxHeapSizeMB.** Indicates the maximum heap size, in megabytes, of the JVM.
- **TotalMemoryAllocMB.** Indicates the size, in megabytes, of the total memory allocated (consumed) by the job. As a prerequisite, **CollectXdcComponentMetrics**, **CollectThreadDataMetrics**, and **CollectThreadMemoryMetrics** must be enabled.
- **StartHeapSizeMB.** Indicates the active heap size, in megabytes, at the start of the job.
- **StartHeapFreeMB.** Indicates the amount of free, unused heap memory, in megabytes, available at the start of the job.
- **EndHeapSizeMB.** Indicates the active heap size, in megabytes, at the end of the job.
- **EndHeapFreeMB.** Indicates the amount of free, unused heap memory, in megabytes, available at the end of the job.
- **JobCacheAdditions.** Denotes the number of resources that were loaded successfully to the job cache by the job.
- **JobCacheFailures.** Denotes the number of resources that failed to be loaded to the job cache by the job.
- **SysCacheAdditions.** Denotes the number of resources that were loaded successfully to the system cache by the job.
- **SysCacheFailures.** Denotes the number of resources that failed to be loaded to the system cache by the job.
- **SysCacheRetrievals.** Denotes the number of resources that were retrieved by the job from the system cache. If there are unexpectedly no retrievals, it may be because the requested resources were not loaded successfully to the system cache in the first place.

- **StartQueuedJobCount.** Specifies the number of jobs that were waiting in the queue when the current job started running.
- **EndQueuedJobCount.** Specifies the number of jobs that were waiting in the queue when the current job ended.
- **StartRunningJobCount.** Specifies the number of jobs that were running when the current job started running.
- **EndRunningJobCount.** Specifies the number of jobs that were running when the current job finished running.
- **TotalWarnings.** Indicates the number of warnings encountered during the job.
- **TotalErrors.** Indicates the number of errors encountered during the job.
- **MostSevereMessage.** Designates the most severe message logged during the job.

2.1.4 Job Override Parameters

This section describes parameters that override the configuration specific to a job, such as what is set in a project. These parameters can be configured in the system configuration file.

TuningOverride

Definition

TuningOverride is a boolean parameter which specifies whether to globally override the tuning parameters for the parsers and generators in all projects, and to instead use settings for direct translation. Direct translation means that only one page is handled at any given time throughout the project flow. This puts the entire round trip per transaction in one thread.

Setting TuningOverride = True is equivalent to manually setting the thread counts to zero in all projects (.d2epro) as follows:

```
<tuning translatorCount="0"  
iOThreadCount="0"  
inputQueueSize="0"  
outputQueueSize="0">  
</tuning>
```

This applies to the parser and all generators in the project.

Benefit

By default, Output Transformation Engine projects are configured to use one translator thread and one I/O thread for each parser and generator, resulting in a potential of four or more additional threads used per job. Parser and generator queues are set up to better handle batch, as opposed to dynamic loads.

In a dynamic environment, experience has shown that direct translation provides the best performance results. This is desirable because you want to have as few a number of threads as possible in order to have more threads available for more

transactions. With **TuningOverride** enabled, project developers do not have to worry about changing these settings manually.

CacheOverrideMethod

Definition

CacheOverrideMethod (located under the **ResourceManager** settings) overrides the cache settings for all resources. **ForceJob** causes all resources to be stored in the job cache, while **ForceSys** causes all resources to be stored in the system cache.

This is equivalent to manually setting them in the project (d2epro) as follows:

```
<filedeflist name="FdAfpfonts">
  <filedef directives="{afp}"
  location="{d2eAppPath}resources\{0}.fnt"
  ioHandlerFactoryInstance=""
  cache="sys" />
</filedeflist>
<filedeflist name="FdOverlays">
  <filedef directives="{afp}"
  location="{d2eAppPath}resources\{0}.ovl"
  ioHandlerFactoryInstance=""
  cache="sys" />
</filedeflist>
```

Benefit

Introducing a project that is configured for batch environments into a dynamic environment with the resource cache settings unchanged will have a negative impact on memory usage. This parameter ensures that an incorrectly configured project will not compromise the system in this manner.

2.2 Using the Output Transformation System Manager

The **Output Transformation System Manager** can help you oversee your system configuration files within . To launch this feature, in navigate to **Environment > Output Transformation System Manager** and it opens in a new window.

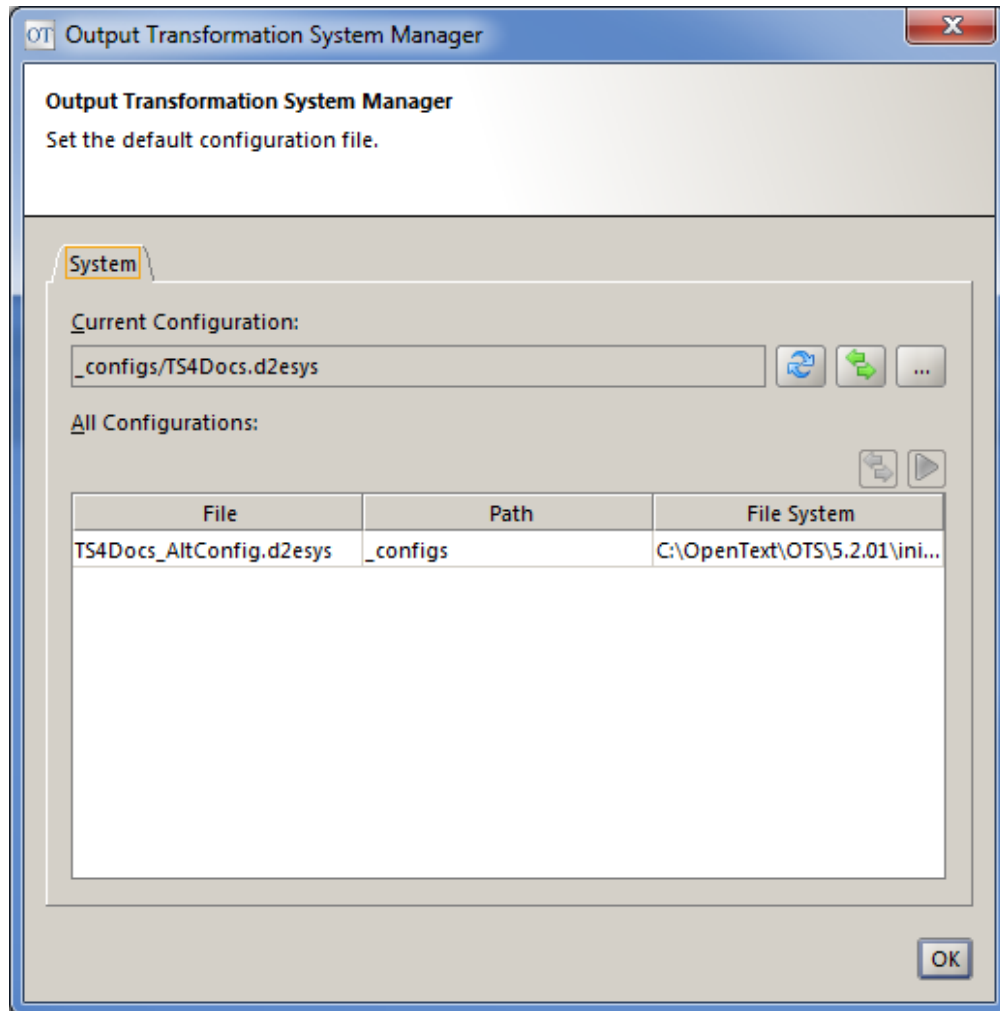


Figure 2-2: Output Transformation System Manager window

2.2.1 Opening a System Configuration File for Viewing or Editing

To open a system configuration file (*.d2esys), first you must launch the **System Manager** and then:

- If you want to open your current system configuration file for viewing or editing, click the ellipsis button, , beside the **Current Configuration** field.
- If you want to open any other system configuration file in your file system, locate the file in the **All Configurations** table, right-click it and from the context menu that appears, choose **Open**. (Only system configuration files stored in any of your mounted file systems appear in this list; if you cannot locate your system configuration file in the table, then you need to check the file path location or mount the file system.)

The system configuration file opens in its own tab in the **Development** window.

2.2.2 Finding the Location of a System Configuration File

If you currently have a system configuration file open in the Development window, but do not know where the file is located in your file system, you can trace its position using the System Manager.

If you want to know the location of your current system configuration file:

- Hover the cursor over the file name in the **Current Configuration** field and the full file path location appears.
- Select the **Show in the file system** button, , and the file is highlighted in the **File Systems** tab while the system tree expands to the file's location.

If you want to know the location of a non-current system configuration file:

- Locate the system configuration file in the **All Configurations** table and view its location in the **Path** and **File System** columns.
- Select a system configuration file in the **All Configurations** table and select the **Show in the file system** button, . The file is highlighted in the File Systems tab while the system tree expands to the file's location.

2.2.3 Setting the Active System Configuration File

The active system configuration file applies to all projects. While you can have multiple system configuration files saved in your file system, only one can be operational at a time. You can determine which file is being used as the active system configuration by checking the file name in the **Current Configuration** field of the System Manager.



Note: You must already have the other system configuration file stored somewhere in your mounted file system.

To set a new system configuration file as active:

1. In , launch the System Manager by going to **Environment > System Manager**.
The **System Manager** opens.
2. In the **All Configurations** table, select the system configuration file you want to set to the active system configuration file.
3. With the file highlighted, click **Execute**, .
The selected system configuration file is established as active and the file name appears in the **Current Configuration** field.
4. Click **OK** to close the System Manager.

In a production environment, a separate `.d2esys` file can be configured to run jobs in batch mode or to configure an input monitor to automatically run certain projects

when the matching input files are found in a defined directory. (For more information about input monitors, see [“Input Monitors” on page 590](#).) To run in batch mode or with the Input Monitors, Java 1.5.x is required.

Configure this project in and associate it with a specific .d2esys file. Subsequently, when the engine is packaged, it will include the associated .d2esys file.

```
C:\OpenText\OTS\<version>\initialFiles\common\_configs\watchdir.d2esys
```

2.2.4 Restarting the System

To restart the Output Transformation Engine system:

1. In Output Transformation Designer, launch the System Manager by going to **Environment > System Manager**.

The **System Manager** opens.

2. Click **Restart the system**, .

The system is restarted.



Note: Any open Page View tabs will be closed during system restart.

2.3 System Variables

The **JobVariables** parameter, found in the **System Configuration tab**, allows you to define job variables. These variables can be used in setting file paths. (For more information about using variables to set file paths, see [“File Paths” on page 100](#).)

To add or remove variables:

- **Add.** Right-click on the JobVariables parameter and select **Add**. You may then enter a new name and value for that variable.
- **Remove.** Right-click on the variable you wish to remove, and select **Remove**.



Note: For information about using system variables in command line operation, see [“Optional Command Line Arguments” on page 595](#).

The following is a list of system variables whose names are reserved:

Variable	Description
d2eOutputPath	This variable is used to group all output file definitions for a job into a common, controlled location, and would typically be set to a location on the user's local drive. If not set, the engine will default it to " ". Example: {d2eOutputPath} = c:\xpro\output\

d2eSysPath	This variable stores the full path to the directory where the System Configuration file is located. It allows users to develop their application using relative paths for all configurations, which ensures that the application is portable and can be easily zipped or copied to another location without configuration changes. Example: {d2eSysPath} = c:\OpenText\OutputTransformation\myProject\config\
d2eAppPath	This is the location of the application (where the project files are kept). This is set automatically by the engine. Users are encouraged to base file definitions for resources held in the application subdirectory off of this variable.
JobInputFile	The name of the input file.
d2eInputFileName	The file name of the input file set in the parser's <i><FdInput></i> parameter. For more information about file paths, see "File Paths" on page 100.
d2eInputFilePath	The file path of the input file set in the parser's <i><FdInput></i> parameter. For more information about file paths, see "File Paths" on page 100.
d2eInputFileExtension	The file extension of the input file set in the parser's <i><FdInput></i> parameter. For more information about file paths, see "File Paths" on page 100.
d2eProjectName	This is the name of the .d2eproproj file being run. The project name is set when the project is created, but may be changed by right-clicking on the project name in the File Systems Window and selecting Rename. For more information on naming your projects, see "Project Properties Tab" on page 87.
d2eJobId	This is the unique job ID (Project Name concatenated with the system provided time stamp).

Date and time variables	Date and time variables can be used to add and format a date and/or time stamp on a report or filename. For example, you can express an output file name that includes the full date and time in the format <code>dd_mm_yyyy@hh-mm-ss.xxx_tim.pdf</code> as the file name by entering the following as the variable value: <pre>{d2eAppPath}--output\{0}--mit_{sysDay}_{sysMonth}_{sysYear}@{sysHour}--{sysMinute}--{sysSecond}.{sysMillisecond}_tim.pdf</pre> See below for valid date and time variables:
sysMonth	This variable uses the month value (1 – 12) from the OS.
sysDay	This variable uses the date value (1 – 31) from the OS.
sysYear	This variable uses the four-digit year value from the OS.
sysHour	This variable uses the hour value (00 – 23) from the OS.
sysMinute	This variable uses the minute value (00 – 59) from the OS.
sysSecond	This variable uses the second value (00 – 59) from the OS.
sysMillisecond	This variable uses the millisecond value (000 – 999) from the OS.
d2eClassPath	This variable sets the path to the Java class files used by . Example: <code>{d2eclasspath} = {classpath}input\afp\my-input.afp</code>
Others	Users can define any variable they wish from the System Configuration tab.



Note: Users should not use the d2e prefix on their custom variables, as these are reserved by the application.

2.4 Reports and Log Files

The system creates several log files that can be used to investigate job failures, configuration issues, or even system errors. While some log files contain information also found in another log file, the scope and purpose of each file is unique. Messages in the log files are issued by system components or by job components.

- **System** log messages are messages that are issued by system components (such as JobManager or LicenseAuthenticator).
- **Job** log messages are messages that are issued by job components (such as parsers and generators).

2.4.1 System Log File

```
<install_home>\initialfiles\common\_sample\d2eVision\reports\system.log
```

The system log file is created at system startup and contains detailed information about the system configuration, startup processes, and all projects that have run. On startup, the existing system.log file is replaced with a new file of the same name. If you need to keep the old file for tracking or troubleshooting purposes, rename the file while it is not running. On startup, a new system.log file will be created and the renamed file will remain. You cannot rename this file while it is running.

If you need to change the name or location of this file, the information must be updated in the system configuration (.d2esys) file.

If there is no system log file, or only an old system log file with no new information being written, ensure that the **EnableSystemReport** check box in the Report Manager section of the system configuration file is selected.

For more information about the system configuration file, see [“Report Manager” on page 30](#).

This log file may also be referred to as the **system report file**, or the **engine log**.

2.4.2 Debug Log

```
<install_home>\initialFiles\common\_logs\Debug\all_debug.textreport
```

The debug log is written to with every system message and job message that is issued. This log contains all the information that is also in the system log, reports log and the error log. Since the system log is overwritten with a new file each time it is started, past configuration information is lost on each startup. The debug log, however, is perpetual, preserving the past data, which may make this an invaluable resource in troubleshooting job or system errors.

This log file was formerly known as the **diagnostic log**.

2.4.3 Error Log

```
<install_home>\initialFiles\common\_logs>ErrorFatal\ErrorsOnly.textreport
```

The error log is written to only in the event of a job error. It contains only the error details of errored jobs, not the entire details of the job. In essence, the error log starts where the system log and the report log end.

2.4.4 Log File

```
<Drive>:\Documents and Settings\<user>\.xenosstudio\vision\<version>\<identifier>\logs\application.log
```

The log file contains an entry for each startup since the file was created. Each entry provides details of the installed Java JVM and Operating System, the number of days until license expiry, and any errors encountered during startup.

By default when the file reaches 1 megabyte, it will be renamed to `studio.1` and a new log file will begin. To change the maximum size or the maximum number of files in a log file:

1. Choose **Preferences > Logging** in .
2. Choose the **Logging** tab.
3. Modify **MaxSize** or **MaxFiles**.

2.4.5 Installer Log

```
<install_home>\logs\installer.log
```

The installer log is created when is installed. It contains details of file locations, including details of the Java JVM. If any errors or warnings occurred during the installation, they will be detailed in the installer log.

2.4.6 Report Log

```
<install_home>\initialFiles\common\_sample\d2eVision\reports\<project name><#>.log
```

A report log is created for each job that runs. It contains messages about the job detailing all job settings and run details. It does not, however, contain error details of failed jobs. If a job fails, it will only list that the job errored and a completion code. See the error log and debug log for error details.

The number that the project name is appended with in naming the log file is a 13-digit timestamp representing milliseconds since January 1, 1970. Free conversion tools are available on the Internet if you need to convert it to a standard date/time format.

This log file may be referred to as the **job report**, or the **project report log**.

If you see no report logs in the location listed above, ensure that the **EnableJobReport** check box is selected in the JobManager section of the system configuration file.

If there is a report log, but it contains no data, ensure that the **VerboseJobReport** check box is selected in the JobManager section of the system configuration file.

2.4.7 Managing Log Files

2.4.7.1 Report Manager

The Report Manager section of the system configuration (.d2esys) file details how reports will be compiled.

The **Sleep Interval** determines the wait time between intervals of calling the report listeners. The default value of 1000 means that the program will wait 1 second between loops of calling the report listeners. Increasing the frequency may reduce overall performance as more CPU resources will be used to process report messages.

Enable System Report is active (selected) by default. If enabled, the system report file will be created. If cleared, no system report file will be created, however system events and messages will still be written to the debug log.

2.4.7.2 Log Message Options

The **Log Message Options** section of the system configuration (.d2esys) file determines how log messages are displayed in the output pane of when a job is run.

The default message format, **XenosFramework**, is the message format used throughout . This message format provides all the message details within the entry, making the log easy to read and interpret.

See “[Messages](#)” on page 620 for details.

2.5 Dynamically Renaming Reports

There are instances where multiple system logs are required, such as when working with several simultaneous sessions. In these cases, each process will need to write to a different log file. There are a few ways to achieve this arrangement, but the theory is the same; the filename specifying the location of the system log file will contain a variable that is set uniquely for each running instance of Output Transformation Engine.

To dynamically name multiple logs by using a command line argument to provide the instance variable:

1. In the .d2esys file, locate the **defaultReportListener** parameter and replace the default log location with the file path of the new log location. For example:

```
<jobManager defaultReportListener="file:c:\OpenText\OutputTransformation\report\
{instanceName}--system.log" verboseJobReport="false">
```

You must indicate with **instanceName**, the desired variable name to uniquely identify of the system report file.

2. The **jobVariable** attribute in the command line must also be changed to rename the log file. For example:

```
- jobvariable=instanceName="Instance One"
```

The log file generated is called **instance one-system.log**. (For more information on working with the command line, see [“Command Line Operation” on page 592](#).) Additionally, you can allow the engine to use a date and time stamp in the system log name. This is beneficial when you are working with multiple log files as it ensures that existing log files are not overwritten. To allow the engine to restart without overwriting the existing system.log file:

3. In the **defaultReportListener** parameter, you must specify {0} after the instance name. For example:

```
<jobManager defaultReportListener="file:c:\OpenText\
OutputTransformation\report\{instanceName}-{0}-system.log"
verboseJobReport="false">
```

Thus, a **jobVariable** command line of `- jobvariable=instanceName="instance1"` in conjunction with usage of {0} in the **defaultReportListener** parameter returns a system log with a name: `instance1-25.Jul.2008_14.50.0218-system.log`.

2.6 Tuning for Performance

The following sections discuss various aspects of performance configuration and tuning.

2.6.1 Factors Affecting Performance

To achieve its maximum potential, several variables must be carefully considered during implementation planning:

The complexity and mix of the input print streams

- Input streams with low graphic content such as line data will transform faster than streams with a lot of vector or image content.
- Size of the printstream, i.e. number of pages.
- The mix of various levels of complexity within the input printstreams.

How and where the print stream resources are stored

- Resources stored locally can be more efficiently cached than resources stored inline or in ACIF resource files.

Configuring Output Transformation Engine to maximize the use of correlated fonts and minimize rasterization or imaging of resources

- Rasterization is time consuming and memory intensive. Generally rasterization should not be considered an easier way of font correlation.
- Proper font correlation is a key to excellent fidelity and fast transaction times.

Specific application requirements

- Specialized document handling, content additions or complex indexing requirements will impact performance.

Systems architecture

- The number of servers required to process the expected volume.
- The number of CPUs on each server.
- The amount of memory available to the JVM.
- Efficient configuration of the Output Transformation Engine resource cache based on the number and usage of resources across applications.
- Ensuring the network backbone has enough bandwidth to support high traffic if the inputs, outputs and resources are to be streamed across the network.

System load

- The number of jobs over a given time period.
- The number of jobs processing at the same time and the memory used to parse the input and generate the output.
- The number of jobs waiting to be processed and memory used by the waiting job.

External factors

- Memory allocated by processes outside the control of Output Transformation Engine, such as streamed input/output.

2.6.2 Performance Tuning Assumptions

Understanding the following assumptions and technical facts is key to Output Transformation Engine performance tuning.

- Output Transformation Engine's persistent engine starts up only once.

Jobs are submitted to the running engine.

Project files can be cached so that they are loaded only once.

- The application's engine is optimized to use the maximum amount of CPU available.

Thread prioritization cannot be reliably controlled across platforms.

Job processing can be throttled by setting the `MaxConcurrentJobs` parameter to less than the number of CPUs available. This will lower the potential performance of the system, as not all the available CPUs will be utilized. For more information about throttling jobs, see [“MaxConcurrentJobs” on page 39](#)

- System files are opened once at engine startup.

Applications can be configured so that project files are parsed once for the first transaction and reused for subsequent transactions.

Input resource file definitions can also be configured to parse resources once for the first transaction and are reused for subsequent transactions.

- The Output Transformation Engine logging subsystem can be configured to write verbose, basic or no log information at all.

The API exposes capability for custom logging.

2.6.3 Recommended Project Configurations

Batch Environment

The default way to configure a Output Transformation Engine project is to set up each parser and generator with multiple translator threads. This is very effective in a batch-processing model where large input files are transformed.

The default cache settings are also tuned to a batch environment at the job level. Job level cache means there is a separate cache for each transaction.

For infrequent large, long-running jobs where the engine is started and stopped, for example a monthly batch process, a job cache will be faster. Job level caches reduce unnecessary synchronizations on the cache lookups where there is only one job running at a time.

For many small running jobs, a system level cache is recommended because each job shares the same cache instead of each job having its own cache.

Dynamic Environment

The second way to configure a Output Transformation Engine project is to configure the parsers and generators to use a single thread supplied by the calling application. By setting the `TranslatorCount` and `IOThreadCount` parameters to 0, you can put the entire round trip per transaction in one thread. This is desirable because you want to have as few a number of threads as possible in order to have more threads available for more transactions.

This is most effective in dynamic environments where multiple small transactions are required.

For dynamic environments, job level caches are inefficient because each job gets its own cache. This increases the amount of memory required to run the engine.

Setting the cache level to `sys` eliminates the duplicate resource caches and drastically reduces the memory required, allowing all transactions to access the same cached resources.

For more information about planning memory requirements, see [“Dynamic Environment Memory Requirements Planning”](#) on page 62.

Threading and Queuing Model

The diagram below illustrates the threading and queuing architecture used within Output Transformation Engine. In batch transforms, which involve documents with many pages, configuring the tuning parameters within the parser and generators will achieve optimal performance.

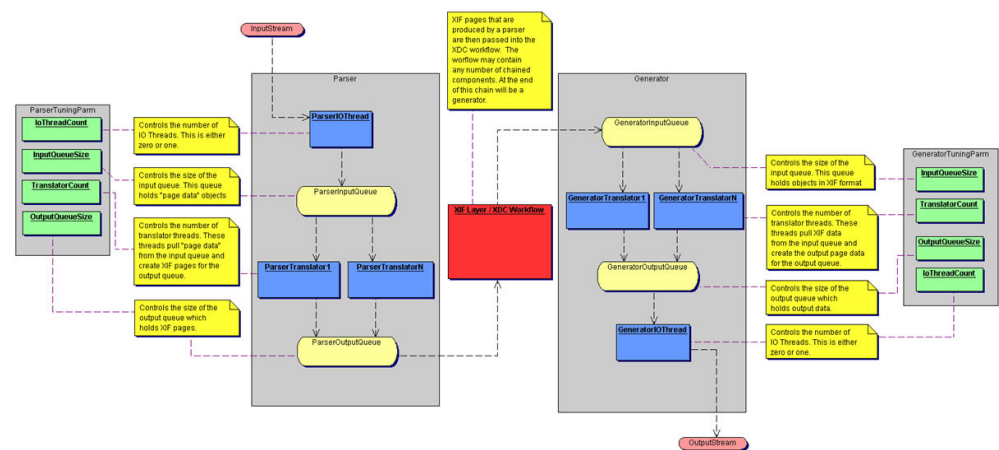


Figure 2-3: Threading and Queuing Model for Output Transformation Engine

2.6.4 Dynamic Environment Memory Requirements Planning

This section demonstrates how project configuration impacts memory usage in a dynamic on-the-fly transform and retrieval environment. The default sample project included with Output Transformation Engine is set up for a batch environment. Without some configuration changes it will not run efficiently in a dynamic environment. Through awareness of each major contributor to memory usage, a better understanding of the impact of improperly configured projects can be used as a planning tool for integrating with Output Transformation Engine in dynamic environments.

When a job is submitted to the engine, a limited number of jobs are immediately sent for transformation. The rest are queued and wait for one of the submitted jobs to complete. As soon as a job completes, one of the queued jobs is removed from the queue and transformed.

The maximum number of concurrent jobs allowed to run before queuing takes place is controlled by the `MaxConcurrentJobs` parameter in the **JobManager** settings of the `sys.d2esys` parameter file. This parameter defaults to 5.

If you use an improper configuration, memory usage will be substantially higher.

As the number of jobs increases, the impact of duplicate resources also increases as each job has its own cache. Changing the resources to a system level cache results in reducing the amount of memory by the amount of memory taken up by each job cache.

Translation is the act of parsing a single input page into the XIF Intermediate Format and also refers to the act of generating a single page of output from the XIF. Multiple translator threads will each translate a page at a time.

Reducing the number of translator threads further reduces the amount of memory required by the system. Translator threads greater than zero are only effective in batch environments where there is a single large input stream to parse.

Dynamic environments typically do not benefit by having multiple threads translating because these threads are better utilized elsewhere transacting multiple jobs.

There is always at least one thread translating. When the translator count in the parameter file is set to zero, the thread is provided by the job, or the application that submitted the job. The reduction in the number of translation threads reduces the amount of memory required. Once total memory is reached, the system stabilizes and works within the available memory.

Another variable of concern is the number of queued jobs. This is a function of several variables that make up the architecture of the environment running Output Transformation Engine, which all contribute to the potential number of queued jobs, including:

- Total potential users
- Peak access times
- Number of transaction servers
- Number of CPUs per server
- Speed of the CPUs
- Size of the document transformed

The best way to reduce the number of queued jobs is to make the unit of work (transaction time) as small as possible given the performance requirements. This allows more jobs to be transformed in a smaller amount of time, allowing more jobs to be transacted.

It is also important to note that the number of queued jobs represents concurrent transactions and not concurrent users on the system. Typically the number of concurrent users is several orders of magnitude greater than the number of concurrent transactions ongoing for that number of users.

Again, there are many variables outside the control of Output Transformation Engine that contribute to this number.

Output Transformation Engine can also enforce a limit to the number of queued jobs. For more information about the use of specific parameters, see [“Job Manager Parameters” on page 39](#).

2.6.4.1 Other Factors Influencing Memory Usage

If there are many different applications to transact by a single engine, this can be problematic. Ideally the engine would run most efficiently if all resources were kept in the cache. This is not practical if there are so many resources that they consume most of the memory in the system.

For this reason the system cache can be tuned to one of five different options which balance the need for cache persistence to efficient memory usage. The cache can be turned off completely (not recommended for performance reasons), configured to use a Soft Cache or configured to use a Hard Cache (default configuration).

The Soft Cache is the best balance between persistence and memory usage for systems that use many different applications. The Soft Cache will only hold onto a predefined number of resources. The rest of the resources are held as Soft References that will be eligible for garbage collection when memory becomes low.



Note: Alternatively, consider the `CacheOverrideMethod` parameter. For more information about this parameter, see [“CacheOverrideMethod” on page 32](#)

2.6.5 Interpreting the FdStats File

The **FdStats** file is an optional statistics file whose location is defined in the **FdStats** parameter. When this parameter is present, the parser or generator will write out the tuning statistics to the FdStats file in the specified location. These statistics may be useful for performance tuning.



Note: The FdStats parameter is available for all Output Transformation Engine parsers and generators.

Example FdStats File (generator):

```
2 38 4015 GeneratorInputQueue[addWaits=23,removeWaits=2,capacity=12]
GeneratorOutputQueue[addWaits=0,removeWaits=22,capacity=12]
```

In the above example:

- There is a total thread count of 2.
- There were a total of 38 objects processed. These objects can include pages, XFT events, or begin/end documents.
- The elapsed time, indicating the total lifetime of the generator, is 4015 milliseconds.
- The remaining values are the **queue statistics** for both queues (input and output) in the generator. See the table below for a general interpretation of queue statistics.

Troubleshooting Using Queue Statistics

You can use the queue statistics generated in the FdStats file to troubleshoot performance issues.

Queue	High addWaits Value	Low addWaits Value	High removeWaits Value	Low removeWaits Value
Parser Input Queue	Parser I/O thread is running efficiently, but has to wait because parser translators are not keeping up with the workload. May indicate a bottleneck downstream.	Parser I/O thread is not waiting to send data downstream. Could be running smoothly or may indicate I/O bottleneck reading the input stream.	Could indicate an I/O bottleneck on the input stream. You may want to consider higher performance alternatives.	Parser I/O thread is running optimally in the current configuration.

Parser Output Queue	Parser I/O thread is often blocked from reading more input because the queue is full. Bottleneck is downstream in the other XDC components and/or the generator.	Generator is keeping up with removing XifPages from this queue.	Generator is waiting for work to do. Bottleneck appears to be in the parser.	Generator is not waiting for the parser. Could be running smoothly, or there could be a bottleneck in the parser.
Generator Input Queue	Bottleneck appears somewhere in the generator (either in translators or output I/O stream).	Transform is either running optimally, or there is a bottleneck in the parser/XDC components.	Generator translator threads are waiting for work to do. You can either lower the generator translator threads, or address the bottleneck in the parser/upstream XDC components.	Generator translator threads are keeping up with the workload. If waits are high in other areas, you may need to increase the generator translator count.
Generator Output Queue	Could indicate an I/O bottleneck writing to the output stream if removeWaits value is also low. You may want to consider higher performance alternatives.	No output I/O bottleneck detected.	Generator writer thread is waiting for work to do. Bottleneck is upstream in the parser, XDC components and/or generator translators.	Writer thread may not be keeping up with the workload. Could indicate I/O bottleneck in the output stream. You may want to consider higher performance alternatives.

addWaits and removeWaits Values

`addWaits` and `removeWaits` values refer to components waiting to add or remove an object to or from the queue when the queue is currently full. (A waiting thread cannot perform any further processing, preventing optimal performance.)

If the `addWaits` and `removeWaits` values appear well balanced, then optimal performance has likely been achieved. If some wait values are higher compared to others, tuning should be performed to address the bottleneck.

Capacity

The `capacity` value refers to the queue size, which is configured in the tuning parameters. The default value is typically set to 10. Increasing this value may help

reduce wait times, but larger queue sizes mean larger numbers of complex objects can be stored in memory, resulting in `OutOfMemoryErrors` if the value is set too high.



Note: When trying to balance the parser and generator configuration, it is most often recommended to adjust the thread counts instead of the queue size, especially when heap memory is limited. If there is sufficient room to increase the heap size from current levels, then increasing the queue sizes may be beneficial. However, keep in mind that the larger the queue size, the larger the memory footprint that will be required when the queue is full of objects.

2.7 Output Transformation Engine Configuration from the Classpath

In most situations, elements like project, message, or system configuration files need to be tailored to a particular project, but by employing the classpath method, a set of the necessary license or configuration files can be stored in one location for usage by multiple instances. Multiple projects are able to share these configuration resources making the process much more manageable. Generally, the motivation for users to set a classpath argument is their interest in packaging the application in a WAR, JAR, or EAR file for deployment to a web-based application server or as part of a more comprehensive standalone Java application.

A few key points to note:

- The types of files referenced by the relative paths in the `.d2esys` file are limited to read files only; this method will not work when referencing files that are written to.
- The classpath variable can either point directly to the configuration files, or the files may be enclosed within a JAR file.
- Currently, this method of loading configuration files is not applicable to the loading of any debug utility config files. You must still define the debug configuration files through the usual methods of either setting the absolute file path using the API by calling `D2eProcess.getInstance().setDebugConfig()` or through the command line when calling `com.xenos.d2e.d2eMain`.
- You must ensure that all of your classpath location paths are valid, otherwise a Java exception will be thrown.

The following topics are available on configuration of the classpath variable:

- [“Configuring the System Configuration File with a Classpath Argument” on page 67](#)
- [“Modifying the Batch File with a Classpath Argument” on page 67](#)
- [“Implementing a License File from a Classpath Argument” on page 68](#)

2.7.1 Configuring the System Configuration File with a Classpath Argument

You must arrange your system configuration files to utilize the classpath variable. To configure your .d2esys file with classpath arguments:

1. Locate your system configuration file and open it with your preferred script editor. The default system configuration file is named `runjob.d2esys` and can be located in the `engine\<version>\config` folder of your installation directory.

You must indicate which of the properties will utilize the classpath variable. Pinpoint the parameters that you want to set a classpath variable for, and change its value to your classpath argument. For instance, the following script excerpt indicates that messages will be read from the file path location outlined in your classpath variable:

```
<system localizedmessagefile="{CLASSPATH}english.d2emsg">
```



Tip: Multiple classpaths can be added to the value of a property by using a semicolon to separate each one.

2. **Save** your script.

The file location assigned to your classpath variable is used when the specific property is called.

For a use case involving the employment of classpath variables in practice including script excerpt examples, see [“Implementing a License File from a Classpath Argument”](#) on page 68.

2.7.2 Modifying the Batch File with a Classpath Argument

To modify the batch file to use a classpath argument:

1. Locate the batch file used to call your instance of to run and open it with your preferred script editor. The default batch file used to launch is named `rund2e.bat` and can be located in the `engine\<version>\tools` folder in the installation directory. If you have packaged the engine, the batch file can be found inside the package file within the same `tools` subfolder.

You must create the name that references your classpath argument and indicate the value that your classpath argument will reference. At a logical point in your batch file script, enter the argument while mimicking the following pattern:

```
SET <CLASSPATH_NAME>=<FILE_LOCATION>
```

Therefore, the argument `SET D2E0BJECTPATH="..\d2esys.jar"` signals to the system that whenever the `D2E0BJECTPATH` classpath argument is called, the JAR file at the specified location will be referenced.

2. **Save** your script.

The classpath variable can now be accessed by multiple projects when `{CLASSPATH}` is called upon in the project’s configuration files.

For a use case involving the utilization of classpath variables in practice including script excerpt examples, see [“Implementing a License File from a Classpath Argument” on page 68](#).

2.7.3 Implementing a License File from a Classpath Argument

The following procedure is for illustration purposes only. It explains all the steps involved in fully executing a license file with a classpath argument for use on multiple machines. Nevertheless, you can employ the general framework of this process and apply it toward other circumstances to utilize classpath arguments. For more detailed information on setting the classpath, see [“Configuring the System Configuration File with a Classpath Argument” on page 67](#) and [“Modifying the Batch File with a Classpath Argument” on page 67](#).

To implement a license from the classpath:

1. Place all necessary set up and configuration files, whether standalone or packaged in a JAR, in your chosen destination folder.
2. Locate the script file used to call to run (in most instances, `rund2e.bat`), and open the file in your preferred script editor for editing.
3. Set the absolute file path of the setup and configuration files as the value of **D2ETEMPCLASSPATH**. The program references this value to detect the location of the configuration files.

In the sample script excerpt from the `rund2e.bat` file shown, the configuration files are located in a file named `xenos-classpath.jar`, with its full path location laid out:

```
@ECHO This will run d2eVision passing up to 9 args into d2eMain.@ECHO .
@ECHO If no args are given, d2eMain will still be called causing
@ECHO its command line syntax to be shown, and a "pause" is issued.
@ECHO .
@ECHO For example: rund2e "-config=system.d2esys" "-app=test.d2eproject"
@ECHO .
@ECHO OFF
SET D2ETEMPENGINEPATH=".."
SET D2ETEMPCLASSPATH=.;c:\xenos\d2e-vision\engine\V3.0.40\classpath-config\xenos-
classpath.jar
```

The content of the `xenos-classpath.jar` file varies from customer to customer since it is dependent upon your own configuration setup, but will resemble the example below:

```
xenos-classpath.jar
+ english.d2emsg
+ license/
license.d2elic
+ icutables/
<*.xml ICU files>
```

4. **Save** your script.
5. Locate your system configuration (`.d2esys`) file, and open the file in your preferred Java editor for editing.

6. The value of **licenseauthentication licensefile** points to the location of your license file. You must enter {CLASSPATH}, along with any further file path information. The program is then able to call the license file with the classpath variable since you previously resolved the file path that it references.

In the sample script from the `runjob.d2esys` file shown, the location of the license file is established by the `licensefile` parameter. The value uses the {CLASSPATH} variable as part of its file path location to signify that it should be read from the classpath's set location:

```
<?xml version="1.0" encoding="UTF-8"?><system
localizedmessagefile="{CLASSPATH}english.d2emsg"
defaultreportpath="{d2eSysPath}../../../../reports">
<jobvariable name="d2eOutputPath" value="{d2eSysPath}../../../../output/"
ispath="true"/>
<reportmanager sleepinterval="1000"/>
<jobmanager defaultreportlistener="file:{d2eSysPath}../../../../reports/system.log"
sleepinterval="500" statusreportinterval="10" maxthreads="100"
defaultjobreportlistener="default"
applicationconfigpath="{d2eSysPath}../../../../apps" consolemessages="true">
<filedeflist name="FdIcuTables">
<filedef directives="CLASSPATH" location="icutables/{0}.xml"
ioHandlerFactoryInstance="null" cache="sys"/>
</filedeflist>
</jobmanager>
<licenseauthentication licensefile="{CLASSPATH}settings/license.txt"/>
<resource manager resourcefile="" />
</system>
```

7. **Save** the file.
8. Run your projects normally, either through the console or Output Transformation Designer.

The configuration and license files will automatically be shared among all projects.

2.8 System Properties

In the `<install_home>\settings` folder, there are several properties files that users can modify in order to establish system-wide settings. Setting these values incorrectly can have adverse effects on your installation, so you must be certain when editing these files. Many of these files contain some brief comments on their functionality and additional information can be found in this section.

2.8.1 Setting a Separate Work Directory

By default, your work directory is kept inside the installation directory. This is suitable for most situations when there is only one user, but administrators with multiple users may want to provide each user with their own workspace area to minimize the possibility of conflicts with project settings. If you want to keep your installation directory separate from your working directory, then you can follow these steps:



Note: Your base installation folder and subsequent references to it will be represented by the `<OTS_home>` variable while the work area folder is represented by the `<OTS_workarea>` variable.

1. Navigate to the <OTS_home>\settings\ folder and open the SetOtsWorkarea.bat file for editing.
2. Either delete or comment out the following two lines:

```
set OTS_WORKAREA=  
goto END
```
3. To define the new working directory, locate the set OTS_WORKAREA=%UserProfile%\ots\workarea_16.1 line and replace the existing value with the desired working directory. If the work area folder you specify does not already exist, it will automatically be created. (By default, %UserProfile% is a system variable that indicates where a user's home directory is located, which is set to the C:\Users directory for Windows users.)

When you are finished, **save** your work.

4. Navigate to the <OTS_home>\startup folder and launch the application by running the startDesigner.bat file.

While Output Transformation Designer is starting, the new work area is initialized. Subsequent launches of the application by this user can use either the startDesigner.bat or startDesigner.exe files under <OTS_home>\startup or the user's own <OTS_workarea>\startup folder since the work area is switched automatically for the particular user. Also, bear in mind that any users running the program will hold full permissions to the working directory.

2.9 Running JavaScript

Consider the following guidelines when using JavaScript in your projects.

2.9.1 Maintaining JavaScript Security

Vulnerabilities in software can result in compromises in your security posture, which can circumvent the safety measures set up to protect your organization from both internal and external threats. While the probability of a breach is highly unlikely, it is important to be aware that the risk still exists and the damage caused by a malicious script can be critical, due to either the crashing of a server or the loss of sensitive data. The software supports JavaScript and unauthorized users may attempt to gain access by executing malicious JavaScript so it is imperative to secure your system to safeguard against these types of attacks.

Within all products in the Output Transformation Suite, a JavaScriptPermissions.properties file is used to authorize which parameters can be used to run @js and @jsFile JavaScript functions. This file is stored in your <install_home>\BaseRepositories\<appserver>\<instance_name>\common_configs directory. By default, no parameters are allowed to run @js functions while all parameters are allowed to run @jsFile functions.



Note: The @jsFile function runs a JavaScript located on the server's file system and is wholly controlled by administrators, thus, it does not pose the same threat as @js functions.

When opening a project that includes a property containing some JavaScript, the **Security** dialog appears and prompts you to decide how to handle execution of the script since permission must be explicitly granted in order for the script to run. The following options are available:

- **Keep blocking.** Continues to block the script and does not allow it to run.
- **Allow during this session.** Permits the script to run for the duration of the current session. Upon shutting down your VM session and then later reconnecting to the VM, you will be asked again whether the script is allowed to run.
- **Always allow.** Permanently allows the script to run. Users can go into the `JavaScriptPermissions.properties` file to manually add any scripts that are permitted to run, however, selecting the Always allow option performs the same function.

2.9.2 JavaScript library compatibility

This application provides JavaScript customization for dynamic configuration parameter evaluation and custom component logic. Beginning with Java version 15, JVM installations discontinued providing a JavaScript engine by default. Starting with Output Transformation Engine version 22.4, the Nashorn JavaScript library that was previously included with the JVM is now shipped as a third-party JAR library with the product. Including the Nashorn library provides seamless support for users who upgrade their product to version 22.4 and their Java to version 17 or later.

2.10 Packaged Embedded Output Transformation Engine File

A prepackaged Output Transformation Engine file known as the Embedded Output Transformation Engine ZIP file is available to assist with manual deployments to another machine that does not have a display (such as machines running Linux or IBM z/OS USS), or to a production machine where a GUI is not required. The engine's file name is `EOTE-<version>.zip` and is shipped in the `subproducts` directory along with your installation package.

For detailed instructions on deploying the engine file, refer to *OpenText Embedded Output Transformation Engine - Standalone Installation Guide (VDTOTS-ITE)*.

Chapter 3

Getting Started with Projects

This chapter contains topics to help you get started on working with Projects.

3.1 Getting Started with Projects and Components

The following topics related to working with projects and components are available to help you get started:

- [“Project Components” on page 74](#)
- [“Adding Components to a Project” on page 75](#)
- [“Viewing Component Resources” on page 76](#)
- [“Editing a Component” on page 77](#)
- [“Removing a Component” on page 77](#)
- [“Removing a Component Connection” on page 78](#)

The following list represents a typical set of tasks that are performed when transforming a print or data stream into high-fidelity output. For every project you want to run, you should do the following:

- Create a new project by launching the New Project Wizard .
- Add the Parsers and Generators you require into the Flow window and connect them to represent the project flow.
- Review the following by working your way down the Project window, looking at the information that has been assembled into the project:
 - Parameter settings
 - Resources paths
 - Resource Collection (if used)
 - Field Definitions in XFT
 - Tool components setup (if used)
 - Font correlation (MFCT)
 - IO Handlers (if used)

3.1.1 Fine Tuning a Project

To fine tune a project:

1. Run the d2e Vision Project and check the output. For more information about process, see [“Processing Overview” on page 587](#).
2. Make changes where necessary.
3. Run the project again.
4. Continue to make adjustments, if necessary, to get the output you require.

3.1.2 Project Components

Project components are predefined accompaniments that handle specific operations in your project. The following Process Flow components are available:

	Description
	“Best Viewed Rotation Component” on page 237
	“Constant Print Component” on page 238
	“Document Breaking” on page 282
	“Field Modifier Component” on page 265
	“Index Writer Component” on page 272
	“Indexer Component” on page 281
	“Large Print Component” on page 293
	“Page Markup Component” on page 295
	“Replace Text Component” on page 324
	“Script Component” on page 326
	“Simple Branch Component” on page 327
	“Skip Page Component” on page 327
	“Table of Contents Component” on page 330
	“URL Link Component” on page 335
	“XFT Field Technology Component” on page 340
	“XFT Writer Component” on page 498

For more information on each component and its use, please see its respective section in this user guide.

In addition to these predefined components, custom components can be created by our Professional Services department or by a Java programmer to perform tasks outside of the base product capabilities. For information about creating and adding

custom components, see *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-H-PTE)*.

3.1.3 Creating a New Project

Create a new project using the **New Component Wizard**:

1. After starting, launch the New Component Wizard in one of the following ways:
 - Select **File > New** from the menu.
 - Select  from the toolbar.
 - Select **Tools > Wizard** from the toolbar.
 - Right-click a mounted file system from the **File Systems** tab, and select **New > Component**.
2. Choose a file type, name, and location for the project.

Note: The New Component Wizard has a maximum of three steps depending on the options you chose in step two.
3. Select a project to use as a template or create a project from scratch.
4. Choose a parser, at least one component, and generator(s) for your project.
5. Decide whether to launch wizards once the New Component Wizard closes that define your chosen parser and generator(s).



Note: You may also define a project's properties using the **Properties** tab.



Note: For more information on the New Component Wizard, see *OpenText Embedded Output Transformation Engine - User Guide (VDTOTS-H-UTE)*.

3.1.4 Adding Components to a Project

Components are the building blocks of your project; you must add the components and position them in the desired sequence to construct a project or process flow. Connections must be established between these components for them to pass data to one another upon running the job according to your mapping of the project.

When connecting components, you must keep in mind that producer components can only be linked to consumer or other producer components, while consumer components can only be linked to a producer component. Producers are classified as any element that manipulates some kind of data prior to making it into XIF format, such as a parser or a document control component. Consumer components are those that receive XIF data to create output, like generators. When the criteria for a connection are not satisfied, the connection cannot be made and you are notified of the reason by way of one of the following error messages:

- **Component is not a producer.** Occurs if you try to connect a consumer to another consumer.
- **Component is not a consumer.** Occurs if you try to connect a producer to another producer.
- **Component cannot have any more producers.** Occurs when you attempt to connect a producer to a component that already has the maximum number of producers linked.

To create a project in the Development window:

1. In the **Palette**, locate the component you want to add. The components are separated into categories so you must click either **Parser**, **Component**, or **Generator** to expose the associated components.
2. Select the component you want to use and then drag it to the **Development** window. When adding subsequent components, it is recommended you position them roughly in the order you want them to run to make drawing connections later easier to assemble.
3. Repeat steps 1 and 2 until you have added all the components you need for your project and when you are finished, proceed to the next step.
4. Next, you must link the components together. In the Development window, hover the pointer over the check box  a component icon and the pointer changes into the **connection pointer**, .
5. Click the check box and drag the pointer to the component you want to connect to.
6. Repeat steps 4 and 5 until you have linked all the components in the Development window in the order you want your project to work.
7. When you have drawn connections between all your components in the Development window, **save** your work.

With your process flow set up, you are ready set up the configuration of each component. More information about the properties for each component is available in their respective chapters.

3.1.5 Viewing Component Resources

A component's resources are all the different types of files that support the parser or generator in processing the input file. The values contained in resources are the paths and file names for, amongst others, fonts, forms or overlays, images, and setup files like PageDef or FormDef files.

To view a parser or generator's resources, in either the Output Transformation Engine tab or Development window, right-click on the component you want to view and from the context menu that appears, select **View Resources**.



Note: Not all components will have viewable resources. If this is the case, the component will not have **View Resources** available in the context menu.

3.1.6 Editing a Component

Editing components is relatively quick and easy with the use of the included wizards. The wizards can help you apply changes to the component's parameters, or, while on the component's edit dialog, you can click on the Properties tab to see the list of all parameters related to the component that you can edit manually.



Note: Some tool components do not have wizards for them, but include Designers to assist in their configuration. All parsers and generators include wizards.

There are two ways to edit a component:

- In either the Output Transformation Engine tab or Development Window, right-click the component you want to edit and from the context menu, select **Open**.
- Double-click the component in the Development window.

Both of these methods will display the component's respective edit dialog. The **Wizard** tab is shown by default, if available for the selected component. To manually edit the parameters, select the **Properties** tab.

3.1.7 Removing a Component

To remove a component from your project:

1. In the **Development** window, click on the component you want to remove.
2. Next, either press **Delete** on your keyboard or right-click on the selected component and from the context menu that appears, select **Remove**.
3. You must confirm the deletion of the component. Select **Yes** if you are sure you want the component removed, or **No** if you would like to cancel the operation.



Note: Deletion of a component cannot be undone. You can re-add the component from the palette, however, you must also form all component connections and reconfigure all the settings again.

3.1.8 Removing a Component Connection

There are two ways to remove a component connection in the Development window:

- Click the connection line you want to remove and when handles appear on the connection line, press **Delete** on your keyboard.

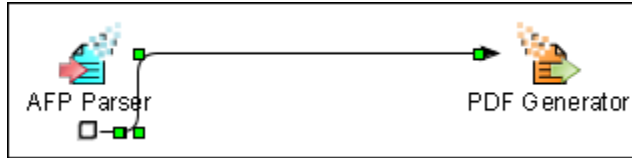


Figure 3-1: Handles on the selected connection line

- Right-click the connection line you want to remove and from the context menu that appears, select **Remove**.

3.1.9 Special characters

The At sign (@) is commonly used when setting values for parameters in your projects. Starting with version 22.4.12 of Output Transformation Engine, support for the tilde (~) in the application's syntax as an escape character with respect to @function() calls was improved. Refer to the following table to determine the behavior when these special characters are used in conjunction with each other.

Usage of At sign (@) and/or Tilde (~)	Expected behavior
At sign (@) is followed by any text and open parenthesis For example, @concat("string1", "string2")	Represents a function call. The example results in string1string2.
A tilde (~) in front of At sign (@) For example, trigger text for URL could be [- a - zA - Z0 - 9 ~ @ : % . _ \ + ~ # =] { 1 , 256 } [a - zA - Z0 - 9 ()] { 1 , 6 } \ b . com . *	The tilde (~) is used as an escape character. Without the tilde (~), the example text would be treated as a function.
If your text string starts with the At sign (@) and is followed by text and open parenthesis, then it is a function unless the tilde (~) is used as an escape character	For example, ~@concat("string1", "string2") would not be a function call since the tilde (~) is used, which results in @concat("string1", "string2").
A tilde (~) cannot be followed without any spaces by the results of a function call	If you require the tilde (~) to function as a normal character, you can express it similar to @concat("~", @concat("string1", "string2")), which results in ~string1string2.

At sign (@) not followed by open parenthesis For example, email@company.com	Represents literal usage of the at sign (@)

3.2 Projects Overview

The unit of work for is a **project**. The project defines all configuration data required to process your data, and keeps all of this information in one place. Each project allows for the definition of an input stream being transformed into one or more destination data streams. For more information on project files, please see [“Output Transformation Engine Terminology” on page 24](#).

3.2.1 Project Checklist

The following represents a typical set of tasks that are performed when transforming a print or data stream into high-fidelity output.

For every project you want to run, do the following:

- Open a project.
- Add the parser and generator(s) you require into the **Development** window and connect them to represent the project workflow.
- Review the following by working your way down the Transformation Projects tab, looking at the information that has been assembled into the project:
 - Parameter settings
 - File paths
 - Resource Collection (if used). (For more information, see *OpenText Embedded Output Transformation Engine - User Guide (VDTOTS-H-UTE)*).
 - Field Definitions in XFT
 - Process Flow components setup (if used)
 - Font correlation (MFCT)
 - I/O Handlers (if used)

3.2.2 Fine Tuning a Project

To fine tune a project:

1. Run the project and check the output. (For more information about processing, see [“Processing Overview” on page 587](#).)
2. Make changes where necessary.
3. Run the project again.
4. Continue to make adjustments, if necessary, to get the output you require.

3.2.2.1 Creating a New Environment

An environment is a collection of mounted file systems, relevant application files, and resources that are used to build a project.

The environment is compressed into a single file to aid portability and deployment across platforms. The compressed environment will need to be manually extracted once it reaches its target system.

To create a new environment:

1. Access the **Environment Manager Wizard** from the menu bar, under **Environment**.
2. Click **New** in the Environment Manager window to launch the **New Environment Wizard**.



Note: If you choose to **Edit** an Environment, the screens are identical to the New Environment Wizard except that the title bar reads **Edit Environment** and *all* the options on the Mount a File System screen are available to you (see Step 4).

The **Create a new Environment** screen appears.

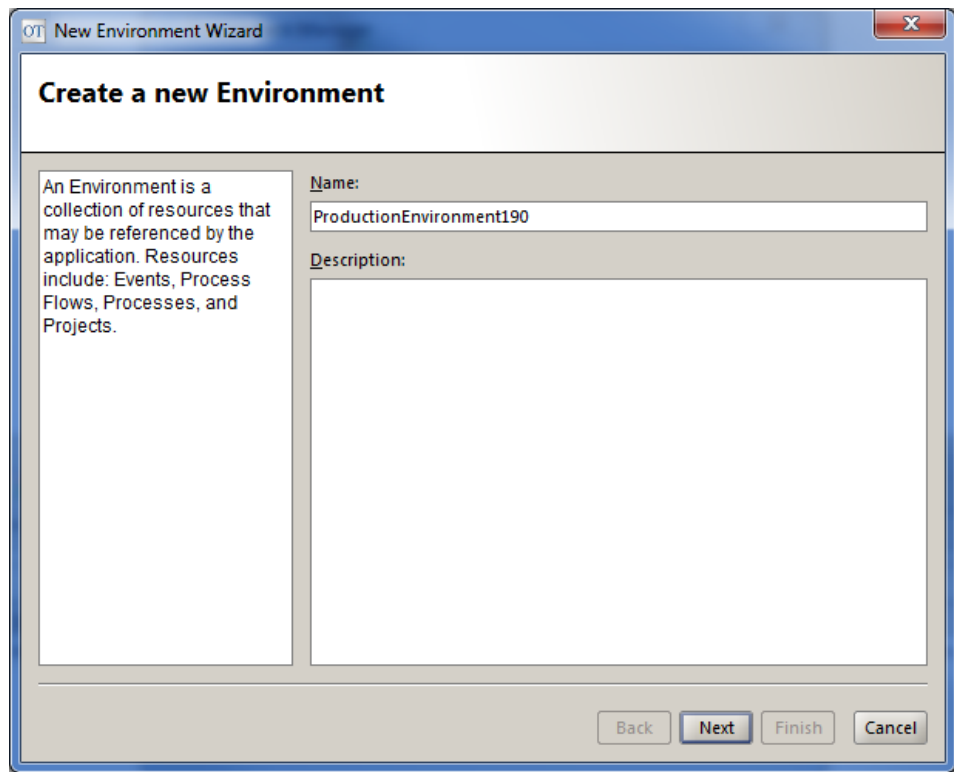


Figure 3-2: Create a new Environment screen from New Environment Wizard

3. Give the new environment a meaningful name in the **Name** field and, optionally, a description in the **Description** field.
4. Click **Next** to open the **Mount a File System** dialog.

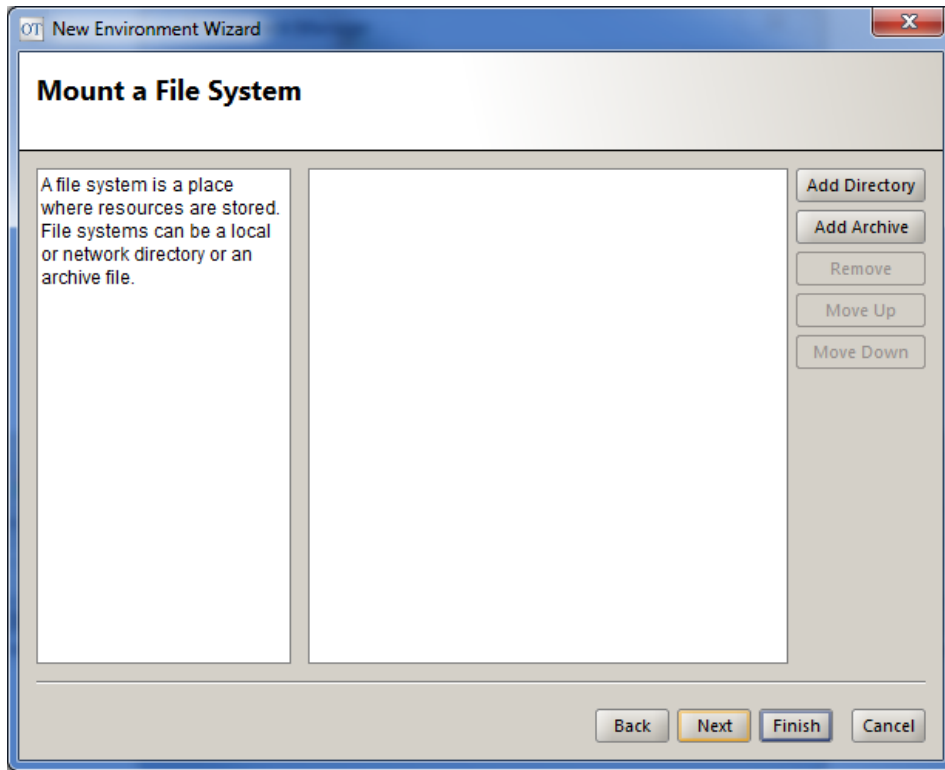


Figure 3-3: Mount a File System from New Environment Wizard

5. Mount the file system as a **Directory** or an **Archive**:
 - A **Directory** is any file system that contains resources to be used in projects.
 - An **Archive** is any archive (*.jar or *.zip) file which contains resources to be used in projects.
6. Navigate to the directory where you would like to mount your files, then click **Add**.



Note: Mount your files in a different directory than the main file system as the main file system is in a standard file location in and cannot be moved or edited.

The added file system(s) is listed in the **Mount a File System** dialog.

The following options are available to you on the Mount a File System screen if you chose to **Edit** one of the environments listed in Environment Manager:

- Add additional file systems by clicking **Add Directory** or **Add Archive**.
- To remove file systems, select the file system and click **Remove**.

- To change the order of file systems to be mounted, select the file system and click **Move Up** or **Move Down**. This is the order in which they will appear in the **File Systems** window.

7. Click **Finish**.

The mounted file systems will be listed in the **File Systems** window.

3.2.2.1.1 Unmounting a File System

To unmount a file system, in the **File Systems** window right-click the unwanted directory or archive and from the context menu that appears, choose **Unmount**.

3.2.2.2 Opening a Project

To open an **existing project**:

1. In the **File Systems** window, expand the mounted file system, and then expand the application where the project is located.
2. Double-click the `.d2epro.j` file that you want to open.

The project opens in the **Development** window in its own tab. (For more information about the Project tab, see [“Project Tab” on page 106](#).)



Note: For more information about creating a new project, see [“New Component Wizard” on page 95](#).

3.2.2.3 Transformation Projects Tab

The **Transformation Projects** tab shows the components of your project in a tree format, with its parser, generators, font correlation table, and other resources. This tab appears in the File Systems window when you open an project.



Note: Multiple projects can be opened simultaneously and will appear on the same tab, with the tab name changing to reflect the project with which you are working.

To expand a branch of the tree, click the plus (+) sign. To collapse a branch of the tree, click the minus (-) sign.

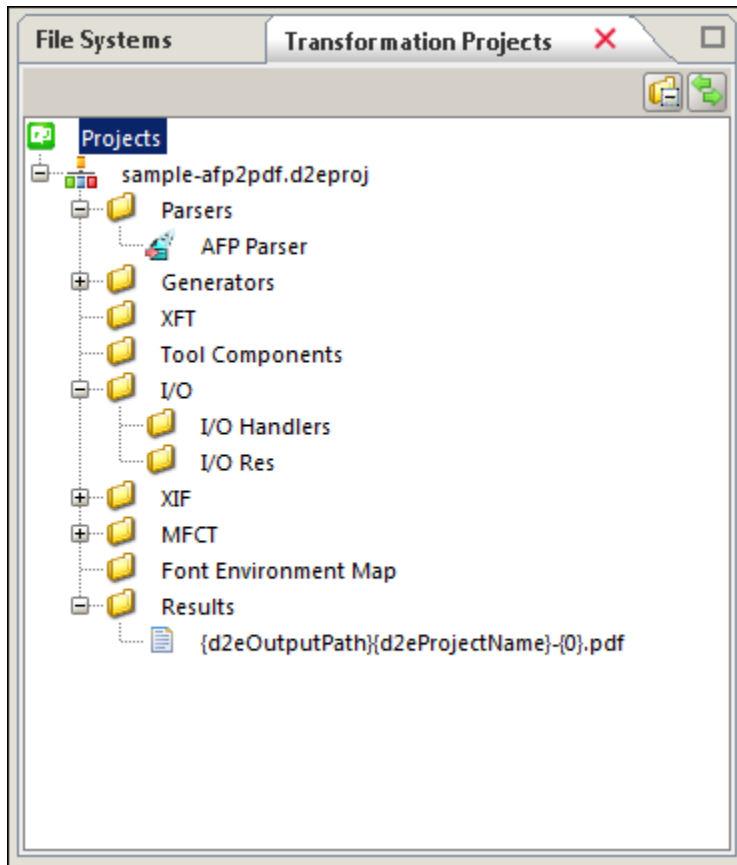


Figure 3-4: Transformation Projects tab with sample project open

3.2.2.3.1 Project

Open projects are displayed in the **Transformation Projects** tab. You can right-click on a project name to access the following options:

- **Run.** Launches the **Execution Options** window, allowing you to run the project .
- **Show Workflow.** Displays the project workflow in the Project tab of the **Development** window.
- **Show Properties.** Opens the Project Properties tab in the **Development** window.
- **Show Manage Fonts/Map Glyphs.** Opens the MFCT tab in the **Development** window.
- **Close Project.** Closes the project.

3.2.2.3.2 Parsers

The parser used in your project is displayed here. Right-click on the parser name to access its shortcut menu:

- **Edit.** Launches the parser wizard specific to that parser.
- **View Resources.** Launches the Resources tab in the **Development** window, displaying the resources for that parser.
- **Page View.** Launches the Page View tab- a file viewer allowing you to see the formatted input file pages.
- **Remove Component.** Removes the parser from the project.

3.2.2.3.3 Generators

All of the generators used in your project are listed in the tree in order of creation. Right-click on the generator name to access its shortcut menu:

- **Edit.** Launches the generator wizard specific to that generator.
- **Generators.** Launches the Resources tab in the **Development** window, displaying the resources for that generator.
- **Remove Component.** Removes the generator from the project.

3.2.2.3.4 XFT

The XFT component that will be used with this project is also shown in the **Transformation Projects** tab. To edit or view the XFT component, either double-click on it or right-click and select **Edit**. The XFT tab appears in the **Development** window.



Note: For more information about working in the XFT tab, see “**XFT Window**” on page 350.

3.2.2.3.5 Tool Components

The components that will be used with this project are located under **Tool Components**. Right-click on the component and select **Edit** to edit details about the component or view component properties.

3.2.2.3.6 IO

The I/O utility components used with this project are listed here:

- **IO Handlers.** Displays the I/O Handlers defined in the current project. (For more information about I/O Handlers, see [“I/O Handlers” on page 112](#).)
- **IORES.** Displays the Resource Collection (IORES) parameters in the **Properties** window when selected. IORES parameter values can be edited in this window.



Note: For more information about the I/O utility, see [“I/O Utility” on page 112](#).

3.2.2.3.7 XIF

Displays the XIF Options parameters in the **Properties** window when selected. XIF Option parameters enable the joining of word and shade fragments and lines segments. XIF Option parameter values can be edited in the Properties window.

3.2.2.3.8 MFCT

The path to the current MFCT (Master Font Correlation Table) is shown to the right of the MFCT icon in the **Transformation Projects** tab tree. Right-click on **MFCT** to access the font table shortcut menu:

- **Add MFCT.** Opens the **Select Resource Path/File window** where you can select the MFCT you wish to add to the project.
- **Save Active MFCT As.** Opens a dialog window to allow you to save the MFCT.
- **Show Active MFCT.** Opens the MFCT tab in the **Development window**, showing the current MFCT.
- **New Default MFCT.** Opens the **Select Resource Path/File window** where you can select the MFCT you wish to set as the default MFCT.

Additionally, you can right-click the MFCT file path entry under the MFCT branch of the project tree to **remove**, **edit**, or make that MFCT the **active** MFCT.



Note: For more information about Master Font Correlation Tables, see [“Master Font Correlation Table” on page 502](#).

3.2.2.3.9 Font Environment Map

The Font Environment Map is a project level component designed to simplify the complex management of font resources.

Right-click on the Font Environment Map node to display a shortcut menu with the following options:

- **Add Font Environment Map.** Opens the **Select Resource Path/File window** where you can select the Font Environment Map you wish to add to the project.
- **Save Active Font Environment Map As.** Opens a dialog window to allow you to save the Font Environment Map.
- **Show Active Font Environment Map.** Opens the Font Environment Map tab in the **Development window**, showing the current Font Environment Map.
- **New Default Font Environment Map.** Opens the **Select Resource Path/File window** where you can select the Font Environment Map you wish to set as the default.

Once you create and activate a Font Environment Map file (.d2efontenv), you can:

- Double-click it to open the **Font Environment Map** tab.
- Right-click on it to remove, edit, or make it the active Font Environment Map.



Note: For more information about Font Environment Maps, see [“Font Environment Map” on page 565](#).

3.2.2.3.10 Results

This section lists the output documents that were created during the processing.

3.2.2.4 Project Properties Tab

Project properties can be defined and edited in the **Project Properties** tab. You can view the tab by right-clicking on the project name in the Transformation Projects tab , and selecting **Show Properties**.

Related Links

- [“Transformation Projects Tab” on page 83](#)

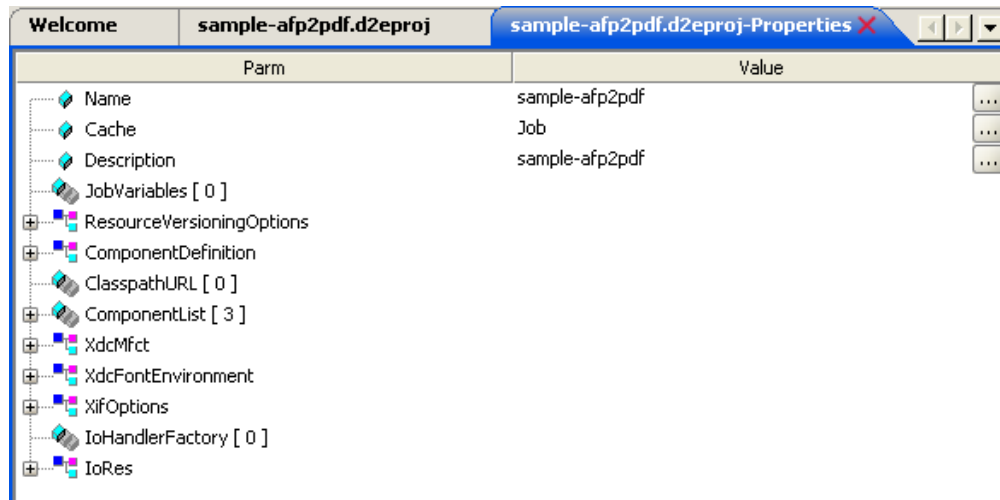


Figure 3-5: Project Properties tab

The following properties and their values are displayed:

Name	Indicates the name of the project. This value can be edited directly under the Value column.
Cache	Determines how or if the project file will be cached in the Resource Manager. Valid values are:
Job	The project file will be parsed for each job.
Sys	The parsed project file will be stored at the system level.
Description	Displays a brief description of what the project does. This value can be edited directly under the Value column.
JobVariables	Displays the variables available for the jobs using the project file. You can right-click on this parameter and select Add to add a variable, or select Remove to remove a variable.
ResourceVersioningOptions	Specifies the following resource versioning options for the project:

VersioningEnabled	Indicates whether or not versioning is enabled in the current context. The default value is True , ensuring that when a ResourceVersionController is detected in the system, versioning will be performed on all resources in the current context. When set to False , resource versioning will be disabled in the current context.
ResourceVersionController	Specifies the class name of the ResourceVersionController (RVC) that implements the <code>com.xenos.d2e.resource.IResourceVersionController</code> interface.
RvcAttributes	Contains the attributes which configure the RVC class.
ShowAttsInLog	Displays RVC attributes in the project report log and the Output Transformation Server engine log during execution if the parameter is enabled. If set to False , an information message similar to RvcAttributes: Parameter... 7 items not shown appears. The number of items in the message refers to the number of attributes you suppress, such as sensitive information like database connection or password information.
ComponentDefinition	Points to the location of custom component definitions.
FileName	Specifies the full path to a Professional Services custom component definition file.
ClasspathURL	Specifies a standard URL to use as a classpath entry.
ComponentList	Lists and displays details about the parser, generators, and tool components used in the project.
XdcMfct	Displays options for the MFCT file.
FdMfct	Indicates the file definition for the MFCT .
XdcFontEnvironment	Displays options for the XDC Font Environment.
FdFontEnvMap	Indicates the file definition for the Font Environment Map .
XifOptions	For more information about XifOptions, see “XIF Options” on page 124 .
IoHandlerFactory	For more information about I/O Handlers, see “I/O Handlers” on page 112 .

IoRes	For more information about IORES parameters, see “IORES Parameters” on page 117 .

3.2.2.5 Page View Tab

The **Page View** tab is a file viewer which allows you to see the formatted input file, at a specific stage of a Output Transformation Engine project flow, one page at a time. The Page View tab opens on its own tab in the Development window and can be accessed through any of the following methods:

- In the **Transformation Projects** tab, right-click on the parser or generator name and select **Page View** from the context menu.
- Open the project in the **Project** tab ; right-click the parser, the generator, or a specific component name in the Output Transformation Engine project flow, and select **Page View** from the context menu.
- Double-clicking an **XFT Field Technology** or a **Constant Print** component in a Output Transformation Engine project flow.



Note: Only one instance of the Page View tab can be opened at any time so if the Page View tab is already open, subsequent requests to view a file will cause the current tab to refresh and display the new file.

Related Links

- [“XFT Field Technology” on page 341](#)
- [“Constant Print Component” on page 238](#)

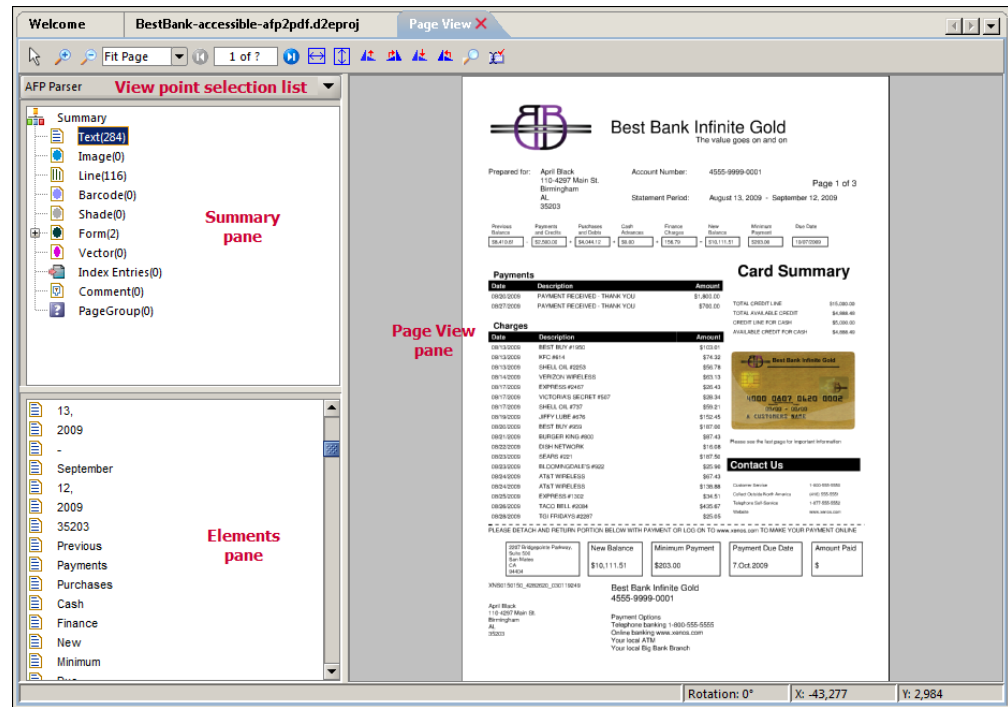


Figure 3-6: Page View tab with Viewer

The Page View tab is comprised of four panes:

- The **View point selection list** shows in tree format the parser, the components, and the generator that are in a Output Transformation Engine project flow. Select any stage of the project flow to see the current view of the inputted file during the file's processing.

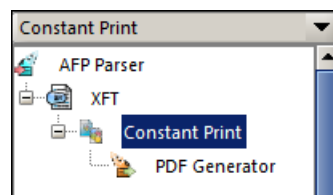


Figure 3-7: View point selection list

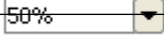
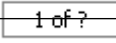
- The **Page View** pane provides a page-by-page view of the input file.
- The **Summary** pane (top left) displays the categories, and the number of elements within each category, that can be found on the page. All parsers and generators, as well as all components, except those listed below, display the categories as shown in the screen shot above.

Exceptions: The Constant Print, Page Markup, and XFT Field Technology components have unique categories displayed in the summary pane.

- The **Elements pane** (bottom left) lists each element on the page that falls under the category that has been selected in the Summary Pane. Click on an element to have its position indicated in red on the Page View Pane, and to display its parameters in the **Properties Window**.

3.2.2.5.1 Page View Toolbar

You can navigate through the document pages using the Page View Toolbar. The toolbar provides the following options:

	Description
	Select Tool. Changes your cursor to the pointer tool.
	Zoom In. Changes your cursor to the Zoom In tool, allowing you to click on the page to increase its magnification.
	Zoom Out. Changes your cursor to the Zoom Out tool, allowing you to click on the page to decrease its magnification.
	Magnification. Specifies the magnification level at which to view the page. You can use the drop-down list to select a magnification level, or enter the desired magnification in the field.
	Previous Page. Changes the view to the previous page.
	Go to Page. Indicates the current page number. You can also enter the page number of the page you want to view.
	Next Page. Changes the view to the next page.
	Fit Page Width. Adjusts the view to fit the page width.
	Fit Page Height. Adjusts the view to fit the page height.
	Rotate 0° Up. Rotates the page to an upright position.
	Rotate 90° Right. Rotates the page 90 degrees to the right.
	Rotate 180° Down. Rotates the page 180 degrees down.
	Rotate 90° Left. Rotates the page 90 degrees to the left.
	Find. Opens the Find dialog allowing you to search for specific text on the page.

	Filter. Opens the Page View Filter dialog where you can select the element types to display on the page.
	Define Images. Allows you to define alternate text for images in accessible PDF documents. This is only available for the Accessible PDF Generator so for more information, see “Accessible PDF Generator” on page 174

When making changes in Page View, clicking **Save** will refresh the view.

If you make changes to XFT items in the Elements pane, the Page View will update dynamically when the XFT item's editing dialog is saved and closed. You do not have to save the project in this case to see the changes.

3.2.3 Wizards

The application provides several types of wizards: a project wizard, parser and generator wizards, and various designers. The wizards can help you complete the following tasks:

- Gathering resources.
- Setting parameters for the project.
- Modelling a new project file on a current project file. This allows you to reuse the work you have already done in creating the original.

The selections you make on each wizard screen populate values to various points within , usually to the project files and components. Changes to these values can be made directly in the **Properties Window**.

The wizards dynamically adjust to the selections you make on each screen, adding steps as necessary to provide additional selection screens. In the PDF Generator Wizard, for example, selecting multiple options at Step 2 may increase the number of steps to accommodate all of the options.

If you choose not to make any selections in the wizard, and instead proceed through the wizard to the end, the project opens with a basic set of default values. In order for your project to run correctly, you need to enter valid data where necessary in the project files and component tools.

3.2.3.1 Available Wizards

This version has the following wizards:

- “New Component Wizard” on page 95

Parser Wizards:

- “AFP Parser Wizard” on page 138
- “Metacode Parser Wizard” on page 157
- “PCL Parser Wizard” on page 161
- “PDF Parser Wizard” on page 166
- “TIFF Parser Wizard” on page 170

Generator Wizards:

- “AFP Generator Wizard” on page 189
- “Image Generator Wizard” on page 192
- “Metacode Generator Wizard” on page 196
- “PCL Generator Wizard” on page 200
- “PDF Generator Wizard” on page 211
- “PDF/A Generator Wizard” on page 219
- “PostScript Generator Wizard” on page 226
- “TIFF Generator Wizard” on page 229
- “Web Generator Wizard” on page 233

Component Wizards:

- “Best Viewed Rotation Wizard” on page 237
- “Constant Print Operation Designer” on page 244
- “Indexer Component” on page 281
- “Index Writer Wizard” on page 272
- “Indexer Wizard” on page 282
- “Replace Text Wizard” on page 324
- “Simple Branch Component” on page 327
- “Skip Page Wizard” on page 328
- “Table of Contents Wizard” on page 330
- “URL Link Wizard” on page 336
- “XFT Field Technology” on page 341

- “XFT Writer Wizard” on page 499

3.2.4 New Component Wizard

The **New Component Wizard** allows you to create a new project by helping you select the files and resources needed to run the project.

Access the New Component Wizard in *one* of the following ways:

- Click the **New** icon on the toolbar.
- Navigate to **File > New**.
- Right-click on the mounted file system and select **New > Component**.
- Navigate to **Tools > Wizard** (this will bypass the **Select a File Type** step and take you directly to the **Select Project Name and Location** dialog).

When the **New Component Wizard** is launched, the **Select a File Type** window is displayed.

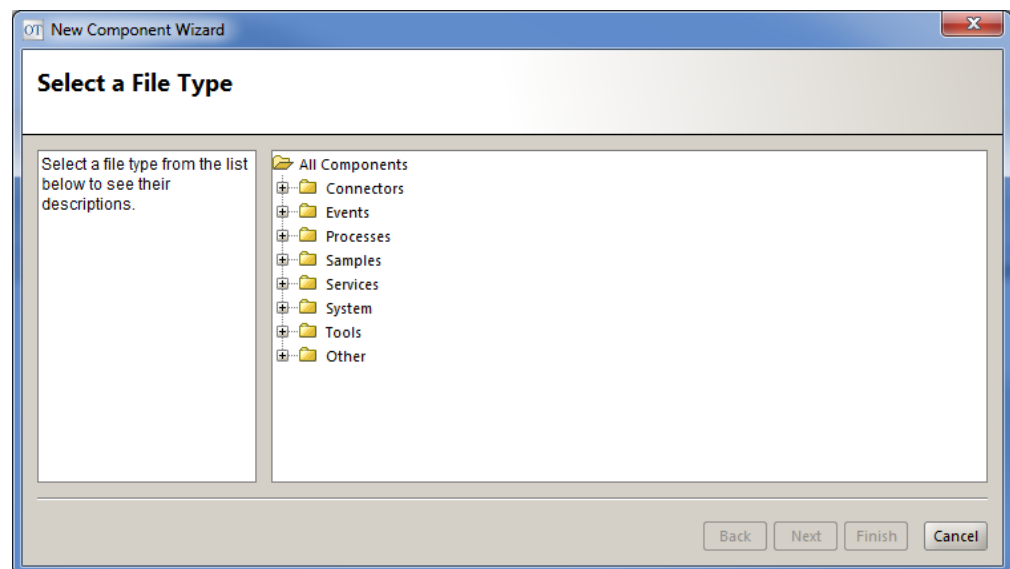


Figure 3-8: Select a File Type screen from New Component Wizard



Note: The components in your list may appear differently than illustrated because the components are governed by your license file.

The **Select a File Type** screen allows you to choose the type of file or component you want to create. To create a project:

1. From the list of component types, expand the **Processes** folder, and then the **Transform** folder.

2. Select **Output Transformation Project** and click **Next**.
3. The **Select Project Name and Location** screen lets you specify the name and location for the new project. The following fields are displayed:
 - **File System.** Use the drop-down list to select the mounted file system where you want the new project to be created.
 - **Directory.** Use the drop-down list to select the directory within the file system where you want the new project to be created.
4. To name your project, you can select one of the following options:
 - Use **standard naming conventions**. The wizard will create a name for the project based on the project type that you select.
 - Enter a **project name**. You can enter any name for the project. The project file extension is `.d2eproj` and cannot be changed.

Complete the fields and click **Next**.

5. The **Select MFCT File** screen allows you to select the Master Font Correlation Table (MFCT) file (`.d2emfct`) to use with this project. The following four options are available:
 - **Select an MFCT file from the list of the application's MFCTs below.** Select a current MFCT file from the drop-down list. Be aware that this action also shares the font table with any other projects that currently have it also set as its font table so any changes to the table settings are also applied across all projects that use the font table.
 - **Browse to select an MFCT file from another location.** Click the ellipsis button and select an MFCT file from another location.
 - **Create a new MFCT file by selecting an existing MFCT file as a base template.** Click the ellipsis button and select an existing MFCT file to use as a template.
 - **Create a new MFCT file based on the default template.** Select this option to create a new MFCT file using the default MFCT shipped with as a template.

Selecting either of the last two options to create a new MFCT file enables the **Using project name as MFCT file name** check box. Select this check box if you want to use the project name defined in the previous step as the new MFCT file name. This is useful if you keep multiple projects in the same directory and you want to associate each project with its own MFCT.

6. Select your MFCT file and click **Next**.
The **Select Parser, Components and Generators** screen is displayed:

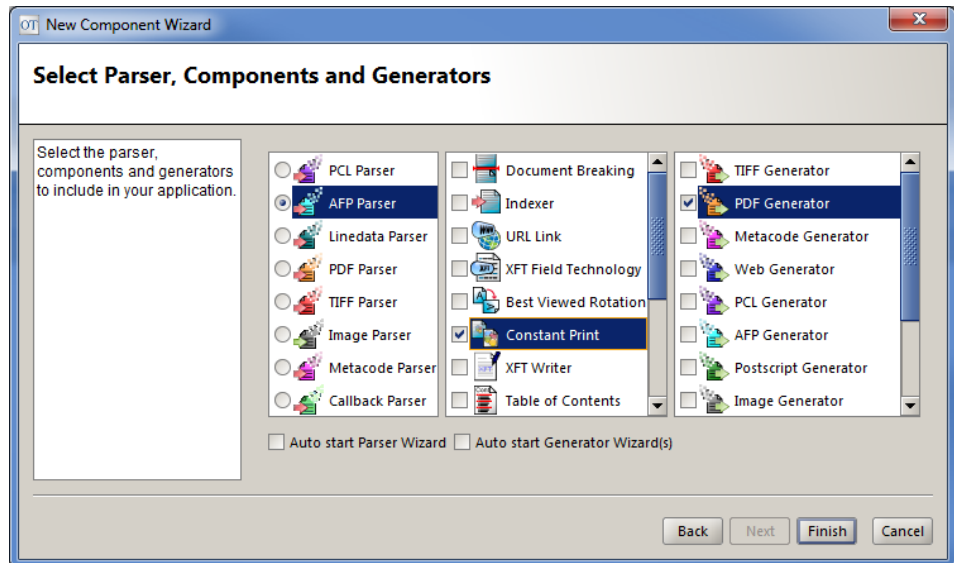


Figure 3-9: Select Parser, Components and Generator screen from New Component Wizard

On this screen, you can select the parser, components, and generators you want to include in the new project. You can also add, remove, or change these later in the Project tab.



Note: It is not possible to have more than one parser running in a project.

7. Select the parser, component(s), and generator(s) for your project.
8. Select **Auto start Parser Wizard** if you want the wizard for the parser you have selected to launch automatically upon completion of the **New Component Wizard**. Similarly, select **Auto start Generator Wizard(s)** if you want the wizard(s) for the generator(s) you have selected to launch automatically.
9. Click **Finish**.

Related Links

- [“Master Font Correlation Table” on page 502](#)
- [“Parsers” on page 131](#)
- [“Components” on page 237](#)
- [“Generators” on page 173](#)
- [“Project Tab” on page 106](#)

3.2.5 Switching Projects in Output Transformation Designer

When Output Transformation Designer is running, you can switch projects in one of two ways:

- Open the **File Systems** tab and double-click any *.d2epro.j file. The open project's process flow becomes the current project in the **Project tab workflow area**.
- Switch between project tabs that are open in the **Project** window.

3.3 Resources

Resources are the various files required for running a Output Transformation Engine project. On a larger scale, these might include MFCT files, project files, and ICU table files. In the context of parsers and generators, resources include fonts, form objects and other data objects needed to process the documents correctly.

Most resource types are automatically loaded via the **Resource Manager**, which provides options to optimize resource management. This improves performance at runtime by avoiding the reprocessing of resources that are used repeatedly.

Related Links

- [“System Configuration Parameters” on page 28](#)

3.3.1 Resources Tab

The **Resources** tab shows the paths and filenames of the resources in the current project. You can display the Resources tab by right-clicking a parser or generator in the Transformation Projects tab and selecting **View Resources**.

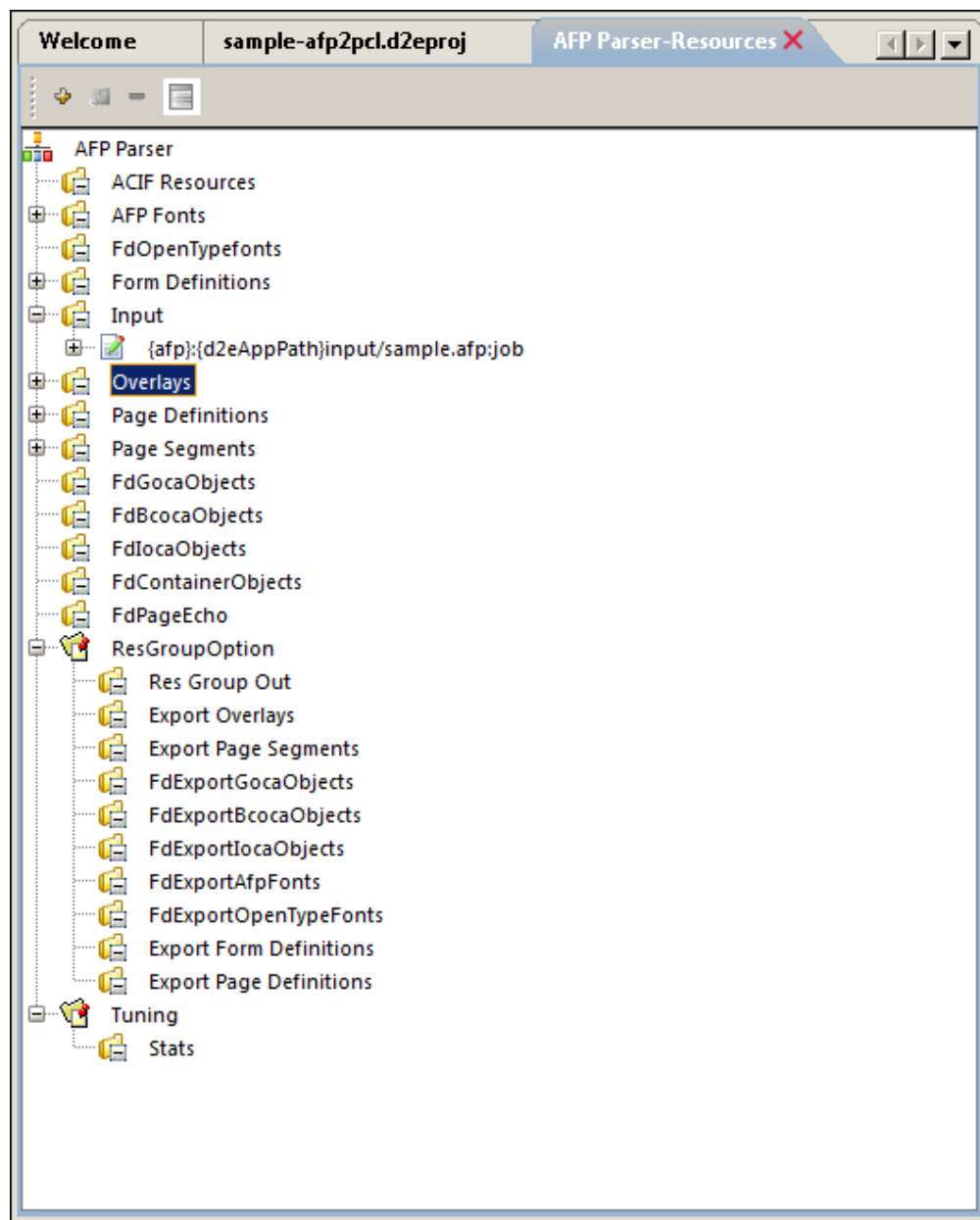


Figure 3-10: Resources tab

Resources can be added, edited, or removed by:

- Using the Resources toolbar.
- Double-clicking on a resource, which displays the Select Resource Path/File window.
- Right-clicking on a resource, which displays a shortcut menu with options to edit or remove the resource.

Related Links

- [“Transformation Projects Tab” on page 83](#)
- [“Resources Toolbar” on page 100](#)
- [“Select Resource Path/File Window” on page 102](#)

3.3.1.1 Resources Toolbar

The Resources toolbar provides the following button shortcuts for the Resources tab functions:

-  or **Add**. Opens the Select Resource Path/File Window to allow you to add a resource to the selected category.
-  or **Edit**. Opens the Select Resource Path/File Window to allow you to edit the selected resource file.
-  or **Remove**. Removes the selected resource from the list.

Related Links

- [“Resources Tab” on page 98](#)
- [“Select Resource Path/File Window” on page 102](#)

3.3.2 File Paths

This section discusses topics related to file paths.

3.3.2.1 File Definitions

Files read or written by Output Transformation Engine are specified by **File Definitions**, which are referred to as “fdefs”. The file definitions are defined to the software through the parameters that begin with “Fd”. They point to a file path where the information will be stored to or retrieved from.

Some of the more common fdefs you will use in Output Transformation Engine are:

- **FdInput**. Defines the input file. For example:

```
FdInput = {afp}:{d2eAppPath}input\sample.afp:job
```

- **FdOutput**. Defines the output file. It’s important to define the output path using the system variable {d2eOutputPath}. Components that produce output would typically have fdefs like:

```
{nocrlf}:{d2eOutputPath}{d2eProjectName}-{0}.pdf:job
```



Note: The {0} is substituted with the resource name.

- **FdAfpFonts.** Defines the Afp Font file location. For example:

```
{afp}:{d2eAppPath}afpfonts/{0}.fnt:job
```

3.3.2.1.1 File Definition Lists (or FDef Lists)

A **file definition list** (or **fdef list**) is a file definition that specifies a location for multiple files, such as font resources. It enables the inclusion of multiple locations for resources. You can specify more than one file definition in your fdef declaration by separating each fdef with a semicolon. For example:

```
FdAfpFonts={afp}:{d2eAppPath}afpfonts\{0}.fnt:job ;\\ {AFP}:{d2eAppPath}{0}.fnt:Job
```

In the **Properties** window, by either double-clicking the **Value** column or selecting the **ellipses** button, , of a **Fd*** parameter, the **Property** dialog for the selected file definition displays allowing you to edit the selected resource category. File definitions are displayed within a tree structure and each branch of the tree is an fdef. The fdefs are evaluated in order from top to bottom.

For more information on file definitions, see [“File Definitions” on page 100](#).

3.3.2.1.2 Fdefs Using Variables

Many file definitions use variables, such as {0}, to represent resource names.

FdOutput	Defines the output file. Typically, you define your output file using a system variable, {d2eOutputPath}. For example: <pre>{nocrlf}:{d2eOutputPath} {d2eProjectName}-{0}.pdf:job</pre>
FdAfpFonts	Defines the Afp Font file location. You define the font name using a resource name variable. For example: <pre>FdAfpFonts={afp}: {d2eAppPath}afpfonts\{0}.fnt:job</pre>

You will need to define a variable for {0}. You can set these values by using a:

- **Wizard.** Identifies the resources, such as an input file.
- **select Resource Path/File dialog box.** Edits the resource path/file.
- **Property window.** Displays current properties for the selected file system.

3.3.2.1.3 Order of Evaluation

Within an fdef list, the fdefs are evaluated in order from top to bottom (or left to right as displayed in the **Properties window**).

For more information on file definitions, fdefs, and file lists, see “[File Paths](#)” on page 100.

3.3.2.1.4 Using Seekable Buffers

When you are using the AFP, Linedata, PDF or TIFF parsers, you can employ seekable (or hybrid) buffers for the file definitions in your document, which make non-seekable input streams seekable as well as allowing seeking backwards during parsing. When parsing AFP or Linedata, enabling the seekable buffer is required if you want to index your original input. Seekable buffer options can be found on the **Property** dialog for your input file definition. The following options are available:

- **Enable seekable buffer.** Indicates that the seekable buffer is enabled for the input stream. The incoming data is buffered using an in-memory buffer and/or a temporary local file. Seekable buffers may be required for processes like parsing PDF or TIFF files over a network stream. By default, it is disabled.
- **Buffer size (KB).** Specifies the size of each buffer, in kilobytes, used to hold data retrieved from the input stream. The minimum accepted value is 8KB.
- **Number of buffers.** Denotes the maximum number of buffers available to hold data before spilling to a temporary file. The minimum accepted value is 1.
- **Only require accurate offsets.** Indicates whether only a stream’s current offset is kept, which does not require any additional memory or temporary files. Users using either the AFP or LineData parsers must enable this option for indexing of their original input. However, if your input stream is a `ByteArrayInputStream`, then this setting is ignored since the input data stream is already held in memory.



Note: If your input stream is already seekable and you enable the seekable buffer on your input, all of these options are ignored.

3.3.2.2 Select Resource Path/File Window

To add or edit a resource path:

1. On the Resources tab, select the type of **resource** you wish to add, or the specific resource you wish to change.
2. Click **Add** to add a resource, or click **Edit** to edit the resource to open the **Select Resource Path/File** dialog:

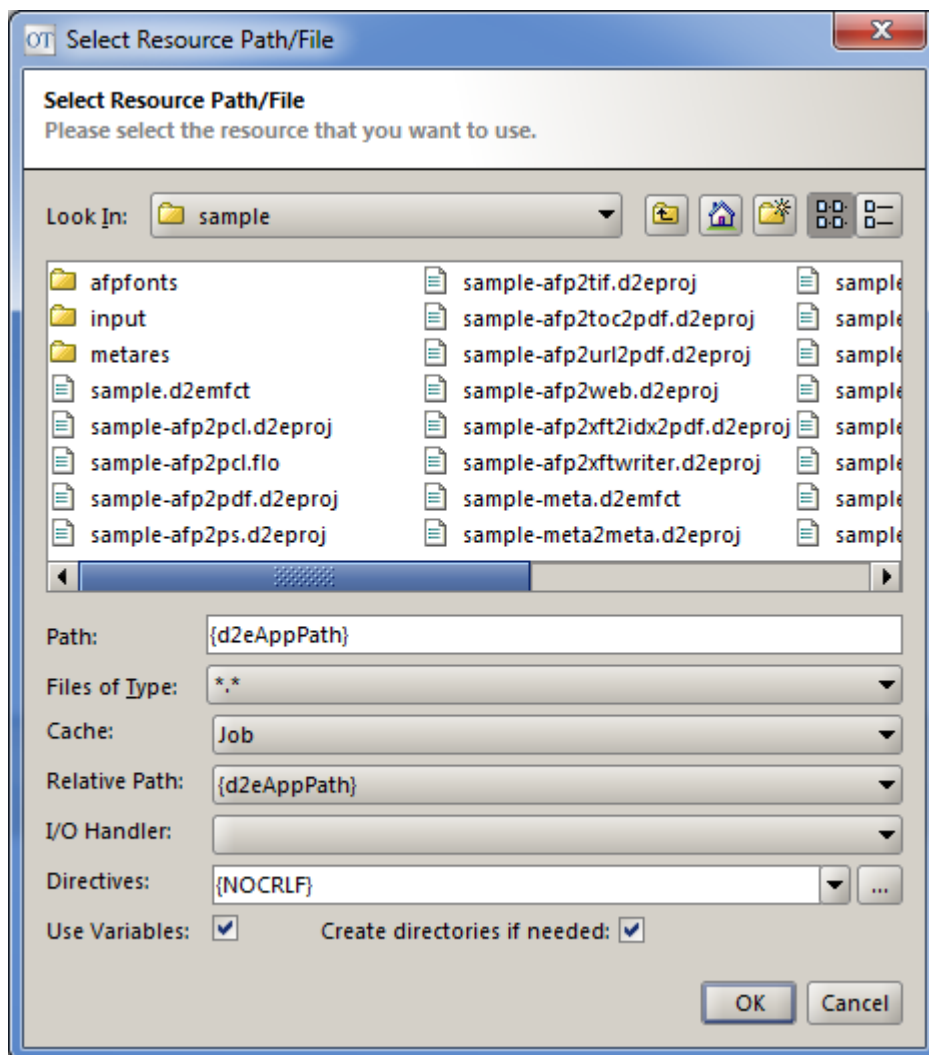


Figure 3-11: Select Resource Path/File dialog

3. Complete the following fields:
 - **Look In.** Use the drop-down list to browse for the resource file you want to use.
 - **Path.** The location of the resource file you want to use. This field is completed for you when you select the resource file.
 - **Files of type.** Select the file type from the drop-down list.
 - **Cache.** Select the cache from the drop-down list. The first time a resource, such as an AFP Font, is loaded into memory, it is placed in the given cache. Valid options are:
 - **Job.** The resource is shared by any process during that job.

- **App.** The resource is cached in memory and available the next time a project from that application is run; the resource is not loaded from disk a second time.
- **Sys.** The resource is available for any job and any application.
- **Relative Path.** Allows you to select a predefined logical location as a path. You can define logical names for directories from the System Configuration tab.

Logical names can be defined for directory locations, such as “CommonAfpResources”. A file definition can be based upon this logical location. The parameter fdAfpFonts could be {CommonAfpResources}fonts/*.fnt and the parameter fdPagedefs could be {CommonAfpResources}pagedefs/*.pde. This allows projects to be relocated based upon the definitions of these logical locations.

- **IO Handler.** Select the I/O Handler you wish to use from the drop-down list. The I/O Handlers that appear have been defined in the System Configuration tab or the Transformation Projects tab under **IO Handlers**.
 - **Directives.** Select the appropriate I/O Directive from the drop-down list..
 - **Use Variables.** Select or clear this check box to switch between using variables or a fully resolved file path. For more information on variables, see “System Variables” on page 52 and “Project Properties Tab” on page 87.
4. Click **OK** to accept any changes you have made, or **Cancel** to exit this dialog without making any changes to the resource path.

Related Links

- “I/O Handlers” on page 112
- “System Configuration Parameters” on page 28
- “Transformation Projects Tab” on page 83
- “I/O Directives” on page 118

3.3.3 Resource Collection (IORES)

The IORES utility provides an easy method to package all resources used by a parser into one single binary archive file. Once the resources are collected into a single file, a project can be run with an IORES file as the only resource.

This enables archived printstreams to be easily transferred as one file stream for on demand conversions, thus reducing complexity and network and/or system overhead. IORES files are also useful for storing resources in ECM systems that do not have support for storing resources associated with the individual documents. When using IORES files, resources do not need to be embedded within the individual documents stored in ECM systems, thus reducing storage requirements.

The popular ZIP format is supported by the standard Java libraries, but it is costly in terms of parsing performance. The IORES Resource Collection format alleviates those concerns.

For further information about resource collections, see [“Resource Collection IORES” on page 605](#).

3.4 Project Workflow

Document processing is handled in through the use of components defined in a project workflow. You can easily add components to your project using the drag-and-drop features of the Project tab.

The simplest project workflow is a document parser directly linked to a generator. This is used for those customers requiring all of the pages of input to be converted into a single output file, with no document breaking or other special handling.

More complex flow diagrams can be defined by adding more process flow components depending on the project requirements.

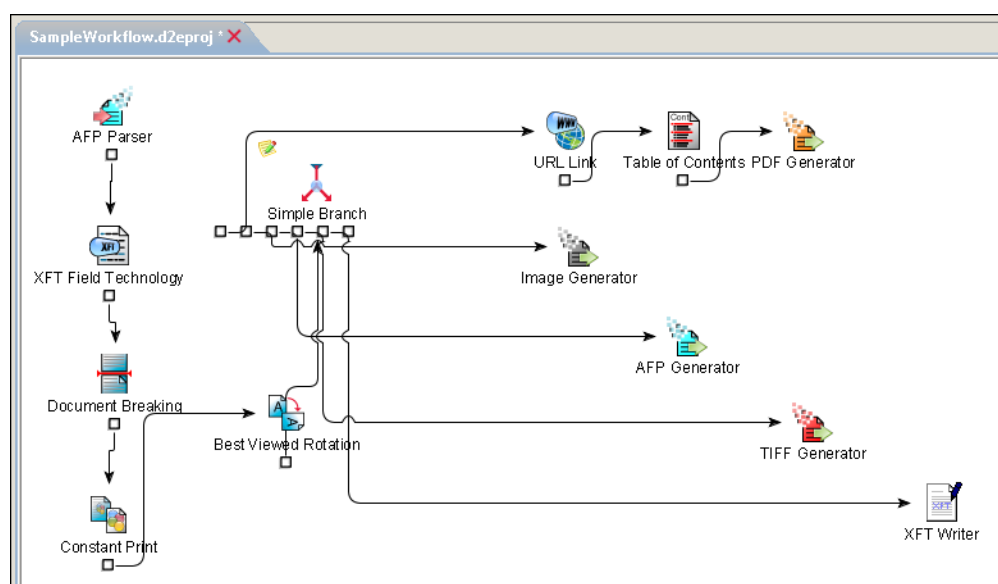


Figure 3-12: Example project workflow

The project illustrated above does the following:

1. Uses the AFP Parser to process and convert the input AFP document to the for subsequent processing.
2. Uses the XFT Field Technology component to define fields in the AFP document for subsequent processing by the following components:

- Document Breaking component
 - URL Link component
 - Table of Contents component
3. Uses the Constant Print component to add a constant text or image to a page or pages in the document.
 4. Uses the Best Viewed Rotation component to rotate each page prior to sending it to the generators.
 5. Uses the Simple Branch Component to distribute parsed AFP, now in XIF format, to produce the following output formats:
 - PDF (Steps 6 and 7 are performed before PDF output is processed by the PDF Generator).
 - Image formats - PNG and TIFF (as processed by the Image Generator).
 - AFP (as processed by the AFP Generator).
 - TIFF (as processed by the TIFF Generator).
 - XML (as processed by the XFT Writer component) to write the information extracted from the input document by the component into a fixed XML format.
 6. Uses the URL Link component to add the URL links to the document before PDF file generation occurs.
 7. Uses the Table of Contents component to add the PDF bookmarks before PDF file generation occurs.



Note: See each component's respective chapter for more information about working with them.

3.4.1 Project Tab

The **Project** tab allows you to drag and drop components into your project. Components are connected in the workflow diagram, where you define document control processing. Using components and connector lines, you can control the flow of your project graphically. As you add, remove, and reposition components from the work area, the project tree is simultaneously changed to reflect the project flow.

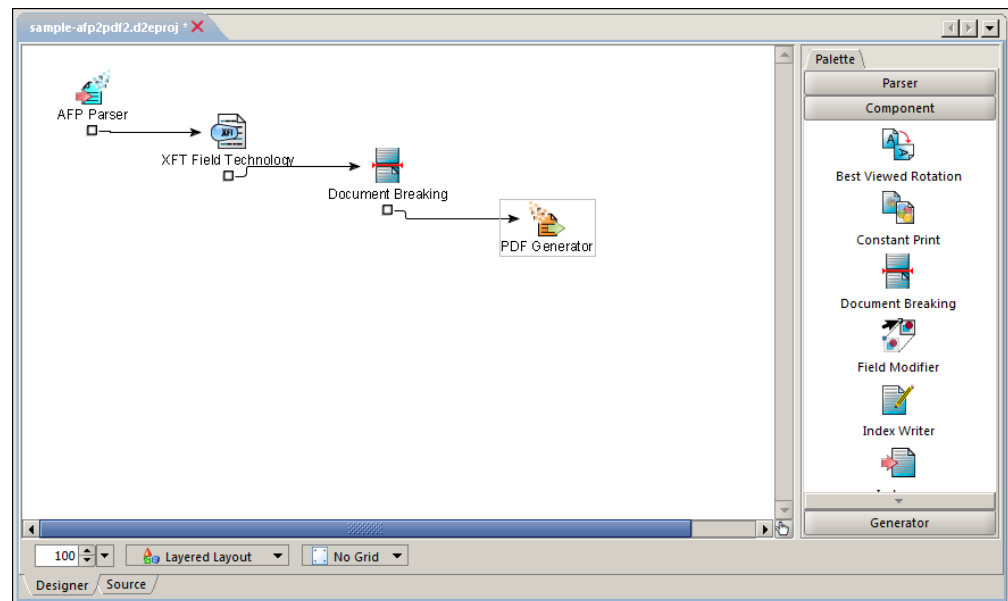


Figure 3-13: Project tab

The Project tab is divided into two areas:

- **Workflow Area.** Provides a work space grid to place components and develop your project flow. For more information, see [“Project Tab Workflow Area” on page 107](#).
- **Sidebar Menus.** The Palette on the right provides icon shortcuts to parsers, generators, and components which may be added to the project. For more information, see [“Project Tab Sidebar Menus” on page 108](#).



Note: You can also open a project by right-clicking a project name in the Transformation Projects tab and selecting **Show Workflow**.

3.4.2 Project Tab Workflow Area

The Project tab workflow area displays a graphical representation of your project flow. It is also where you can drag and drop components into your project. Within the work area you can:

- Add or remove components
- Edit a component
- Connect components
- View component resources

Related Links

- [“Adding Components to a Project” on page 75](#)
- [“Editing a Component” on page 77](#)
- [“Viewing Component Resources” on page 76](#)

3.4.2.1 Producers and Consumers

The work area treats components as a member of one or both of the following types of components:

- **Producers.** A component that outputs data to another component.
- **Consumers.** A component that receives data from another component.

3.4.3 Project Tab Sidebar Menus

The Project tab features separate sidebars for each type of component to facilitate easy access to Output Transformation Engine components.

Click the title of the sidebar you want to view. The available components are listed along the right side of the Project tab.

3.4.3.1 Parser Sidebar Menu

The Parser sidebar lists all of the parsers available to add to your project.

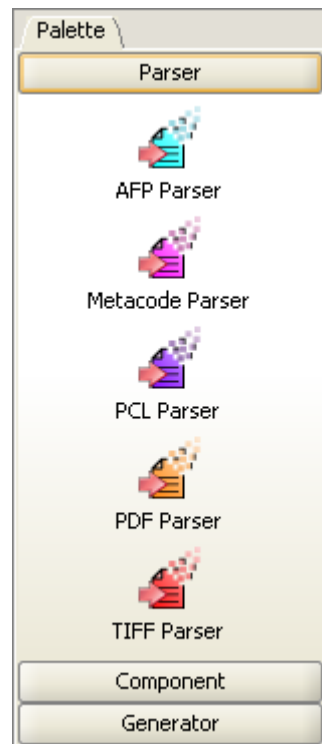


Figure 3-14: Sample Parser sidebar menu

For more information about each parser type, see [“Parsers” on page 131](#).

3.4.3.1.1 Adding a Parser to Your Project

Drag the parser from the sidebar onto the workflow area of the Project tab.

The parser you added to the workflow area is simultaneously added to the Parsers branch of the [“Transformation Projects Tab” on page 83](#).

An icon representing the parser is dropped onto the workflow area when you add a parser. Double-click the parser icon to edit the parser properties and open the parser wizard. For more information on editing parser properties, see [“Parsers” on page 85](#).

3.4.3.2 Component Sidebar Menu

The Component sidebar contains icons representing all the available components.

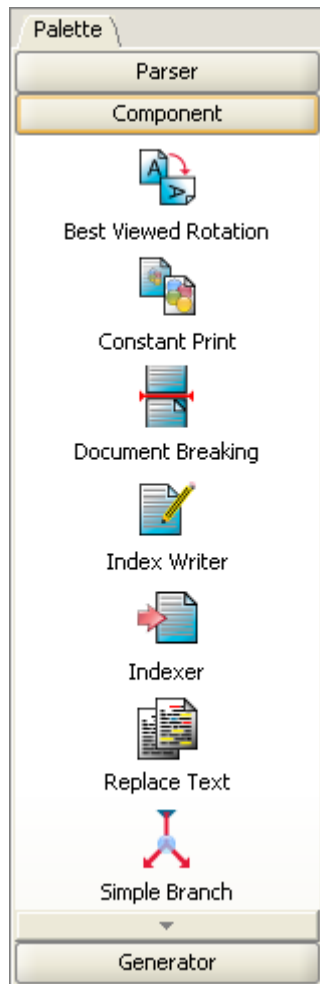


Figure 3-15: Sample Components sidebar menu

For more information about each component type, see [“Components” on page 237](#).

3.4.3.2.1 Adding a Component to Your Project

Drag the component from the sidebar onto the workflow area of the “**Project Tab**” on page 106.

The component you added to the work area is simultaneously added to the Tool Components branch of the “**Transformation Projects Tab**” on page 83.

Adding a component places an icon for that tool onto the workflow diagram. Double-click the icon to edit the component. For more information on editing tool components, see “**Editing a Component**” on page 77.

3.4.3.3 Generator Sidebar Menu

The Generator sidebar lists all of the generators available to add to your project.

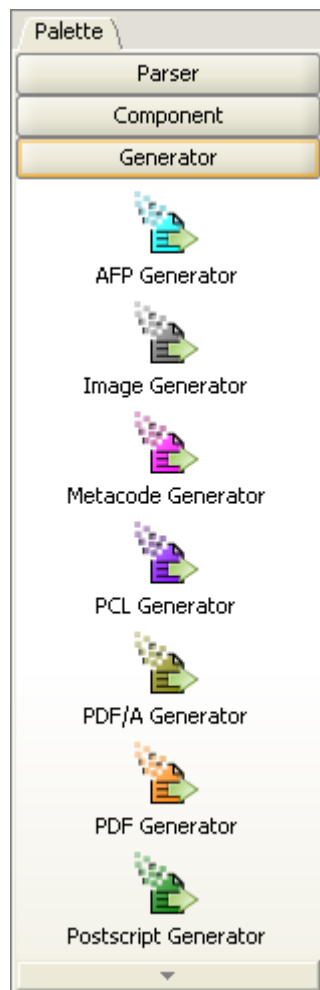


Figure 3-16: Sample Generator sidebar menu

For more information about each generator type, see [“Generators” on page 173](#).

3.4.3.3.1 Adding a Generator to Your Project

Drag the generator icon from the sidebar onto the workflow area of the [“Project Tab” on page 106](#).

The generator you add to the work area is simultaneously added to the Generators branch of the [“Transformation Projects Tab” on page 83](#).

An icon representing the generator is dropped onto the work area. Double-click the icon to edit the generator properties and open the generator wizard.

For more information on editing generator properties, see [“Generators” on page 173](#).

3.5 I/O Utility

This section contains topics related to I/O utility.

3.5.1 I/O Handlers

An I/O Handler is an interface to the I/O Utility. This interface enables internal and external developers to customize input sources, output destinations, record handling and filtering.

Examples of sources and destinations include:

- Byte arrays
- Databases
- Compressed/encrypted files
- Internet based streams
- Document archival systems

3.5.1.1 I/O Handler Factory Lists

Instances of I/O Handler factories may be in the System Configuration tab (system level instances) and in the Application I/O Handler Dialog window (job level instances). These instances are initialized and stored in separate lists, which the I/O Utility queries as it processes [“File Definitions” on page 100](#) (FDefs).

There are two types of I/O Handlers that can be defined:

- System I/O Handlers. These are available to all projects. They are defined in the System Configuration tab by setting the **IOHandlerFactory** parameter. The system level instance list is maintained for the duration of the system run.
- Application I/O Handlers. (Job Level) These are available to one project only. They are defined in the Application I/O Handler Dialog window. The job level instance list exists only for the duration of a job. Although job level instances are

defined in the project, new instances are created for each individual job, and each job maintains its own list.

An I/O handler instance definition consists of the following:

- A unique I/O Handler name, instance name.
- The full class name of the I/O factory and parameters (if any).

Each instance is initialized with its parameters, which are made fully available to the factory for use by custom code.

Once a factory instance has been defined, any file definition may make use of it, as long as the factory has implemented the appropriate method.

Example: To use a factory for an FdOutput **file definition** would require at least one of the four `getOutput` methods to be implemented, and to use a factory for an FdInput FDef would require at least one of the four `getInput` methods to be implemented.

Related Links

- [“System I/O Handlers” on page 114](#)
- [“Application I/O Handlers” on page 115](#)
- [“File Definitions” on page 100](#)

3.5.1.2 The I/O Utility

File definitions will optionally specify the name of an I/O Handler factory instance to use. The `getInputStreamFor` and `getOutputStreamFor` methods determine if User I/O handlers are used, depending on whether there is an appropriate factory in the I/O handler factory list.

When the I/O Utility encounters a file definition with an instance name, it calls `getUserIoHandlerFactory()` via the `IOHandlerFactoryPassback` interface to retrieve the instance of the factory. The `getUserIoHandlerFactory()` method first checks the job level list and then the system level list. If the corresponding instance is found in either list, it is returned to I/O Utility. If it is found in both lists, the instance from the job level list is returned. If it is not found, `getUserIoHandlerFactory()` returns null.

3.5.1.2.1 System I/O Handlers



Note: I/O Handlers require custom code to be written. It is recommended that only people with advanced programming knowledge try to create their own I/O Handlers.

A System I/O Handler is available to all projects loaded into Output Transformation Designer. Before a System I/O handler is available to use in a project it must first be defined in the System Configuration tab.

I/O Handler factory instances are simple to define. All that is required is the full class name (including package name) and the instance name:

```
<!-- instanceName is the "logical" name for this instance -->
<!-- className is the Java class name and can include the package name -->
<iohandlerfactory instanceName="FACTORY1" className="SampleHandlerFactory"/ >
```

The `iohandlerfactoryinstance` parameter enables a file definition to reference a specific I/O Handler factory instance for the I/O Utility to use when it attempts to open the file definition.

Example:

```
<filedeflist name="FdInput">
  <filedef directives="{afp}" location="{inputPath}\sample.afp" cache="job"
    iohandlerfactoryinstance="FACTORY1"/>
</filedeflist>
```

Related Links

- [“System Configuration Parameters” on page 28](#)
- [“File Definitions” on page 100](#)

Defining a System I/O Handler

To define a system I/O handler:

1. Open the System Configuration tab.
2. Right-click the **IOHandlerFactory** parameter and select **Add**.
3. Below the **IOHandlerFactory** parameter, complete the following fields under the **Value** column:
 - **InstanceName.** A name for the I/O Handler.
 - **Classname.** The I/O Handler class name.
 - **Parameter.** Type any parameters and values for the I/O handler.

3.5.1.2.2 Application I/O Handlers



Note: I/O Handlers require custom code to be written. It is recommended that only people with advanced programming knowledge try to create their own I/O Handlers.

An Application I/O Handler is available only to the project in which you define it.

The job level instance list exists only for the duration of a job. Although job level instances are defined in the project (application), new instances are created for each individual job, and each job maintains its own list.

I/O Handler factory instances are simple to define. All that is required is the full class name (including package name) and the instance name:

```
<!-- instanceName is the "logical" name for this instance -->
<!-- className is the Java class name and can include the package name -->
<iohandlerfactory instanceName="FACTORY1" className="SampleHandlerFactory" / >
```

The `iohandlerfactoryinstance` parameter enables a file definition to reference a specific I/O Handler factory instance for the I/O Utility to use when it attempts to open the file definition.

Example:

```
<filedeflist name="FdInput">
  <filedef directives="{afp}" location="{inputPath}\sample.afp" cache="job"
    iohandlerfactoryinstance="FACTORY1" />
</filedeflist>
```

Related Links

- [“File Definitions” on page 100](#)

Defining an Application I/O Handler

To define an Application I/O handler in a project:

1. On the [“Transformation Projects Tab” on page 83](#), expand the **I/O** folder.
2. Right-click the **I/O Handlers** folder and from the context menu that appears, click **Add**.

The **I/O Handler** dialog appears.

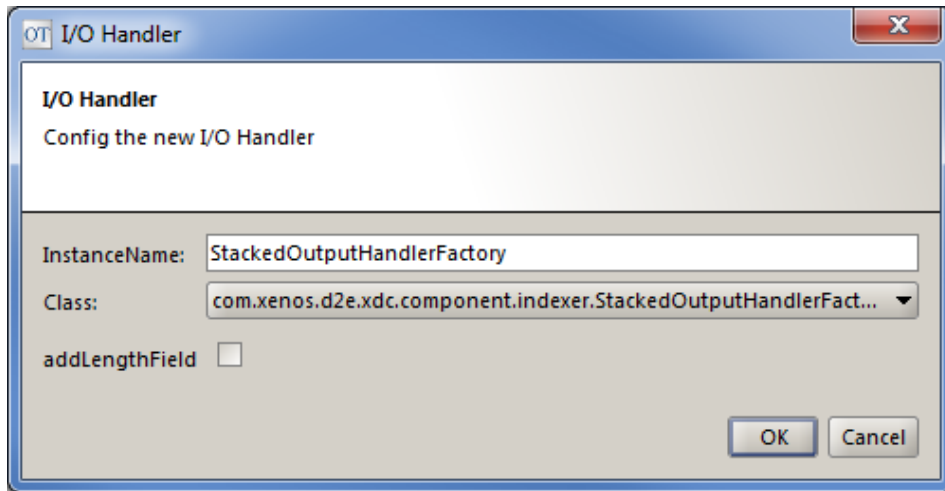


Figure 3-17: Sample configured I/O Handler dialog

3. In the **Class** dropdown menu, you have the option of choosing an existing I/O handler factory class name or creating a new **Custom** class.

For existing registered I/O handler factory classes, the application retrieves a list of known services that extend the `com.xenos.d2e.io.`

`AbstractIoHandlerFactory` base class in order to populate the Class list.



Tip: For more information on registering your own custom I/O factory classes, see .

By default, the `com.xenos.d2e.xdc.component.indexer.`

`StackedOutputHandlerFactory` class, which enables the stacking of output, appears in all projects since it is bundled with the application. Upon loading a I/O handler class, a query is sent to your selected I/O handler factory class and any parameters included your instance of the factory are also displayed. You must select one of the existing classes from the Class list and proceed to the next step.

For custom classes, first select **Custom**. Then using the **Add a new item**, , and **Remove selected item**, , buttons in the **Properties** section, add or remove any new parameters you wish to incorporate into your new custom class. The **Name** field defines the name for the parameter while the **Value** field associates any parm values you want assigned to your I/O handler. Once you are finished setting up your new custom class, proceed to the next step.

4. In the **InstanceName** field, either accept the default name given to your new class or type in a new name for your new I/O handler. This instance name will need to be referenced by FileDef parameter that wishes to utilize this configured I/O handler.

Once you are finished making your selections, click **OK**.

Your new application I/O handler is created and added to the I/O Handlers folder in the tab.



Tip: For more advanced information on working with I/O handlers, see .

3.5.2 Quick Configuration of Predefined I/O Handlers

When configuring the **Output File** for generators on the **Select Resource Path/File** dialog, in the **I/O Handler** dropdown menu you can either select any predefined I/O handler factories you have in your current project or opt to quickly configure a new I/O handler factory by selecting **{New...}**. If you want to create a new I/O handler factory, upon selecting **{New...}** an I/O Handler dialog appears similar to the one shown when you are defining an application I/O handler where you can create a new Custom class or select the `com.xenos.d2e.xdc.component.indexer.StackedOutputHandlerFactory` class to enable and configure a new stacked I/O handler factory. Once you are finished configuring the factory class and saved your new settings, the `FileDef IoHandler` parm is automatically set to use it. Subsequently, the I/O handler will also be accessible in the I/O Handlers folder in the Transformation Projects tab if you wish to review or modify it.

Related Links

- [“Defining an Application I/O Handler” on page 115](#)

3.5.3 IORES Parameters

The IORES parameters provide an easy method to package all resources used by a parser into one single binary archive file.

To ensure that you have all of the settings you need, we recommend you review the IORES parameters in the **Properties** window before running your project.

There are two types of resource collections: **input** and **output**.

- The **InputCollection** parameter only specifies the file definition for the **input** IORES file.
- The **OutputCollection** parameter specifies the:
 - File definition for the **output** IORES
 - Compression type (**deflated** or **none**)
 - Maximum buffer size for an individual IORES entry output stream

If the output bytes for an entry exceed the maximum buffer size, a temporary file is used to store the bytes before they are written to the new IORES file.



Note: For detailed information on resource collection, see [“Resource Collection \(IORES\)” on page 104](#).

3.5.3.1 Examples of the IORES Parameter

```
<iores>
<object name="InputCollection">
<filedeflist name="FdInput">
<filedef directives="" location="{d2eAppPath}font-resources.res"/> </filedeflist> </
object> <object
name="InputCollection">
<filedeflist name="FdInput">
<filedef directives="" location="{d2eAppPath}overlay-resources.res"/> </filedeflist> </
object> <object
name="OutputCollection">
<!-- Compression type, default is "none" -->
<enum name="Compression" value="deflated"/>
<!-- Buffer size in kilobytes -->
<int name="MaxBufferSize" value="20"/>
<filedeflist name="FdOutput">
<filedef directives="" location="{d2eOutputPath}resources-output.res"/> </filedeflist> </
object> </iored>
```

3.5.4 TempPath Job Variable

Temporary files are used to store the resource data while the IORES file is being built.

The **tempPath** job variable specifies the directory in which to create any temporary files that are needed. If **tempPath** is not specified, it is set to the Java system property **java.io.tmpdir**.

The temporary files are deleted on normal termination of the JVM.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area at the bottom of the window will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

3.5.5 I/O Directives

I/O directives are used to control special considerations that the I/O Utility respects when reading and writing files. These I/O directives are specified by setting the I/O directive in the **Select Resource Path/File** dialog. The keywords, or parameter values, are enclosed in braces.

Multiple I/O directive formats can be combined to create more complex instructions. The example below contains an I/O directive that tells the I/O Utility to translate the data from EBCDIC to ASCII when the file is read. It also indicates to the I/O Utility that the records within the file are all 80 bytes in length.

```
IODirective = '{E2A}{F80}'
```

If you are already familiar with the I/O directive formats, you can type the I/O directives directly into the **Directives** field on the **Select Resource Path/File** dialog box. The Directives field also operates as a drop-down list where you can select a particular directive format to use. Alternatively, you can also click the  button to

display the **I/O Directives** dialog box which will assist you with configuration of your I/O directive settings. The following sections are shown in the **I/O Directives** dialog box:

- **Format.** Designates the particular I/O directive format to use. See below for a complete list of the available I/O directives. (Not all I/O directives listed below are available from the **Directives** drop-down list; you can access more I/O directives by selecting  or by typing them in the **Directives** field manually.)
- **Details.** Displays any special settings exclusive to your designated format. If no settings are available, nothing will be displayed in this pane.
- **Translation.** Indicates the type of translation to apply to your files. You can choose from **None**, **ASCII to EBCDIC**, or **EBCDIC to ASCII**.
- **Skip.** Specifies the number of bytes from the beginning of the file to skip and the number of bytes to use from that position. Use the **Start Bytes** and **Length Bytes** options to specify the range to use.
- **Options.** Specifies any secondary options you want applied to your I/O directives. You can choose from **Append**, **Dump to debug file**, **End of File**, **Filter SNA**, or **No Output**.

All available I/O directives are described in the subsequent sections.

Related Links

- [“I/O Utility” on page 112](#)
- [“Select Resource Path/File Window” on page 102](#)

3.5.5.1 {NOCRLF}

The {NOCRLF} I/O directive indicates that the data will have no indicator to separate it into records.

3.5.5.2 {CRLF}

The {CRLF} I/O directive indicates that the records are delimited by a CRLF, which is the 2-byte hex value of 0d0a. The hex value for Windows is a 2-byte delimiter 0xd 0x0a (CR LF) and for Unix is a 1-byte delimiter 0x0a (LF) and these values are only used as delimiters and are not considered as part of the data. If the hex value 0a (LF) is preceded by the hex value 0d (CR), then neither the CR nor LF are considered part of the data.

3.5.5.3 {AFP}

The {AFP} I/O directive is used for AFP input files as the data contains AFP-structured field records without any line data. These records are split based on the built-in length field contained within each structured field. The records may contain a carriage control with a hex value of 5a in column one and the records may end with a CRLF. The first record is integrated for the hex value 5a and a CRLF to determine if all records contain the hex value and CRLF. However, since AFP is binary, the application may incorrectly process data as a hex value 5a carriage control or CRLF. While this situation is very rare, it is important to be aware that it can occur. To mitigate the possibility of this, you can specify the presence of the hex value 5a and CRLF by adding `-5a`, `-no5a`, `-crlf`, or `-nocrlf` to the I/O directive. For instance, `{AFP-5a-nocrlf}` specifies that the file is AFP and has a hex value of 5a carriage control with no CRLFs.

3.5.5.4 {AFPCRLF}

The {AFPCRLF} I/O directive is exclusively used for AFP input files since the data contains AFP-structured field records and line data records. In addition, each record **must** end with a CRLF. This format is slightly different from {AFPLINE} in that the CRLFs must be present.

3.5.5.5 {AFPLINE}

The {AFPLINE} I/O directive is exclusively used for AFP input files since the data contains AFP-structured field records and/or line data records. If the file has line data records, the file must include either an LF or a CRLF at the end of records to separate the line data into individual records. The first record delimiter found in an input file (LF or CRLF) is then used throughout the rest of the file.

3.5.5.6 {AFPLINE-CRLF}

The {AFPLINE-CRLF} I/O directive is used for AFP input files since the data contains AFP-structured field records and/or line data records. If the file includes any line data records, the file must contain a CRLF at the end of each record to delineate the line data into individual records.

3.5.5.7 {AFPLINE-LF}

The {AFPLINE-LF} I/O directive is used for AFP input files as the data contains AFP-structured field records and/or line data records. If the file includes any line data records, the file must contain an LF at the end of each record to delineate the line data into individual records.

3.5.5.8 {AFPNO5A}

The {AFPNO5A} I/O directive is used for AFP input or output files as the data contains AFP-structured field records without any line data. If the input does not contain 5A at the start of each record but is otherwise {AFP} format, then use {AFPNO5A} for input. Likewise, for output if you do not want to include the hex value 5A at the start of each record in the generated AFP, use this directive.

3.5.5.9 {CRLF-DOS}

The {CRLF-DOS} I/O directive is only used for writing files. The file created is a DOS-style text file with records delimited by a hex value of 0x0d 0x0a. This is most commonly used for the report log when an Output Transformation Engine IDC is running on a UNIX server, but note that the report log can only be viewed on a Windows machine.

3.5.5.10 {CRLF-UNIX}

The {CRLF-UNIX} I/O directive is only used for writing files. The file it creates is a UNIX-style text file with records delimited by a hex value of 0x0a.

3.5.5.11 {Fnnn,nnn}

The {Fnnn,nnn}... I/O directive denotes that the records are all the same length, as indicated by the first set of nnn, and are grouped in blocks with a block length indicated by the second set of nnn. (If the data is not blocked, then the second set of parameters is not required. For example, for 80-byte records that are not blocked, you could use {F80}.) This is equivalent to an MVS Fixed Block data set, *recfm=FB*.

Common examples of these files are JSL resources read directly from a Xerox tape. The records are 80 bytes, but are contained in 512-byte blocks, creating 6 records per block with 32 bytes of extra data every 6 records. Using the proper format, this example would be written out as {F80,512}.

3.5.5.12 {LF***}...

The {LF***}... I/O directive indicates a length field at the beginning of the records. Each of the three asterisks, or variables, represents a different option to describe the format of the length fields.

The first asterisk specifies the number of bytes for the length. Valid values are either 2 or 4, representing either a 2 or 4-byte length field.

The second asterisk indicates the format of the bytes. Valid values are **M** for binary in most to least byte order, **L** for binary in least to most byte order, **A** for ASCII digits, or **E** for EBCDIC digits.

The third asterisk specifies whether the value of the length field includes itself. Valid values are either **N** (for inclusive) or **X** (for exclusive).

Examples below show the hex values for records with 64 bytes of data and the I/O directives that represent each:

```
{LF2NM} 00 42
{LF4XM} 00 00 00 40
{LF4XL} 40 00 00 00
{LF4NE} F0 F0 F6 F8
```

3.5.5.13 {RDW}

The {RDW} I/O directive specifies that records have a 4-byte length field.

The length fields for the records are four bytes, with the first two bytes representing the length while the last two bytes are ignored. This directive is equivalent to MVS Variable data sets, `recfm=V`.

3.5.5.14 {BDW}

The {BDW} I/O directive indicates that the records have a length field and are grouped into blocks, which also have a length field.

The length fields for both the blocks and records are four bytes, with the first two bytes representing the length while the last two bytes are ignored. This directive is equivalent to MVS Variable Block data sets, `recfm=VB`.

3.5.5.15 {MVSFTP}

The {MVSFTP} I/O directive indicates that records have a three-byte length field. The first byte has a hex value of 40 unless it is the last record, in which case the hex value is 80. The remaining two bytes are a binary length field. This is the format most commonly used by FTP (File Transfer Protocol) for transferring files to/from MVS. The FTP commands needed prior to sending or receiving files in this format are:

```
QUOTE MODE B
QUOTE TYPE E
```

3.5.5.16 {METABAR}

The {METABAR} I/O directive is used to read Metacode input files. It indicates that Metacode records are delimited by a CR (Carriage Return hex 0d), LF (Line Feed hex 0a), CRLF (Carriage Return Line Feed hex 0d0a), or FF (Form Feed hex 0c). This is generally found when working with offline Metacode printed through a BARR transfer system.

When working with the LineData Parser, traditional line data files have a CC command byte at the beginning of each line to control how spacing and printing should occur. If using the LineDataParser to process regular text files that have no CC byte, select the MACHINE CC option and configure the FdInput parameter to use the {METABAR} I/O directive.

3.5.5.17 {METABARR}

The {METABARR} I/O directive is used for input only and is similar to {METABAR}, but skips the 5-byte header 0xFF 0xFE 0x0D 0x00 0x0A. This directive does not appear in the list and must be explicitly typed in.

3.5.5.18 {METABAR2}

The {METABAR2} I/O directive is equivalent to [4,0] {LF2XL}. This directive does not appear in the list and must be explicitly typed in.

3.5.5.19 {METASPUR}

The {METASPUR} I/O directive converts offline ASCII Metacode to online EBCDIC Metacode with IBM carriage controls. (This differs from {METABAR} which converts offline ASCII Metacode to online ASCII Metacode with ANSI carriage controls.) It indicates that Metacode records are delimited by a CR (Carriage Return hex 0d), LF (Line Feed hex 0a), CRLF (Carriage Return Line Feed hex 0d0a), or FF (Form Feed hex 0c). This is typically found when working with offline Metacode printed through a spur transfer system.

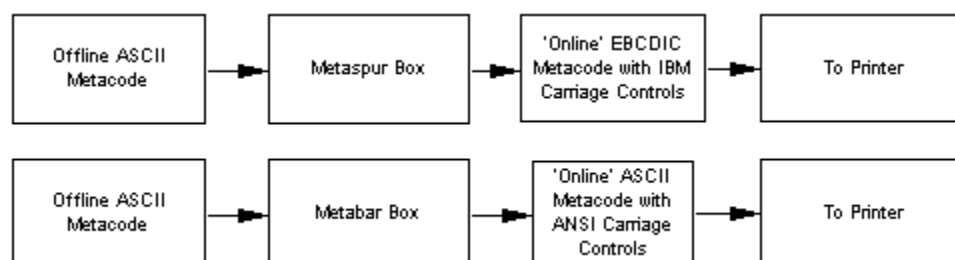


Figure 3-18: Comparison of {METABAR} and {METASPUR}

3.5.5.20 {LF2NM}

The {LF2NM} I/O directive denotes that the record length is also inclusive of the field length, which is set at 2 bytes, and that the bytes are arranged in order from most significant to least.

3.5.5.21 {CRLF-MVS}

The {CRLF-MVS} I/O directive permits the use of 0x15 as a record terminator, which is useful for processing MVS-based data. 0x15 can also be used as a record delimiter in both input and output and is not included as part of the data. 0x15 is similar to the use of 0x0a (LF) in Unix.

3.6 XIF Options

The XIF Options parameters, accessible from the XIF folder on the Transformation Projects tab, provide a variety of options that can be executed at the XIF Document or XIF Page level of the project workflow at runtime. Typically, a Output Transformation Engine parser will construct **XifPage** objects, which represent a single page in the XIF Intermediate Format, and which are later processed by a Output Transformation Engine generator. While other parameters will take effect after the parser completes an entire XifPage, certain parameters defined in the **XifOptions** will take effect as the parser adds elements, such as text, images, or forms, to the XifPage.

For more information about the parameters available for XIF options, see “[XIF Options Parameters](#)” on page 124.

3.6.1 XIF Options Parameters

There are several categories of XIF Options parameters, which are displayed in the **Properties** window. A description of the function of each XIF Option parameter is displayed in the message window when you select a parameter. The concepts of each category are summarized in the following sections.

3.6.1.1 XifFragmentJoining

Some input documents may contain text that is *fragmented*, meaning characters are not grouped into whole words.

As an example, consider the following three examples of how the text “Hello World” might be represented technically in a document. **Document C** contains fragmented text and **Document A** and **Document B** are examples of non-fragmented text.

Document A, positions text as follows:

- Use font "Arial", Set pagePosition = (x,y)=(100, 100)
- Print text "Hello World"

Document B, positions text as follows:

- Use font "Arial", Set pagePosition = (x,y)=(100, 100)
- Print text "Hello"

- Set pagePosition = (x,y)=(110, 100)
- Print text "World"

Document C, positions text as follows:

- Use font "Arial", Set pagePosition = (x,y)=(100, 100)
- Print text "He"
- Set pagePosition = (x,y)=(103, 100)
- Print text "ll"
- Set pagePosition = (x,y)=(108, 100)
- Print text "o"
- Set pagePosition = (x,y)=(110, 100)
- Print text "W"
- Set pagePosition = (x,y)=(113, 100)
- Print text "or"
- Set pagePosition = (x,y)=(115, 100)
- Print text "l"
- Set pagePosition = (x,y)=(117, 100)
- Print text "d"

The type of fragmentation in **Document C** tends to be problematic for processes such as document conversion, indexing, or data mining. In document conversion, for example, text must often be mapped to an output font that has slightly different display properties (metrics) than the original input font. If the text is fragmented and the font is slightly narrower, visual gaps may appear between words.

The `<XifFragmentJoining>` parameters provide the ability to join fragments together to help eliminate these problems. The `<threshold>` value specifies the maximum amount of space required to cause two fragments to be joined. The `<threshold>` value is in 72000 DPI (XIF resolution), meaning that 72000 is equivalent to 1 inch. An example of XIF fragment joining is shown in Figure 3-10.

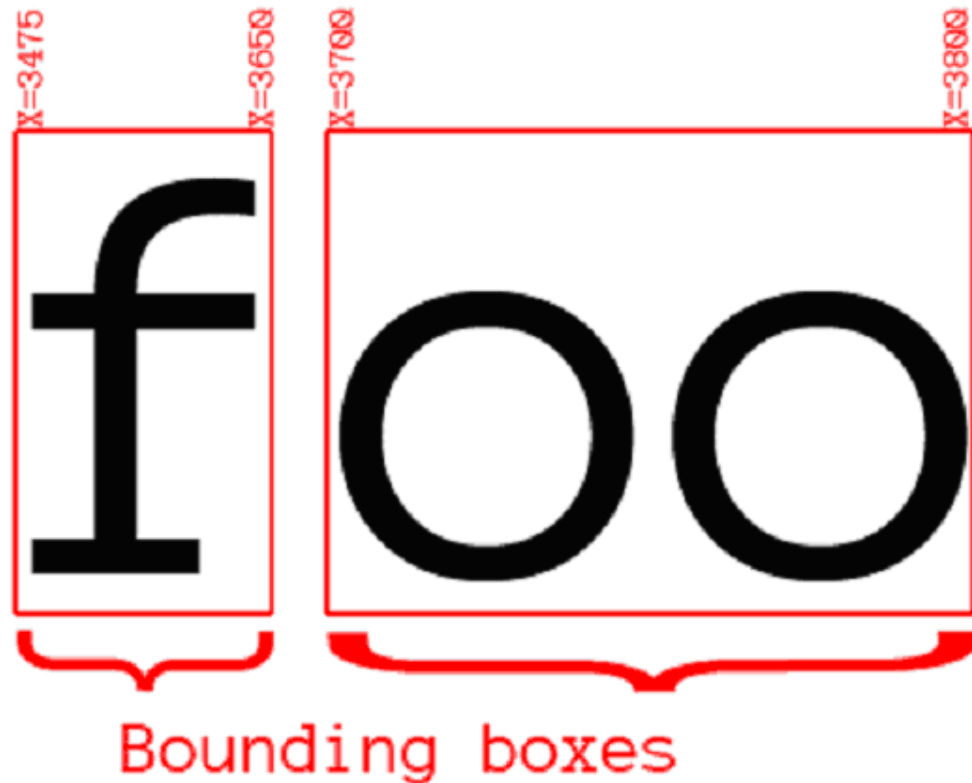


Figure 3-19: XIF fragment joining example

In this case, there are two fragments, **f** and **oo**. However, the desired single word element is **foo**. Based on the bounding boxes of the two elements, we calculate that there is a gap of 50 XIF units (72000 DPI) between the two elements ($3700 - 3650 = 50$). So the threshold parameter value would have to be 50 or larger to result in a joined fragment.

You are encouraged to experiment with threshold values to see what works best for your documents. It is recommended that a threshold of zero (0) be used initially, because this may result in the desired effect if the bounding boxes actually have no gap between them.

Because the XFT component also offers the functionality to display all of the text elements on a page in list form (with their bounding box coordinates), the XFT component is a very useful tool when determining appropriate threshold values.

3.6.1.2 XifLineJoining

The **XifLineJoining** parameter offers functionality similar to the **XifFragmentJoining** parameter. The difference is that it affects line elements rather than text elements. This helps eliminate occasional visual issues, such as gaps between lines, in the output format of some projects.

XifLineJoining can also successfully reduce the file size of some output documents that contain a large number of line drawing commands.

Related Links

- [“XifFragmentJoining” on page 124](#)

3.6.1.3 XifShadeJoining

The **XifShadeJoining** parameter offers functionality similar to the **XifFragmentJoining** parameter. The difference is that it affects shade elements rather than text elements. This helps eliminate occasional visual issues in the output format of some projects.

XifShadeJoining can also successfully reduce the file size of some output documents that contain a large number of shade elements.

Related Links

- [“XifFragmentJoining” on page 124](#)

3.6.1.4 XifVectorOptions

The **XifVectorOptions** parameter enables you to customize how vector graphic elements, such as curved lines, rectangles, polygons, and pie charts, are processed. The key function of this parameter is to allow the vector elements to be rasterized and treated as images in the output document. This is occasionally an essential requirement.

For example, although the AFP specification supports vector elements using GOCA objects, some printers and archiving systems cannot process GOCA commands. In these cases, using **XifVectorOptions** to rasterize all vector graphics elements ensures the AFP Generator creates raster images to represent the vectors rather than generating GOCA content.

3.6.1.5 XifDrawingOptions

The **XifDrawingOptions** parameter enables you to customize how the generator components process graphical XIF elements in the output document(s).

The **DrawingOrder** value controls the drawing order of XIF graphical elements (such as text, shades, lines, and vector graphics) by the output generators. The default value is **Optimized**, which causes generators to group elements of the same type together in the output document structure. This provides the most efficient document structure in terms of file size and transform time. However, in some cases, the optimized option may not preserve the desired visual fidelity of the input document, resulting in elements, such as vector-based pie charts, to be hidden (or masked) by other elements, such as lines or images. To correct this, set the value to **Original**, which will ensure that the generators preserve the original drawing order of the input document.

If you are creating accessible PDFs and using the artifacts tag, bear in mind that setting XifDrawingOptions to Optimized retains all elements tagged as artifacts on the output page. Furthermore, the drawing order for Optimized groups elements of the same type together in the output document structure but can be specified by the Page Markup component so tagging elements as artifacts can help to control the drawing order. For more information about artifacts, see [“Tagging Elements as Artifacts” on page 300](#).

3.6.1.6 XifRasterOptions

The **XifRasterOptions** parameter enables you to control the behavior of rasterized output. This functionality applies to components such as the TIFF Generator , Image Generator, Page Viewer, and XFT Wizard.

The **HalftoneOptions** control how some XIF elements are converted from color to black and white.

3.6.1.7 XifTextOptions

The **XifTextOptions** parameter provides options for controlling how text is processed.

The **TextPositioningOverride** option allows users to override how all text strings in the job are created (this can be controlled at the individual font mapping level as well). **ForceWordPositioning** causes the parser to break up text strings found in the input document into individual words (each having an absolute position on the page). Space characters are discarded in word positioning.

ForceRelativePositioning will preserve the original strings in the input document intact. It is generally recommended that ForceRelativePositioning only be used when the font mapping cannot be improved to achieve the desired output document fidelity.

3.6.1.8 XifAcroFormOptions

The **XifAcroFormOptions** parameter enables you to customize the style of generated AcroForms. Form fields that can be customized are check boxes, radio button, input text, and push buttons.

3.6.1.9 XifImageOptions

The **XifImageOptions** parameter contains options related to image processing.

The **StencilImageOption** value specifies the type of processing used to get data bytes for the one-bit stencil image mask checksums for use in Alternate Text Manager. You can choose from **Original**, **New**, or **Development**. It is important to be aware that the Original and New settings produce different checksums; the Original method provides slower performance while selecting New uses a faster method to acquire data bytes for the checksum. The Development method provides both an Original and New checksum, but only uses the Original checksum for processing.

Chapter 4

Parsers

The Output Transformation Engine parsers transform incoming printstreams to the XIF Intermediate Format. The XIF contains all the objects from the parsed pages, which can then be read and/or manipulated by other Output Transformation Engine components.

The parser evaluates your input data stream and parses it according to the parameters that have been set in the **Properties window**. The parser looks for particular values, control sequences, keywords, and other specific parameter options and passes this information to the generator in order to provide correct output.

The parameters you set in the project file determine the following:

- How your input file will be parsed and prepared for processing.
- How it is handled by the generator.
- How it is then output.

The parser has a set of default parameters, however, these are often not sufficient to transform other than the most basic documents.

Output Transformation Engine Parsers

The parsers available with Output Transformation Engine include:

- AFP Parser. Parses input IBM AFP printstreams including MO:DCA-P, IOCA, GOCA, BCOCA, ACIF.
- Image Parser. Processes and converts various image formats.
- Linedata Parser. Parses unstructured text-based printstreams.
- Metacode Parser. Parses input Xerox Metacode and LCDS, often referred to as DJDE printstreams. Line data and mixed mode files that print on Xerox Centralized printers are supported, as well as pure Metacode files.
- PCL Parser. Parses HP PCL including PCL 5 and PCL 6 (PCL XL).
- PDF Parser. Parses input Adobe PDF 1.0 to 1.6 files (Acrobat 7.0 or earlier), including those generated by d2e Platform and Output Transformation Engine PDF Generators files.
- TIFF Parser. Parses industry standard multi-page TIFF Class B, G, P, and R formats, and 1, 2, 16, 256, RGB and CMYK color depths. Supports CCITT Group4 and Group3 compression.



Note: You will have access to these parsers based on the permissions granted in your license file.

Parser Wizards

All parsers have wizards associated with them to help you define their parameters. Whether or not you use a Parser Wizard to create your parser project file, please review your parameters to ensure you have established all of the required parameters for your Output Transformation Engine project.

Related Links

- [“Wizards” on page 93](#)

4.1 AFP Parser



The Output Transformation Engine AFP Parser is designed to process and convert IBM's Advanced Function Presentation Data Streams (AFPDS) to XIF for subsequent processing by Output Transformation Engine generators.

Features

Features of the Output Transformation Engine AFP Parser include:

- Line data and mixed mode files that print on AFP Printers through IBM Print Services Facility (PSF).
- Fully composed MO:DCA-P files.
- BCOCA Barcode Objects.
- Overlays, Page Segments (Psegs).
- Single color IOCA images with Function Set 10, 11, 20, 42, and 45 support.
- GOCA vector graphics, including mixed CMYK and RGB color definitions.
- ACIF Resource files.
- Extraction of information from comment, NOP, and TLE records for indexing purposes.
- AFP Outline Fonts. Supported for mapping only; no rasterization is done by mapping to a True Type font, such as for TIFF output.
- Double-byte character sets (DBCS) support and double-byte SOSI scheme (PRMODE).
- Bounded box fonts and core interchange family (CIF) fonts.
- Standard International Components for Unicode (ICU) table can be used to simplify mapping AFP encodings into Unicode.
- Auto-resource preview of Overlays and PSegs. By selecting a single parameter, you can preview resources used within an AFP job, creating a PDF or TIFF file for each resource.

- Assisted font mapping in conjunction with OpenText Output Transformation Designer. After running the project, the Manage Fonts tab shows what fonts were used and how they were correlated. With a single click, you can change the font correlation and run the project again to see the results.
- Resources, either inline or in an ACIF file, can be optionally extracted into separate files, for easier resource and data analysis. (For more information about setting external resources, see [“ExternalResources” on page 136.](#))
- Flexible Open Type font support with Unicode (UCS-2) encoded data. (For more information about parsing AFP data with OpenType fonts, see [“Parsing AFP Data Utilizing OpenType Fonts” on page 143.](#))
- Ability to process multiple input documents. (For more information about using the {LIST} I/O directive, see [“I/O Directives” on page 118.](#))

Related Links

- [“Running the Project” on page 587](#)
- [“Manage Fonts” on page 562](#)

4.1.1 AFP Parser Parameters

The AFP Parser parameters are used with the AFP Parser. A default set of parameters and values is used when you add the AFP Parser to your project. When you work through the [“AFP Parser Wizard” on page 138](#), you develop parameters that are specific to your project.



Note: If you make changes directly in the Properties window, remember to tab or click out of Edit mode to ensure the changes are captured.

To ensure that you have all of the settings you need, we recommend you review the Parser parameters in the **Properties window** before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties window** and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

4.1.1.1 Setting Page Range Parameters

Page range values can be changed in the **Properties** window, in its associated wizard, or in the Job Options window of the Job Execution Options form.

The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Properties window. Similarly, if you change these values in the Parameters, the new values will override any previous settings until a change is made to either the Job Options tab or the Parameters. Changes made in the **Properties** window will be displayed in the **Job Options** window, but not vice-versa.

4.1.1.2 ContentFilter

The **ContentFilter** parameter allows you to control the content being added from an AFP page to the intermediate XIF page.

The following types of resources may be present in an AFP page:

- Overlays referenced by IPO structured fields
- Page segments referenced by IPS structured fields
- Object containers referenced by IOB structured fields
- Embedded images (IOCA, IM)
- Embedded vector graphics (GOCA)
- Embedded barcodes (BCOCA)
- Text

Using the ContentFilter parameter, you can specify which of these resources to convert to XIF elements. For example, you could exclude a certain type of resource, such as a barcode, from being processed by a generator. You could also configure the filter to process certain required fonts if only a small amount of text in a document needs to be captured by XFT. This will reduce the number of XIF text elements created for XFT processing, increasing performance substantially.

ContentFilter Properties

The ContentFilter properties correspond to each of the resource types whose inclusion or exclusion can be configured within the ContentFilter parameter. In addition, each resource type has the following options:



Note: If all 3 options are left blank, then all instances of the resource will be included.

- **ExcludeAll.** Select this check box to exclude all instances of the resource. This will ignore any entries in the Include and Exclude lists.

- **Include.** Specify the resource name(s) to be included. This list is processed before the Exclude list.
- **Exclude.** Specify the resource name(s) to be excluded. After the Include list is processed, the matching entries in the Exclude list are subtracted.



Note: The Include and Exclude lists can be used together with the wildcard symbol (*). For example, you could specify to include all overlays matching "01AB*", but to exclude overlay "01AB0001".

The ContentFilter properties are:

Enabled	Select this check box to enable the ContentFilter functionality. This is disabled by default.
Overlays	Specify whether overlays referenced by an IPO are included or excluded from the XIF page.
PageSegments	Specify whether page segments referenced with an IPS are included or excluded from the XIF page.
ObjectContainers	Specify whether objects referenced with an IOB are included or excluded from the XIF page.
Images	Specify whether images embedded in a page are included or excluded from XIF page.
Goca	Specify whether vector graphics embedded in a page are included or excluded from the XIF page.
Bcoca	Specify whether barcodes embedded in a page are included or excluded from the XIF page.
Text	Specify whether text on a page is included or excluded from the XIF page. This value must be entered as a font name.
Lines	Specify whether PTOCA lines on a page are included or excluded from the XIF page. This resource type offers only the ExcludeAll option.

4.1.1.3 SkipInlineResOptions

The AFP parser provides **CacheByName** and **CacheByChecksum** options under **SkipInlineResOptions** to optimize caching of resources that are found in inline and/or ACIF resource groups. This benefits environments with the following criteria:

- The majority of input AFP resources are kept in resource groups instead of individual external resource files.
- Those resource groups often contain the same resources.
- The engine will be performing multiple transformations in the same instance, such as dynamic web requests.
- Resource caching is configured using system level caching, set in the Resource Manager in the System Configuration (d2esys) file.



Caution

When using the **CacheByName** and/or **CacheByChecksum** options, the **ResourceManager** cache scheme in the d2esys file must be set to **hard cache** scheme. Soft caching schemes cannot be used with these configurations because resources contained in the resource groups can only be added to the cache as the resource group is being loaded at the start of the job before the AFP document data is processed. The risk is that the soft cache may be emptied sometime during the processing of the document. If this happens, the dropped resources would be unavailable when a page further in the document is processed.

4.1.1.4 ExternalResources

To use some functions of , such as the Constant Form feature of the Constant Print Component, the AFP parameter **ExternalResources** must be set. The **ExternalResources** parameter pre-loads resources to improve performance. For example, an AFP overlay is preloaded for use with the Constant Print component.

To set the AFP Parser parameter **ExternalResources**:

1. Select the **AFP Parser** in the Transformation Projects or the Project tabs.
2. In the **Properties** window, right-click on **ExternalResource** and select **Add**.
3. Type the **name of the resource** you want to preload for use with components. Use the name of the resource without the file extension, for example, overlay1 not overlay1.ovr.



Note: To preload a resource, the resource must be present in the Overlays file definition on the Resources window.

Related Links

- [“Constant Form” on page 256](#)
- [“Constant Print Component” on page 238](#)
- [“Transformation Projects Tab” on page 83](#)
- [“Project Tab” on page 106](#)

4.1.1.5 UnresolvedResourceOption

The **UnresolvedResourceOption** parameter provides the ability to flag when an AFP resource is missing or corrupt and the type of error severity message to issue. The engine’s subsequent behavior upon flagging an unresolved resource is also dictated by the various engine code levels. You can choose from either **WarnOnly**, **ErrorOnly**, or **FatalError** level messages. **WarnOnly**, the default, indicates a warning message is issued along with a return code of at least 4 and the job proceeds, if possible. **ErrorOnly** indicates an error message is issued along with a return code of 8 and the job is allowed to proceed, if possible. **FatalError** indicates an error message is issued along with a return code of 8, but the job is stopped. The types of resources checked are font resources (CharSet, CodedFont, CodePage, and OpenTypeFonts), FormDef, Overlay, PageDef, PageSeg, and JPEG/TIFF resources specified in object containers.



Note: When FormDef resources fail to load or are corrupted, the engine automatically attempts to use the default FormDef instead to continue processing your job. However, the default FormDef may not be suitable for your project so your job can still fail.

4.1.1.6 OptimizedAfpOptions

The **OptimizedAfpOptions** parameter contains options related to processing optimized AFP files. The following parameters can be configured.

- **FdOptimizedAfp**. Specifies the full file path location of the optimized AFP input file to be parsed. If a file name is not set, no optimized AFP processing occurs and the input comes from the FdInput parameter instead.
- **DocumentOffset**. Designates the offset in the file identifying the start of document processing.
- **DocumentLength**. Designates the length, in bytes, of the document to process. If the length value is 0, then no optimized AFP processing takes place and the input comes from FdInput the parameter instead.
- **MaxAfpDocumentBufferMB**. Indicates the maximum size, in megabytes, that decompressed AFP documents can be when held in memory. When a document exceeds this value, they are decompressed using a slower method to conserve memory and avoid out of memory errors.

4.1.2 AFP Parser Wizard

The following sections describe the AFP Parser Wizard.

4.1.2.1 Launching the AFP Parser Wizard

The AFP Parser has a wizard associated with it to help you assemble the resources and establish the parameters for your project. The AFP Parser wizard launches when you:

- Create a new project that uses AFP input. The **New Component Wizard** launches the AFP Parser Wizard if the **Auto Start Parser Wizard** option box is checked.
- Double-click the AFP Parser on the **Project** tab.



Note: For more information, see “[New Component Wizard](#)” on page 95 and “[Project Tab](#)” on page 106.

The AFP Parser Wizard guides you through the steps to identify your input file and assemble your resources. You are then able to select AFP input options that will be written to your project file. In this way, helps you build the project file as you create a project.

The AFP Parser Wizard steps are:

1. Identify Your Parser.
2. Select AFP Form and Page Definition.
3. Select AFP Record Format.
4. Select Additional Settings.

If you choose not to make any selections in the wizard, and instead step through the wizard to the end, the project opens with a basic set of default values. In order for your project to run correctly, you will need to enter valid data where necessary in the project files and component tools.

4.1.2.1.1 Identify Your Parser

Parser Name

Enter a name for your parser or accept the default of AFP Parser.



Note: It is not possible to have more than one parser running in a project.

Model parser on AFP Parser in the project file

Click the ellipse button to browse to an existing Output Transformation Engine project file that you want to use as a model to set your starting AFP Parser parameters.

Leave this field blank to create a simple project file with default AFP Parser values. You can make changes to any of the parameters later in the Parameters window.

Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point while creating a new parser, you will need to enter your parser parameters before you can run the project.

4.1.2.1.2 Select AFP Form and Page Definition

In this step, you confirm the location, file extensions, and categorizations of the AFP resources that have been found in the project directory you specified. You can override the location of any file. You can add files or directories as required. Output Transformation Designer can accept a mixture of paths for your resources, so having resources in different directories, even across a network, is not a problem.

Review each path name to ensure that it identifies the correct file.

You may not require entries in all of the fields, depending on the intricacies of your input:

- *<ACIF Resources.>* Advanced Conversion and Indexing Facility is an IBM utility that collects resource files (fonts, overlays, pagedefs, formdefs) and combines them into a single file. The ACIF may:
 - Put resources in the same file as the input data, inline; if so, the program automatically sees the information.
 - Put the resources in a separate resource file; if so, you need to set the FDacif parameter to the full path & file name for that file (such as FDACIF = 'c:\applications\test\acif.res').

When you have all the resources together in a resource file, either inline in the data or in a separate resource file, you do not need to specify any of the other resource parameters: FDafpfonts, FDoverlays, FDPagedefs, FDformdefs, FDPagesegs, FormDef, or PageDef.

An entry in this field sets the FDACIF parameter in the project file.



Note: A plus (+) on the folder indicates that multiple files may be specified in this field. The project appends additional files to any existing entries in the field. Where multiple files have been found, but only one can be utilized, you can choose the file you want. Output Transformation Designer prompts you to select the file you want to use.

- **AFP Fonts.** AFP font resources include Coded Fonts, Character Sets, and Code Pages, which are abbreviated as CodedFont, CharSet, and CodePage. A font reference within AFP is accomplished through a single CodedFont reference, or by referencing the CharSet and CodePage directly. When a CodedFont is referenced, that resource determines the CharSet and the CodePage. This field sets the FDAFPFonts parameter in the project file.
- **Form Definitions.** A FormDef resource is almost always required. Form Definition resources define the size of the page, the initial fonts being used, and the overlays that are automatically included on the page. This is accomplished

with Medium Map definitions. This field sets the FormDef parameter in the project file. If the page is a standard 8.5"x11" and the AFP input file contains all font definitions and resource calls, the FormDef is not required.

- **Input.** Identifies where your AFP input file is to be found.
- **Overlays.** overlays are AFP electronic forms, which can be referenced by the AFPDS or included inline within the AFPDS. This field sets the FDOverlays parameter in the project file.
- **Page Definitions.** Page definition resources define how the line data is placed on the page using one or more Data Maps. This field sets the associated parameter in the project file.
- **Page Segments.** Page segments are one form of graphics files to be printed on AFP printers. Page Segments do not have any compression and must be monochrome. Page segments can be raster graphics that stand alone, or which contain GOCA or IOCA objects. This field sets the FDPPageSegs parameter in the project file.
- **Res Group Out.** Sets the output resource group path and file name. Use with ResGroupOptions to determine which resources will be included in the file.
- **FdExportOverlays.** Specifies the location for extracted overlays.
- **FdExportPageSegs.** Specifies the location for extracted page segments.
- **FdExportFonts.** Specifies the location for extracted fonts.
- **FdExportFormDefs.** Specifies the location for extracted form definitions.
- **FdExportPageDefs.** Specifies the location for extracted page definitions.
- **FdStats.** Specifies the location of the optional stats file. When this parameter is present, the parser/generator will write out the tuning stats to the file. For more information, see ["Interpreting the FdStats File" on page 64.](#)

Click **Next** to go to the next step.

Click **Back** to change the resource directory you want to scan.

Click **Cancel** to exit the wizard.

4.1.2.1.3 Select AFP Record Format

The default settings are AFP default values. If you have a standard AFP input file, or you are unsure whether you require any specific record format parameters, leave the AFP Default box checked. You can change these parameters later in the parameters window if necessary.

- **AFP Default.** AFP Default format tells the program that each record in the data starts with a hex '5A' followed by a 2-byte inclusive length field in MSB/LSB format (no special characters at the end of the record). MSB/LSB format refers to Most Significant Bit/Least Significant Bit, meaning the data bits that indicate the beginning or end of a record.

- **Record Format.** These options establish additional parameters for your input file:
- **BDW.** This I/O Directive indicates that the records have a length field and are grouped into blocks, which also have a length field. This is equivalent to MVS Variable Block data sets, recfm=VB. The length fields for both the block and the records are four bytes, with the first two bytes representing the length. The last two bytes are ignored.
- **RDW.** This I/O Directive indicates that the records have a four-byte length field. This is equivalent to MVS Variable data sets, recfm=V. The length fields for the records are four bytes, with the first two bytes representing the length. The last two bytes are ignored.
- **MVS FTP.** Parameter is `FDinput = {MVSFTP}'\path\input.ext'` It tells the program that the data was FTPed from an MVS system.
- **CRLF.** This I/O Directive indicates that these records are delimited by a CRLF, which is the 2-byte hex value of 0d0a. The hex value 0a actually is used as the delimiter and is not considered as part of the data. If the hex value 0a (LF) is preceded by the hex value 0d (CR), neither the CR nor the LF are considered part of the data.
- **Length Field (LF).** Designated with `FDinput = {LF#x}'\path\input.ext'`, it tells the program that each record starts with some type of length field. This is not complete in itself. Start with {LF, then select the options appropriate to your data:

Whether records are inclusive or exclusive:

- Inclusive
- Exclusive
- Unknown

Then the record length

- **2** - Designating a 2-byte
- **4** - Designating a 4-byte
- Unknown

The next character is:

- **M (MSB / LSB)** - Length field value is in hex with Most significant byte first
- **L (LSB / MSB)** - Length field value is in hex with Least significant byte first
- **A** - ASCII decimal
- **E** - EBCDIC decimal
- Unknown
- **Fixed Length (F).** Line data is usually fixed length, with no length fields or record end indicators. This is designated with the parameter `FDinput = {F#}'\path\input.ext'` where the # is the number of bytes in each record, such as {F132}.

- **Line Data.** Line data records are formatted with a and formdef. Line data does not contain the x,y coordinates for the text, or the names of the fonts to use. That is set in the separate file. Line data does not contain page size or orientation. They are set in the formdef.
- **Fixed Length.** Is enabled when the Fixed Length radio button (under Record Format section) has been selected. Enter the Record Length or accept the default of 255.
- **Content records begin with 5A.** It is not necessary to specify that the records begin with 5A if you have default input because most fully composed AFP records start with a hex '5A' (see AFP Default above). Some UNIX devices support AFP with no 5A, just the 2-byte length field. This gets designated as No5A. AFP with no 5A is not normal, and is not supported on MVS.

Click **Next** to go to the next step.

Click **Back** to change any of your resource information.

Click **Cancel** to exit the wizard.

4.1.2.1.4 Select Additional Settings

This final step sets additional page parameters.

- **Page Range.** Provides options for the range of pages to be processed. This is useful for testing your implementation. Instead of processing an entire large file, you can process only a few pages to determine if your settings are correct. When a range is selected, only those pages appear in the output:
- **Start Page.** Specifies the beginning of a range of pages to print. The numeric value for Start Page must be less than or equal to Stop Page, except for the Stop Page default value of zero (0). Zero is an invalid value for the Start Page parameter.
- **Stop Page.** Specifies the end of a range of pages to print. The numeric value for Stop Page must be greater than that of Start Page (except for stoppage=0). The default value is zero (0), meaning all pages are processed.



Note: The page range values can also be changed in the **parser Properties**, or its associated **wizard**. The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Properties, despite what may be displayed in the Properties. Similarly, if you change these values in the Properties, the new values will override what's entered in the Job Options tab of the Execute Options form until a change is made to either the Job Options tab or the Properties.

- **<Page Size.>** Sets the parameters for the default size of your output paper. The AFP Parser will use this value only when the paper size cannot be determined via a Form Definition. Choose a standard paper size from the drop-down box.

Your choices are **Letter**, **Legal**, **A4**, **Custom**, **Ledger**, **A3**, **Executive**, **Monarch**, **COM10**, **DL**, **C5**, or **B5**.

Custom lets you define your own paper size by specifying the following:

- **Width.** (X size) Enter the width of your output. By default, it is set to 8.5 inches or standard portrait letter size.
- **Height.** (Y Size) Enter the height of your output. By default, it is set to 11 inches, or standard portrait letter size.
- **Inches or Centimeters** determines whether the measurements specified for width and height are in inches or centimeters.



Note: Changes to the Width, Height, and/or Units parameters will have no effect unless the page size selection is set to Custom.

- **AFP Characteristics.** Indicates typical AFP characteristics:
- **TRC.** Select this if the input AFP file contains font Table Reference Characters.
- **Carriage Control File.** Select this if the file has carriage control bytes.
- **Form/Page Definitions.** If you did not indicate Form or Page Definition resources in Step 2, you may do so now. For more information, see [“Select AFP Form and Page Definition” on page 139](#).
- **Start Page Definition.** Click the folder icon to browse to the location of your start page definition file.
- **Start Form Definition.** Click the folder icon to browse to the location of your start form definition file.

Click **Finish**, or click **Back** to change any of your previous settings.

4.1.3 Parsing AFP Data Utilizing OpenType Fonts

The AFP Parser is capable of parsing MOD:CA and line data that uses OpenType fonts, a superset of TrueType fonts. This extension of the AFP architecture contains some fundamental technical differences compared to classic AFP raster and AFP outline font mechanisms.

The primary difference is that, unlike AFP raster and AFP outline fonts, OpenType fonts are not referred to in the AFP printstream by their filename. For an AFP outline font, for example, the printstream might indicate the character set (charset) named CZJHMNU and the codepage named T11200 be used to represent certain presentation text (PTX) data. In the traditional context of the printer or AFP Parser, this means there are two physical files accessible: one named CZJHMNU and one named T11200. These two files are then parsed to obtain the font metrics required to process the text. However, for OpenType fonts, the resources are not referred to by filename; they are instead referred to by typeface name, such as Times New Roman instead of `times.ttf`. This key difference means that the OpenType typeface must be resolved into the real font name before any text data can be processed.

For IBM AFP print environments, the Resource Allocation Table (RAT) manages the resolution of the font file name from the typeface name. IBM supplies a toolkit of utilities to enable users to graphically import OpenType fonts and construct the RAT.



Note: See the IBM publication *Using OpenType Fonts in an AFP System* (G544-5876) for more information.

Integrating the AFP Parser with the Font Environment Map uses a model similar to the IBM RAT. For more information, see [“Font Environment Map” on page 565](#).

When the AFP Parser encounters MOD:CA or line data that references OpenType fonts, it attempts to resolve the font filename in the Font Environment Map using the typeface name supplied in the AFP printstream.

It is therefore essential that you are aware of the OpenType fonts your printstream uses, and ensure that these OpenType fonts are properly imported and cataloged in the Font Environment Map associated with the project.

As with all font types used by Parser components, OpenType fonts used by the AFP Parser utilize the font mapping mechanism of the Master Font Correlation Table (MFCT). Creating a font mapping for an OpenType font is virtually the same process as creating a font mapping for an AFP charset or an AFP typeface.

4.2 Image Parser



The Image Parser is used to process and convert various image formats. The parser can handle TIFF, BMP, JPG, GIF, and PNG files.

Features

Features of the Image Parser include:

- Multi-Page. Any numbers of pages.
- Right-to-Left and Left-to-Right Bit Fill orders.
- Multi-Strip, supporting any number of data strips per page.
- Photometric Interpretation with both normal and inverse video formats.
- Compression Formats Supported: None, CCITT T.4 (Group 3), CCITT T.6 (Group 4), Packbits, LZW and JPG (JPEG in TIFF 6).
- Output page size is automatically calculated based on the image size.
- Indexing supported through the use of comment elements and metadata elements.
- Fully supported in Output Transformation Designer, including in the Page View tab.
- Ability to process multiple input documents. For more information, see [“I/O Directives” on page 118](#).

Supported Color Depth of the Output Transformation Engine Image Parser include:

- Black and white, 1 bit per pixel images.
- 2 color, 1 bit per pixel with palette.
- 16 color, 4 bits per pixel with palette.
- 256 color, 8 bits per pixel with palette.
- RGB color 24 bits per pixel.
- CMYK color 32 bits per pixel.

Known Limitations

- LZW, CCITT Group 3 -2d zero fill and JPEG in TIFF7 compression formats are not currently supported.
- Page size selection; for example, A4 page size input to letter page size output.
- Anisotropic DPI is not supported, for example, when the horizontal DPI is not the same as the vertical DPI.
- Tiled images.

4.2.1 Image Parser Parameters

The Image Parser parameters are used with the Image Parser. A default set of parameters and values is used when you add the Image Parser to your project. When you work through the **Image Parser Wizard**, you develop parameters that are specific to your project. For more information, see [“Image Parser Wizard” on page 146](#).



Note: If you make changes directly in the Properties Window, remember to tab or click out of edit mode to ensure the changes are captured.

To ensure that you have all of the settings you need, we recommend you review the parser parameters in the **Properties window** before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties window** and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

Depending on the image file types you are working with, you must be aware that the **PageWidth** and **PageHeight** parameters behave differently. For TIFF image types, these two parameters specify the page width and height in inches. However, for all other supported image types, the **PageWidth** and **PageHeight** parameters actually indicate the image width and height in inches while the **Width** and **Height** parameters located under **SizingOptions > Size** are used to indicate the page width

and height. (By default, the Width and Height parameters are disabled so you must **Enable** them for use.)

If you are using the Image Parser to resize images, you must also bear in mind the image's file type as only non-TIFF images can be resized. Additionally, for non-TIFF images, if you try to specify an image size that is larger than the page size then the image size is used for the page size. If you are specifying an image size that is larger than the page size for a TIFF image, this results in the image size being preserved, but truncated in order to fit the page size.

4.2.1.1 Setting Page Range Parameters

Page range values can be changed in the Properties window, in its associated wizard, or in the Job Options window of the Job Execution Options form.

The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Properties window. Similarly, if you change these values in the Parameters, the new values will override any previous settings until a change is made to either the Job Options tab or the Parameters. Changes made in the Properties window will be displayed in the Job Options window, but not vice-versa.

4.2.2 Image Parser Wizard

4.2.2.1 Launching the Image Parser Wizard

The Image Parser has a wizard associated with it to help you assemble the resources and establish the parameters for your project. The Image Parser Wizard launches when you:

- Create a new project that uses images as input. The New Component Wizard launches the Image Parser Wizard if the Auto Start Parser Wizard option box is checked.
- Double-click the Image Parser on the Project tab .

The Image Parser Wizard guides you through the steps to identify your input file and assemble your resources. You are then able to select image input options that will be written to your project file. In this way, helps you build the project file as you create a project.

The Image Parser Wizard steps are as follows:

1. Identify your image parser.
2. Select image input details.
3. Select property settings.

If you choose not to make any selections in the wizard, and instead step through the wizard to the end, the project opens with a basic set of default values. In order for

your project to run correctly, you will need to enter valid data where necessary in the project files and component tools.

4.2.2.1.1 Identify Your Image Parser

The following fields are shown on the **Image Parser Wizard – 1 of 3** screen:

- **Parser Name.** Enter a name for your parser or accept the default Image Parser.



Note: It is not possible to have more than one parser running in the same project.

- **Model parser on Image Parser in the project file.** Click the ellipsis button to browse to an existing Output Transformation Engine project file that you want to use as a model to set your starting Image Parser parameters.

Alternatively, you can leave this field blank to create a simple project file with default Image Parser values. You can make changes to any of the parameters later in the **Properties window**.

Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point while creating a new parser, you will need to enter your parser parameters before you can run the project.

4.2.2.1.2 Select Image Input Details

Confirm the location and file extensions of the image resources.

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see [“Select Resource Path/File Window” on page 102.](#)

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

4.2.2.1.3 Select Property Settings

This final step allows you to specify the start and stop pages.

- **Start page.** Specifies the beginning of a range of pages to print. The numeric value for the Start page must be less than or equal to the Stop page, except for the Stop page default value of zero (0). Zero is an invalid value for the Start page parameter.
- **Stop page.** Specifies the end of a range of pages to print. The numeric value for the Stop page must be greater than that of the Start page (except for Stop page=0). The default value is zero (0), meaning all pages are processed.



Note: The page range values can also be changed in the parser Properties, or its associated wizard. The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Properties, despite what may be displayed in the Properties. Similarly, if you change these values in the Properties, the new values will override what's entered in the Job Options tab of the Execute Options form until a change is made to either the Job Options tab or the Properties.

Click **Finish**, or click **Back** to change any of your previous settings.

4.3 Linedata Parser



The Linedata parser is used to parse unstructured text-based printstreams. The input contains no formatting, only one font and size, and no graphical elements. The print commands essentially have a start and end point with only ASCII (or EBCDIC) text in between. The input may include simple elements like line feeds or page breaks.

Once parsing is complete, the parsed data can be treated like any other parsed data in . It can be transformed or indexed using any of the available components, such as Constant Print, or Best Viewed Rotation, and then formatted for output by any generator.



Note: The Linedata Parser requires a carriage control byte. Output can still be generated from your input through form feeds and CRLFs if a CC byte is not present, but be aware that the results may not be completely accurate. In these situations, switching the I/O directive from {CRLF} to {METABAR} may help with generating better output. For more information about working with the Linedata Parser and {METABAR}, see “{METABAR}” on page 122.

4.3.1 Linedata Parser Parameters

The following parameters are available:

- **StartPage.** Default is 1 (first page). If you specify a page outside of the available page range, you will get an error message that there are no pages to parse.
- **StopPage.** Default is 0 (last page). Specify a negative number to offset from the last page. For example, -1 is the second-last page. If you specify a page outside of the available page range, you will get an error message that there are no pages to parse.
- **FdInput.** Specifies the path to the input file to be parsed.
- **Tuning.** The default tuning setup is ideal for most systems.
- **PreviewFontAndPageSizes.** If selected, the parser will produce a series of preview pages (one for every entry in the PageSetupArray parameter) at the beginning of the process. These pages are intended for testing/proofing purposes and will contain sample text strings using the font settings defined below. These pages will be contained in their own Xif Document, so you must include the wildcard replacement variable in the generator's FdOutput parameter (for example, output-{0}.pdf).
- **PrinterOptions.** Controls the shift-out and shift-in (SOSI) mechanism for double-byte processing.
- **FontNamesAndAtts.** Defines the name and the vertical/horizontal spacing values for the font.
 - **OpenTypeFontName.** Click the ellipsis to change the font used for the document. Enter an accessible font name and save your changes. The default is Courier New.
 - **Encoding.** Specifies the encoding of the input file. Available options are **AUTO_DETECT** (the Linedata Parser will scan the document and automatically detect whether the encoding is ASCII or EBCDIC), **ASCII**, **EBCDIC** and **ICU_TABLE**. The default value is **AUTO_DETECT**.
 - **IcuTableID.** When the **Encoding** parameter is set to **ICU_TABLE**, this value specifies the table ID to use.
 - **PointsizeToCharsPerInch.** Determines the horizontal spacing in the document. A larger number increases the spacing. The default value is 600.
 - **PointsizeToLinesPerInch.** Determines the vertical spacing in the document. A larger number increases the line height. The default value is 900.
 - **HorizontalScaleFactor.** Defines the horizontal scale factor that adjusts the width of the text characters to make them wider or narrower. The default value of 1.0 keeps the scale factor at 100%, which is the font's default appearance. Increasing the value to 1.1 increases the default character width to 110%. Lowering the value to 0.9 condenses the characters to 90% of their original width. A common use of this parameter is to condense the font width to allow for more text to be printed on the page.

- **ShiftOutFontNameAndAtts.** Defines the name and the vertical/horizontal spacing values when using a double-byte (shift-out) font.
 - **OpenTypeFontName.** Click the ellipsis to change the font used for the document. Enter an accessible font name and save your changes. The default is Courier New.
 - **Encoding.** Specifies the encoding of the input file. Available options are **AUTO_DETECT** (the Linedata Parser will scan the document and automatically detect whether the encoding is ASCII or EBCDIC), **ASCII**, **EBCDIC** and **ICU_TABLE**. The default value is **AUTO_DETECT**.
 - **IcuTableID.** When the **Encoding** parameter is set to **ICU_TABLE**, this value specifies the table ID to use.
 - **PointsizeToCharsPerInch.** Determines the horizontal spacing in the document. A larger number increases the spacing. The default value is 600.
 - **PointsizeToLinesPerInch.** Determines the vertical spacing in the document. A larger number increases the line height. The default value is 900.
 - **HorizontalScaleFactor.** Defines the horizontal scale factor that adjusts the width of the text characters to make them wider or narrower. The default value of 1.0 keeps the scale factor at 100%, which is the font's default appearance. Increasing the value to 1.1 increases the default character width to 110%. Lowering the value to 0.9 condenses the characters to 90% of their original width. A common use of this parameter is to condense the font width to allow for more text to be printed on the page.
- **PageSetupArray.** The page setup default is optimized for printing standard letter-size documents. Make changes based on your desired output.
 - **PreferPortrait.** Select to use portrait orientation. Clear for landscape.
 - **PageWidth.** Default is 8.5 inches for a standard letter-size document.
 - **PageHeight.** Default is 11 inches for a standard letter-size document.
 - **TopMargin.** Default is .5 inches.
 - **BottomMargin.** Default is .5 inches.
 - **PointSizes.** Defines the allowable point sizes to use for text elements. A single value can be entered to apply the same point size on all pages. Alternatively, the value can be a list of point sizes to use with the values separated by a semicolon and the largest desired point size listed first, then descending to the smallest acceptable point size. For example, "12;10;8;6" is a possible valid value. Note that using a list of point sizes may result in a different point size used on each page of the document.
- **CarriageControlType.** Specifies the type of Carriage Control (CC). You can choose from **AUTO_DETECT**, **ANSI**, **MACHINE**, or **ASCII**. There are three types of CC most commonly used in the production printing business: **ANSI**, **MACHINE**, and **ASCII**. **ANSI** is the most universal carriage control, which contains a single prefix character denoting how each line is printed. **MACHINE** is the native carriage control used on IBM mainframe channel attached line

printers. ASCII uses FF (0x0C) for a page break, and CR (0x0D), LF (0x0A), or CRLF (0x0D 0x0A) to end a line. Use {ASCII} IO Directive on FdInput when CarriageControlType is ASCII. The AUTO_DETECT option automatically detects the type of carriage control between MACHINE and ANSI based on the first type used in the document. However, for text documents that have no carriage controls, it is recommended to use ASCII since the AUTO_DETECT setting may not produce the desired results.

- **SuppressBlankPages.** Determines whether all blank pages are suppressed. Blank pages are generally caused by extra form feed carriage control bytes in the line data file. When set to false, all blank pages are preserved in the document.
- **LineBreaking.** Denotes whether a long line of text can be broken into multiple lines so that it fits into the available width of a page. By default, this setting is false.
- **UseBaselineTopMargin.** Denotes whether the baseline of the first line of text on the page is used as the baseline for calculating the top margin. When the parameter is false, the ascender line (an imaginary line representing the distance between the baseline and the top of an ascender character) is used as the baseline for calculating the top margin. Enabling this setting can help to improve text positioning inside the page margins.



Note: For more information about some of these settings, see:

- [“Performance Tuning Assumptions” on page 60](#)
- [“Using Double-Byte Characters with the Linedata Parser” on page 154](#)
- [“Using OpenType Fonts with the Linedata Parser” on page 151](#)

4.3.2 Using OpenType Fonts with the Linedata Parser

To set up an OpenType font mapping to use with the Linedata Parser:

1. Import a TrueType font into the Font Environment Map.
2. Add the TrueType font as an OpenType font in the Linedata Parser properties.
3. Add the new OpenType font in the MFCT and map it to an output font.

4.3.2.1 Step 1: Importing the TrueType Font into the Font Environment Map

The imported font will be used as the OpenType font in your project.

In the **Catalog Manager** tab of the Font Environment Map window, do one of the following:

- Click **Import Folder** to import an entire folder containing fonts, or
- Click **Import Wizard** to import a specific font.

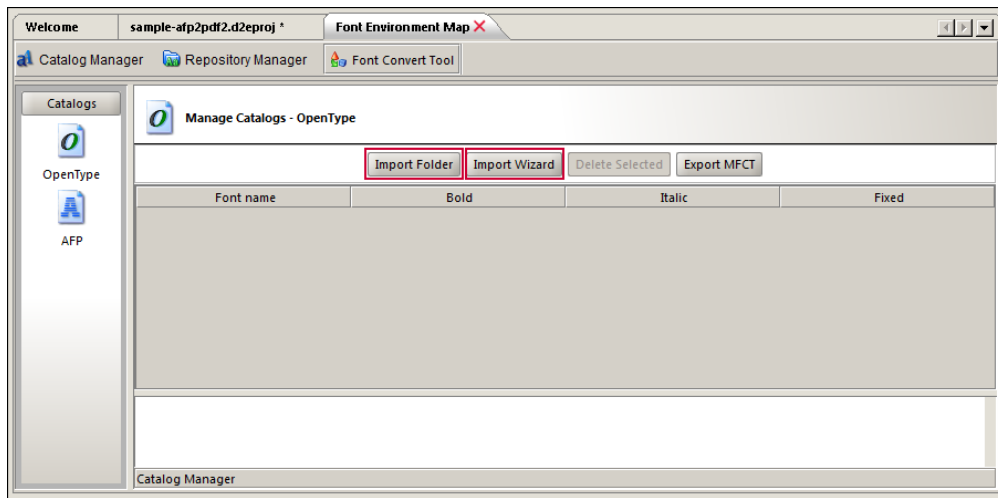


Figure 4-1: Catalog Manager tab

When you are done, the imported font(s) will populate the Catalog Table.

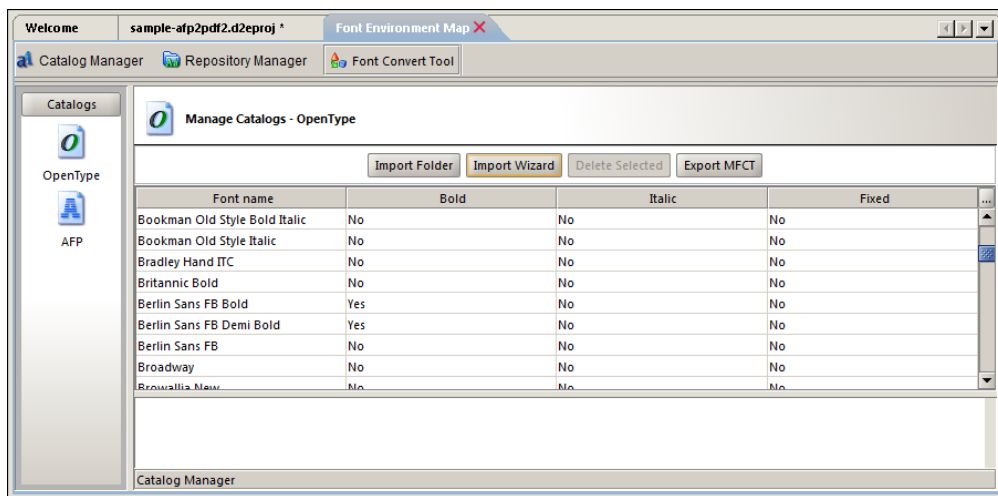


Figure 4-2: Catalog table with fields populated

4.3.2.2 Step 2: Adding the TrueType Font as an OpenType Font in the Linedata Parser Properties

1. In the Linedata Parser **Properties** window, expand the **FontNamesAndAtts** parameter and click the ellipsis beside the **OpenTypeFontName** setting.
The **Property - OpenTypeFontName** dialog opens.
2. Enter the name of the OpenType font that you imported into the **Font Environment Map** in Step 1.

 **Note:** You must use the same font name exactly as it appears in the Font Environment Map's **Catalog Table**.

3. Click **OK** to save your changes.

The OpenType font is added to the Linedata Parser properties.

4.3.2.3 Step 3: Adding the New OpenType Font in the MFCT and Mapping it to an Output Font

1. Open the MFCT for your project and click **Add** in the MFCT toolbar.
The **Add OpenType Font** dialog opens.

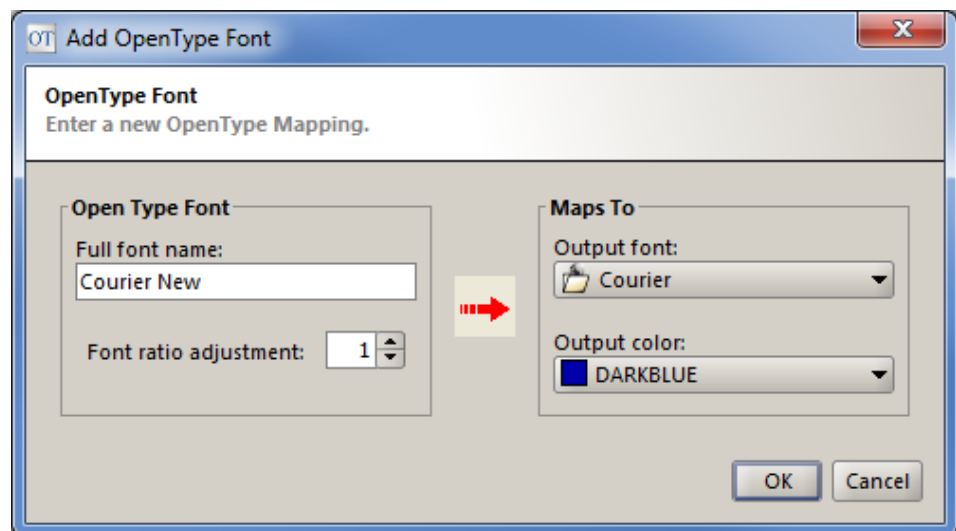


Figure 4-3: Add OpenType Font dialog

2. Complete the following details:
 - **Full font name.** The entire OpenType font name, entered exactly as you specified it in the Font Environment Map and in the Linedata Parser properties.
 - **Font ratio adjustment.** The multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font.

Use a value of less than 1 if your output documents have overlapping text, or a value greater than 1 if the text in your output documents appear smaller than desired.

- **Output font.** The output font to which the OpenType font will be mapped.
 - **Output color.** The color of the output font.
3. Click **OK** to save your changes.

The OpenType font has been added to the MFCT and is mapped to the desired output font.

Related Links

- [“Master Font Correlation Table” on page 502](#)
- [“MFCT Toolbar” on page 508](#)

4.3.3 Using Double-Byte Characters with the Linedata Parser

To set up the Linedata Parser to process double-byte characters:

1. Under **PrinterOptions**, select the **PrinterMode**.

The **PrinterOptions** parameter is used to control the shift-out and shift-in (SOSI) mechanism for double-byte processing. The **PrinterMode** setting allows you to select the SOSI printing mode. SOSI controls delimit the beginning and end of a string of double-byte code points and are not printable characters. Select one of the following options to determine how the SOSI controls are handled in the text output:

- **None.** No SOSI printing mode will be applied. This is the default value and should be selected if not using double-byte characters.
- **Sosi1.** Each shift-out byte (0x0e) is replaced with a blank (0x40) followed by a font change to a double-byte font affecting subsequent bytes. Each shift-in byte (0x0f) causes a font change to the single-byte font, and then the byte is replaced with a blank (0x40).
- **Sosi2.** Each shift-out byte (0x0e) causes a font change to a double-byte font affecting subsequent bytes. Each shift-in byte (0x0f) causes a font change to a single-byte font affecting subsequent bytes.
- **Sosi3.** Each shift-out byte (0x0e) causes a font change to a double-byte font affecting subsequent bytes. Each shift-in byte (0x0f) causes a font change to a single-byte font affecting subsequent bytes and the 0x0f is replaced with two blank 0x40s that will be printed in the single-byte new font.



Note: Choose the appropriate printing mode that corresponds to your data in order to have the correct text alignment. If your text is not aligned properly, you may need to try a different mode.

2. Use the **FontNameAndAtts** options to define the name and the vertical/horizontal spacing values for the font. For double-byte characters:

- Under **Encoding**, select **ICU_TABLE**. This option allows you to specify a standard codepage (character set to Unicode mapping) to match the encoding of your line data.
 - For **IcuTableID**, enter the numeric ID of the ICU table that will be used to determine the Unicode values for your data. These tables are shipped with .
3. Use the **ShiftOutFontNameAndAtts** options to define the name and the vertical/horizontal spacing values for the double-byte (shift-out) font:
- **OpenTypeFontName**. Click the ellipsis to change the font used for the document. Enter an accessible font name and save your changes. The default value is Courier New. For more information, see [“Using OpenType Fonts with the Linedata Parser” on page 151](#).
 - **Encoding**. Specifies the encoding of the input file. Available options are AUTO_DETECT (the Linedata Parser will scan the document and automatically detect whether the encoding is ASCII or EBCDIC), ASCII, EBCDIC and ICU_TABLE. The default value is AUTO_DETECT.
 - **IcuTableID**. When the Encoding parameter is set to ICU_TABLE, this value specifies the table ID to use.
 - **PointsizeToCharsPerInch**. Determines the horizontal spacing in the document. A larger number increases the spacing. The default value is 600.
 - **PointsizeToLinesPerInch**. Determines the vertical spacing in the document. A larger number increases the line height. The default value is 900.

4.4 Metacode Parser



The Metacode Parser is designed to process and convert Xerox Metacode and LCDS, often referred to as DJDE. Line data and mixed mode files that print on Xerox Centralized printers are supported, as well as pure Metacode files. The Metacode Parser outputs to XIF for subsequent processing by generators.

The Metacode Parser provides transformation and repurposing of mission-critical Metacode print applications for the Web and print center, without having to change the legacy applications that create the Metacode.

The Metacode Parser is ideal for migrating high-volume printer applications to other high-volume print output such as IBM AFP, as well as to content management archives for storage and for dynamic Web presentment, in PDF or HTML, for example.

Features

Features of the Metacode Parser include:

- Support for line data with JSL, PDEs and CMEs.
- Support for LCDS, also known as DJDE.

- PDEs can be internal to the JSL, or external in binary format. allows PDEs and CMEs to be in JSL files that are preloaded with a parameter.
- Support for external CMEs.
- Support for font mapping or rasterization at a font or individual character level.
- Support for intelligently parsing Xerox shading and transforming it to AFP shading commands, rather than image output.
- Ability to specify paper stocks by name and paper size.
- Processing of duplex pages.
- Support for RFEED for paper tray selection.
- Support for DJDE FILE=() for Inline Resources.
- Comments that come from DJDEs, Hex 03 CCs, and OTEXT commands are placed into XIF and their selection can be controlled through parameters.

4.4.1 Metacode Parser Parameters

The Metacode Parser parameters are used with the Metacode Parser. A default set of parameters and values is used when you add the Metacode Parser to your project. When you work through the “[Metacode Parser Wizard](#)” on page 157, you develop parameters that are specific to your project.



Note: If you make changes directly in the Properties window, remember to tab or click out of Edit mode to ensure the changes are captured.

To ensure that you have all of the settings you need, we recommend you review the Parser parameters in the **Properties window** before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

Setting Page Range Parameters

Page range values can be changed in the **Properties** window, in its associated wizard, or in the Job Options window of the Job Execution Options form.

The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Properties window. Similarly, if you change these values in the Parameters, the new values will override any previous settings until a change is made to either the Job Options tab or the Parameters. Changes made in the **Properties** window will be displayed in the **Job Options** window, but not vice-versa.

4.4.2 Metacode Parser Wizard

4.4.2.1 Launching the Metacode Parser Wizard

The Metacode Parser has a wizard associated with it to help you assemble the resources and establish the parameters for your project. The Metacode Parser wizard launches when you:

- Create a new project that uses Metacode input. The **New Component Wizard** launches the Metacode Parser Wizard if the **Auto Start Parser Wizard** option box is checked.
- Double-click the Metacode Parser on the **Project** tab.



Note: For more information, see “[New Component Wizard](#)” on page 95 and “[Project Tab](#)” on page 106

The Metacode Parser Wizard guides you through the steps to identify your input file and assemble your resources. You are then able to select Metacode input options that will be written to your project file. In this way, helps you build the project file as you create a project.

The Metacode Parser Wizard steps are:

1. Identify Your Parser.
2. Select Metacode Input Details.
3. Select Property Settings.

If you choose not to make any selections in the wizard, and instead step through the wizard to the end, the project opens with a basic set of default values. In order for your project to run correctly, you will need to enter valid data where necessary in the project files and component tools.

4.4.2.1.1 Identify Your Parser

The following fields are shown on the **Metacode Parser Wizard – 1 of 3** screen:

- **Parser name.** Enter a name for your parser or accept the default of Metacode Parser.



Note: You cannot have more than one parser running in a project.

- **Model parser on Metacode Parser in the project file.**
 1. Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting Metacode Parser parameters. Leave this field blank to create a simple project file with default Metacode Parser values. You can make changes to any of the parameters later in the **Properties window**.

2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point while creating a new parser, you will need to enter your parser parameters before you can run the project.

4.4.2.1.2 Select Metacode Input Details

Confirm the location and file extensions of the Metacode resources.

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

4.4.2.1.3 Select Property Settings

This final step sets the properties of the parser:

- **Page Range.** Provides options for the range of pages to be processed. This is useful for testing your implementation. Instead of processing an entire large file, you can process only a few pages to determine if your settings are correct. When a range is selected, only those pages appear in the output.
- **Start.** Specifies the beginning of a range of pages to print. The numeric value for the Start page must be less than or equal to the Stop page, except for the Stop page default value of zero (0). Zero is an invalid value for the Start page parameter.
- **Stop.** Specifies the end of a range of pages to print. The numeric value for the Stop page must be greater than that of the Start page (except for Stop page=0). The default value is zero (0), meaning all pages are processed.



Note: The page range values can also be changed in the parser Properties, or its associated wizard. The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Properties, despite what may be displayed in the Properties. Similarly, if you change these values in the Properties, the new values will override what’s entered in the Job

Options tab of the Execute Options form until a change is made to either the Job Options tab or the Properties.

- **Start Commands.** Specify start command typed on printer for this job:
- **JDE.** Enter the name for the JDE.
- **JDL.** Enter the name for the JDL.
- **Carriage Control.** When selected, indicates whether the input file contains carriage control characters.
- **Shading.** Select the type of shading from the drop-down box: Light or Dark.

Click **Finish**, or click **Back** to change any of your previous settings.

4.5 PCL Parser



The Output Transformation Engine PCL Parser is designed to parse HP PCL input files to XIF for subsequent processing by generators or other components.

The PCL Parser, together with other Output Transformation Engine components, provides indexing of mission-critical PCL print documents for loading content management and archive applications.

The PCL Parser can be used for transformation of PCL files to other printstream formats, such as AFP for high volume printing, subject to prior verification of the PCL files and the specific output requirement.

Care should be taken with those customers wishing to parse PCL from Windows printer drivers, as this is highly complex and is subject to continual change from Microsoft and the printer driver manufacturers.

4.5.1 Supported Formats

HP PCL 5 and PCL 6 (PCL XL), including those generated by d2e Platform and Output Transformation Engine PCL Generators.

Features

Features of the Output Transformation Engine PCL Parser include:

- Support for raster PCL graphics.
- Support for embedded soft fonts (BMP (Runlength compressed and uncompressed)) and True Type (TTF).
- Support for PCL inline soft fonts specified by attributes, rather than font ID.
- Support for transformation of embedded soft fonts to embedded PDF True Type fonts (only with the PDF Generator).
- Support for color image (excluded indexed color; Runlength, TIFF compressed and uncompressed), color text, shades and vectors.

- Support for CCITT Group4 compressed images.
- Support for PCL macros/overlays and external PCL macros/overlays.
- Support for custom paper source size.
- Support for adjustable intercharacter spacing through rasterization.
- Support for HPGL with Point Factor scaling.
- Support for user-defined symbol sets.
- Support for page, image, and text rotation.
- Support for PCL 5 user-defined fill.

4.5.2 PCL Parser Parameters

The PCL Parser parameters are used with the PCL Parser. A default set of parameters and values is used when you add the PCL Parser to your project. When you work through the “[PCL Parser Wizard](#)” on page 161, you develop parameters that are specific to your project.



Note: If you make changes directly in the Properties window, remember to tab or click out of edit mode to ensure the changes are captured.

To ensure that you have all of the settings you need, we recommend you review the parser parameters in the Properties window before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties window** and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

4.5.2.1 Setting Page Range Parameters

Page range values can be changed in the **Properties** window, in its associated wizard, or in the Job Options window of the Job Execution Options form.

The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Properties Window. Similarly, if you change these values in the Parameters, the new values will override any previous settings until a change is made to either the Job Options tab or the Parameters. Changes made in the **Properties** window will be displayed in the **Job Options** window, but not vice-versa.

4.5.3 PCL Parser Wizard

4.5.3.1 Launching the PCL Parser Wizard

The PCL Parser has a wizard associated with it to help you assemble the resources and establish the parameters for your project. The PCL Parser Wizard launches when you:

- Create a new project that uses PCL input. The **New Component Wizard** launches the PCL Parser Wizard if the **Auto Start Parser Wizard** option box is checked.
- Double-click the PCL parser on the **Project** tab.



Note: For more information, see “[New Component Wizard](#)” on page 95 and “[Project Tab](#)” on page 106.

The PCL Parser Wizard guides you through the steps to identify your input file and assemble your resources. You are then able to select PCL input options that will be written to your project file. In this way, helps you build the project file as you create a project.

The PCL Parser Wizard steps are:

1. Identify Your Parser.
2. Select PCL Input Details.
3. Set Page Attributes.

If you choose not to make any selections in the wizard, and instead step through the wizard to the end, the project opens with a basic set of default values. In order for your project to run correctly, you will need to enter valid data where necessary in the project files and component tools.

4.5.3.1.1 Identify Your Parser

The following fields are shown on the **PCL Parser Wizard – 1 of 3** screen:

- **Parser name.** Enter a name for your parser or accept the default of PCL Parser.



Note: You cannot have more than one parser running in a project.

- **Model parser on PCL Parser in the project file.** Click the ellipsis button to browse to an existing Output Transformation Engine project file that you want to use as a model to set your starting PCL Parser parameters.

Leave this field blank to create a simple project file with default PCL Parser values. You can make changes to any of the parameters later in the Properties window.

Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point while creating a new parser, you will need to enter your parser parameters before you can run the project.

4.5.3.1.2 Select PCL Input Details

Confirm the location and file extensions of the PCL resources.

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.
 - Click **Remove** to remove the resource path.



Note: For more information, see [“Select Resource Path/File Window” on page 102.](#)

4.5.3.1.3 Set Page Attributes

This final step sets the page setup attributes:

- **Rotate for orientation.** Select this check box to rotate non-portrait pages based on orientation. For example, an 8.5 × 11 landscape page would be rotated so the XifPage is 11 X 8.5 with portrait text. This feature should be turned off in some cases, usually when generating output for print, to preserve certain orientations, such as tumble print.
- **Ignore White Text.** Select this check box to ensure that text in a white color is ignored and not converted into XIF elements.
- **Enabled AdvancedCoordSys.** Select this check box to enable the AdvancedCoordSys rotation and scaling settings.

Click **Finish**, or click **Back** to change any of your previous settings.

4.6 PDF Parser



The PDF Parser is designed to parse Acrobat PDF input files to XIF for subsequent processing by generators or other components.

The PDF Parser, together with other components, provides indexing of mission-critical PDF print applications for loading content management and archive applications.

The PDF Parser can be used for transformation of PDF files to printstream formats such as AFP, for high volume printing; subject to prior verification of the PDF files and the specific output requirement.

4.6.1 Supported Formats

Adobe PDF 1.0 to 1.6 files (Acrobat 7.0 or earlier), including those generated by d2e Platform and PDF Generators.

Features

Features of the PDF Parser include:

- Vector graphics support (pie charts, histograms, etc.)
- Images (image objects, logos, etc.)
- True color (24-bit) RGB images supported in PDF output and in AFP output with Function Set 11 RGB color support.
- Decoding Flate, LZW and Ascii85 encoded compressed objects.
- Processing 40-bit and 128-bit encrypted files assuming the Owner and User passwords are provided. Decoding Flate, LZW and ASCII85 encoded compressed objects.
- Ability to process multiple input documents. For more information, see *“I/O Directives” on page 118*.
- Support for parsing tagged PDF documents. For more information, see *“PreserveTags” on page 165*.
- **Bullets.** Supported, with font correlation using glyph mapping, if there is corresponding output font; otherwise it uses the default font.
- **Word shadows.** Supported with font correlation; if two colors they will be slightly offset.
- **Tables.** Supported for graphical tables, i.e. row/column data inside line.

Supported fonts:

- Standard Adobe Base 14
- Type 1 (no rasterization)

- Type 3 (no rasterization)
- TrueType (no rasterization)
- CJK (Chinese, Japanese, Korean) font support for double-byte and multi-byte character fonts

Text orientations:

- 0° - Portrait
- 90° - Inverse Landscape
- 180° - Inverse Portrait
- 270° - Landscape

4.6.2 PDF Parser Parameters

The PDF Parser contains a default set of parameters and values that are used when you add the PDF Parser to your project. When you work through the **PDF Parser Wizard**, you develop parameters that are specific to your project.



Note: If you make changes directly in the Properties window, remember to tab or click out of Edit mode to ensure the changes are captured.

To ensure that you have all of the settings you need, we recommend you review the parser parameters in the **Properties window** before running your Output Transformation Engine project.

The following PDF Parser parameters are discussed in detail in this guide:

- PreserveTags
- PreserveInteractiveForm
- ProcessDigitalSignature



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

Related Links

- [“PDF Parser Wizard” on page 166](#)
- [“PreserveTags” on page 165](#)
- [“PreserveInteractiveForm” on page 166](#)
- [“ProcessDigitalSignatures” on page 166](#)

4.6.2.1 Setting Page Range Parameters

Page range values can be changed in the **Properties** window, in its associated wizard, or in the Job Options window of the Job Execution Options form.

The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Properties window. Similarly, if you change these values in the Parameters, the new values will override any previous settings until a change is made to either the Job Options tab or the Parameters. Changes made in the **Properties** window will be displayed in the **Job Options** window, but not vice-versa.

4.6.2.2 PreserveTags

When enabled, the **PreserveTags** parameter allows the PDF Parser to preserve PDF tags, marked-content operators, and accessibility text spacing information when parsing a PDF document. The logical structure of the input PDF can then be used by the Accessible PDF Generator to create an accessible PDF document.

When the PreserveTags option is disabled (which is the default setting), tags and marked-content operators are ignored by the PDF Parser and the output PDF will not contain logical structure information.

For optimal results, the following settings should also be applied in conjunction with the PreserveTags option when generating an accessible PDF:

- In the PDF Parser properties, select the **TransferResources** check box to ensure that all text is displayed properly in the accessible PDF output.
- In the Accessible PDF Generator properties:
 - Clear the **FlattenForms** check box when the accessible PDF logical structure is generated by the PDF Parser. However, you should select this check box when the Page Markup component defines the logical structure.
 - Select the **CompressObject** option to optimize the output PDF size.

Related Links

- [“Accessible PDF Generator” on page 174](#)
- [“Page Markup Component” on page 295](#)

4.6.2.3 PreserveInteractiveForm

The **PreserveInteractiveForm** parameter allows fillable forms from an input file to be preserved and created in output files from the Accessible PDF and PDF Generators. You must also enable **ProcessAnnotations** as a prerequisite for PreserveInteractiveForm to be recognized, otherwise it is ignored. When enabled, the form object is parsed as an interactive object and if the parameter is disabled, the objects are flattened.

4.6.2.4 ProcessDigitalSignatures

When enabled, the **ProcessDigitalSignatures** parameter parses and processes PDF files that contain a digital signature. Any output PDFs produced from the signed PDF will not contain a digital signature. When this parameter is false, input PDF files containing a digital signature are rejected, then the job fails with a detailed error message and no output is created.

4.6.3 PDF Parser Wizard

4.6.3.1 Launching the PDF Parser Wizard

The PDF Parser has a wizard associated with it to help you assemble the resources and establish the parameters for your project. The PDF Parser wizard launches when you:

- Create a new project that uses PDF input. The **New Component Wizard** launches the **PDF Parser Wizard** if the **Auto Start Parser Wizard** option box is checked.
- Double-click the PDF Parser on the **Project** tab.



Note: For more information, see “[New Component Wizard](#)” on page 95 and “[Project Tab](#)” on page 106.

The PDF Parser Wizard guides you through the steps to identify your input file and assemble your resources. You are then able to select PDF input options that will be written to your project file. In this way, helps you build the project file as you create a project.

The PDF Parser Wizard steps are:

1. Identify Your Parser.
2. Select PDF Input Details.
3. Select Property Settings.

If you choose not to make any selections in the wizard, and instead step through the wizard to the end, the project opens with a basic set of default values. In order for your project to run correctly, you will need to enter valid data where necessary in the project files and component tools.

4.6.3.1.1 Identify Your Parser

The following fields are shown on the **PDF Parser Wizard – 1 of 3** screen:

- **Parser name.** Enter a name for your parser or accept the default of PDF Parser.



Note: You cannot have more than one parser running in a project.

- **Model parser on PDF Parser in the project file.**

1. Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting PDF Parser parameters. Leave this field blank to create a simple project file with default PDF Parser values. You can make changes to any of the parameters later in the Parameters window.
2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point while creating a new parser, you will need to enter your parser parameters before you can run the project.

4.6.3.1.2 Select PDF Input Details

Confirm the location and file extensions of the PDF resources.

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

4.6.3.1.3 Select Property Settings

This final step sets the page range for processing:

- *<Page Range.>* Provides options for the range of pages to be processed. This is useful for testing your implementation. Instead of processing an entire large file, you can process only a few pages to determine if your settings are correct. When a range is selected, only those pages appear in the output.
- *<Start page.>* Specifies the beginning of a range of pages to print. The numeric value for the Start page must be less than or equal to the Stop page, except for the Stop page default value of zero (0). Zero is an invalid value for the Start page parameter.
- *<Stop page.>* Specifies the end of a range of pages to print. The numeric value for the Stop page must be greater than that of the Start page (except for Stop page=0). The default value is zero (0), meaning all pages are processed.



Note: The page range values can also be changed in the parser Properties, or its associated wizard. The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Properties, despite what may be displayed in the Properties. Similarly, if you change these values in the Properties, the new values will override what's entered in the Job Options tab of the Execute Options form until a change is made to either the Job Options tab or the Properties.

Click **Finish**, or click **Back** to change any of your previous settings.

4.7 TIFF Parser



The Output Transformation Engine TIFF Parser is designed to process and convert images with the Adobe TIFF Revision 6.0 specification base line image. This specification includes the older Revision 3, 4 and 5 specifications including:

- Tiff Class B - bi-level images
- Tiff Class G - grayscale images
- Tiff Class P - palette based images (also known as pseudo color)
- Tiff Class R - RGB based images

In addition to the base line TIFF specifications, the following TIFF extensions are supported:

- Multiple pages per file
- CCITT bi-level encoding
- LZW compression

- Differencing predictors
- CMYK images
- JPEG compression
- Flate compression (also known as zip-in-tiff)

Applications of the TIFF Parser include repurposing of documents stored in image archive systems by converting them to PDF, PNG, or other formats for Web presentment in e-business applications or to Metacode for production print.

Features

Features of the Output Transformation Engine TIFF Parser include:

- Multi-Page. Any numbers of pages.
- Right-to-Left and Left-to-Right Bit Fill orders.
- Multi-Strip, supporting any number of data strips per page.
- Photometric Interpretation with both normal and inverse video formats.
- Compression Formats Supported: None, CCITT T.4 (Group 3), CCITT T.6 (Group 4), Packbits, LZW, and JPG (JPEG in TIFF 6).
- Output page size is automatically calculated based on the image size.
- Indexing supported through the use of comment elements and metadata elements.
- Fully supported in Output Transformation Designer, including in the Page View tab. For more information, see [“Page View Tab” on page 90](#).
- Ability to process multiple input documents. For more information, see [“I/O Directives” on page 118](#).

Supported Color Depth of the Output Transformation Engine TIFF Parser include:

- Black and white, 1 bit per pixel images.
- 2 color, 1 bit per pixel with palette.
- 16 color, 4 bits per pixel with palette.
- 256 color, 8 bits per pixel with palette.
- RGB color 24 bits per pixel.
- CMYK color 32 bits per pixel.

Known Limitations

- LZW, CCITT Group 3 -2d zero fill and JPEG in TIFF7 compression formats are not currently supported.
- Page size selection; for example, A4 page size input to letter page size output.

- Anisotropic DPI is not supported, for example, when the horizontal DPI is not the same as the vertical DPI.
- Tiled images.

4.7.1 TIFF Parser Parameters

The TIFF Parser parameters are used with the TIFF Parser. A default set of parameters and values is used when you add the TIFF Parser to your project.



Note: If you make changes directly in the Properties window, remember to tab or click out of Edit mode to ensure the changes are captured.

To ensure that you have all of the settings you need, we recommend you review the parser parameters in the **Properties window** before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties window** and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

4.7.1.1 Setting Page Range Parameters

Page range values can be changed in the **Properties** window, in its associated wizard, or in the Job Options window of the Job Execution Options form.

The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Properties window. Similarly, if you change these values in the Parameters, the new values will override any previous settings until a change is made to either the Job Options tab or the Parameters. Changes made in the **Properties** window will be displayed in the **Job Options** window, but not vice-versa.

4.7.2 TIFF Parser Wizard

4.7.2.1 Launching the TIFF Parser Wizard

The TIFF Parser has a wizard associated with it to help you assemble the resources and establish the parameters for your project. The TIFF Parser Wizard launches when you:

- Create a new project that uses TIFF input. The **New Component Wizard** launches the **TIFF Parser Wizard** if the **Auto Start Parser Wizard** option box is checked.
- Double-click the TIFF Parser on the Project tab.



Note: For more information, see “**New Component Wizard**” on page 95 and “**Project Tab**” on page 106.

The TIFF Parser Wizard guides you through the steps to identify your input file and assemble your resources. You are then able to select TIFF input options that will be written to your project file. In this way, helps you build the project file as you create a project.

The TIFF Parser Wizard steps are as follows:

1. Identify Your TIFF Parser.
2. Select TIFF Input Details.
3. Select Property Settings.

If you choose not to make any selections in the wizard, and instead step through the wizard to the end, the project opens with a basic set of default values. In order for your project to run correctly, you will need to enter valid data where necessary in the project files and component tools.

4.7.2.1.1 Identify Your Parser

The following fields are shown on the **TIFF Parser Wizard – 1 of 3** screen:

- **Parser name.** Enter a name for your parser or accept the default of TIFF Parser.



Note: It is not possible to have more than one parser running in the same project.

- **Model parser on TIFF Parser in the project file.**
 1. Click the ellipsis button to browse to an existing project file that you want to use as a model to set your starting TIFF Parser parameters.
 2. Leave this field blank to create a simple project file with default TIFF Parser values. You can make changes to any of the parameters later in the **Properties window**.
 3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point while creating a new parser, you will need to enter your parser parameters before you can run the project.

4.7.2.1.2 Select TIFF Input Details

Confirm the location and file extensions of the TIFF resources.

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see [“Select Resource Path/File Window” on page 102](#).

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

4.7.2.1.3 Select Property Settings

This final step allows you to specify the start and stop pages.

- *<Start page.>* Specifies the beginning of a range of pages to print. The numeric value for the Start page must be less than or equal to the Stop page, except for the Stop page default value of zero (0). Zero is an invalid value for the Start page parameter.
- *<Stop page.>* Specifies the end of a range of pages to print. The numeric value for the Stop page must be greater than that of the Start page (except for Stop page=0). The default value is zero (0), meaning all pages are processed.



Note: The page range values can also be changed in the parser Properties, or its associated wizard. The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Properties, despite what may be displayed in the Properties. Similarly, if you change these values in the Properties, the new values will override what's entered in the Job Options tab of the Execute Options form until a change is made to either the Job Options tab or the Properties.

Click **Finish**, or click **Back** to change any of your previous settings.

Chapter 5

Generators

The Output Transformation Engine generators use the XIF data that was created by the parser and process flow components to create output in the desired particular format.

In order to execute, generators require that you configure data and appearance translation rules, which match or correlate the fonts used in the input file with fonts used to generate the output file.

Output Transformation Engine Generators

The generators available with Output Transformation Engine include:

- Accessible PDF Generator. Generates accessible PDFs, marked up for use with screen reader software.
- AFP Generator. Generates AFP MO:DCA output, and supports all future parsers.
- Image Generator. Generates single page color image output files. Currently supported formats are PNG, TIFF, and JPEG.
- Metacode Generator. Generates Metacode/DJDE (Dynamic Job Descriptor Entry) output from the XIF.
- PCL Generator. Generates PCL5 output from the XIF.
- PDF Generator. Generates Adobe PDF 1.7 output. Supports standard Acrobat 128-bit security for document fraud protection.
- PDF/A Generator. Generates Adobe PDF output that is compliant with the PDF/A-1b (ISO 19005-1:2005) standard.
- PostScript Generator. Generates PostScript Level 3, suitable for printing on any PostScript Level 2 printer, from the XIF.
- TIFF Generator. Generates single or TIFF Class B output files. Supports CCITT Group4 compression and any resolution (dpi).
- Web Generator. Generates HTML output with CSS.



Note: You will have access to these generators based on the permissions granted in your license file.

Generator Wizards

All generators have wizards associated with them to help you define their parameters. Whether or not you use a Generator Wizard to create your generator project file, please review your parameters to ensure you have established all of the required parameters for your Output Transformation Engine project.

Related Links

- [“Wizards” on page 93](#)

5.1 Accessible PDF Generator



The Accessible PDF Generator converts the XIF data that has been logically laid out using the Page Markup component into accessible PDFs. Accessible PDFs are PDFs that have been created with readable text and tags, or markup data, which guide the screen reader application logically through the document. This ensures that the document is not just logically and graphically well designed for sighted users, but that it is easily read in a logical manner by a screen reader application, giving users with visual disabilities a similarly positive experience.

Accessible PDFs are created with a Output Transformation Engine project which includes a parser, the XFT Field Technology component, the Page Markup component, and the Accessible PDF Generator. It is the combination of these components that work synergistically to create a final accessible PDF, easily used by anyone.



Note: Due to performance issues that may cause some undesirable results in your projects, the Accessible PDF Generator is automatically set to run exclusively in single-threaded mode to maintain accurate Accessibility output. An INFO-level message is written to the logs stating this when Accessibility features are active.

Related Links

- [“XFT Field Technology” on page 341](#)
- [“Page Markup Component” on page 295](#)

5.1.1 Using the Accessible PDF Generator

The Accessible PDF Generator supports the same features as the PDF Generator, but contains several additional parameters geared towards accessibility functions. You can configure the Accessible PDF Generator using the instructions for the PDF Generator wizard and when finished, you can manually add the following settings in the Accessible PDF Generator properties:

- **Accessible.** Denotes whether documents are generated as tagged PDFs when tagged content (configured by using the Page Markup component) is available. By default, this is enabled.
- **DocumentLanguage.** Indicates the default language for the entire document as defined by RFC 4646 language subtag registry. This information is critical to the screen reader application.
- **TextFromXftResults.** Determines whether the logical reading order of text objects is derived from XFT results or the graphical drawing order (original order defined by source document). Sometimes the original organization of text in the

source document may conflict with the order required by the screen readers. XFT has many features to sort the text fields based on position and allows for variance in base line offsets of text so it can be useful for the generator to organize the logical order of text based on these results. When the parameter is disabled, the logical order of the text comes from the order of XifTextElements on your pages and forms. By default, this is disabled.

- **EmbedAllFonts.** Denotes whether all fonts are embedded into your output PDFs. By default, this is enabled to satisfy the PDF/UA specifications.
- **AutoArtifact.** Determines whether untagged content is automatically tagged as an Artifact. By default, this is enabled.
- **ReportUntaggedContent.** Designates whether tags not explicitly configured with the Page Markup component should be identified as untagged content and reported in the error log. In the logs, these messages are labeled as UntaggedContent and hold the WARNING log level. The reporting of untagged content helps to ensure that elements are not automatically tagged as artifacts without the user being made aware. Consequently, if any elements are automatically tagged as artifacts in your input document templates, you can update your configuration to establish the correct PDF tags.
- **TagTitle.** Indicates how PDF tags have optional titles applied to them. These titles are descriptive names that can be viewed next to the tags in Adobe Acrobat's tag tree view and provide context about the tags to the viewer of the PDF, but do not provide any accessibility-related benefits for screen readers or other assistive technologies. The default value of **Enabled** results in the XFT field names being associated with the tag elements and are displayed as the tag titles. The **Disabled** value excludes the titles and is especially useful when PDF file sizes need to be minimized. The **Debug** value provides descriptive information that includes some unique numeric identifiers for the tagged page elements. Generally, the Debug option is only used by project developers for troubleshooting purposes and is not intended for use in production environments.
- **TagStructureOutput.** Contains settings related to the accessible tag structure log file, which displays the tag structure of your input document. This log can be particularly useful when project or product updates are made to quickly compare differences between the new tag structure and a previously saved tag structure.
- **Enabled.** Designates whether an accessible tag log file showing the tag structure of your input document is generated, with one log being established for each PDF created. When enabling the tag structure log, you must also enter a valid value for FdTagStructure to specify where the file should be written. Enabling the log in production environments is not recommended due to its negative impact on performance.
- **FdTagStructure.** Indicates where the accessible tag structure log file is written. Additionally, you must also select the Enabled parameter. For jobs that create multiple PDFs, to help with differentiating between logs you can use {0} in your parameter value to insert the document name as part of the tag structure log file name.

Related Links

- [“PDF Generator” on page 203](#)
- [“PDF Generator Wizard” on page 211](#)

5.1.2 Defining Alternate Text

Accessible PDF documents often contain visual elements that screen reader technology cannot interpret. Alternate text must be provided for these elements so that screen readers can describe the visual content for visually impaired users. For example, instead of a logo’s path and file name being read by the screen reader, you can define “company logo” as the alternate text.

With the Accessible PDF Generator, you can use the **Define Images** tool to define alternate text for selected images in an accessible PDF. You then have the option to package the alternate text resources and send them to an administrator for editing and/or approval in the .

5.1.2.1 Defining and Packaging Alternate Text

To define and package alternate text:



Note: This functionality is available only for projects using the Accessible PDF Generator.

1. In Output Transformation Designer, launch the **Page View** tab from either the parser or the Accessible PDF Generator. For more information, see [“Page View Tab” on page 90](#).
2. Click **Define Images**, , at the top right corner of the Page View toolbar. The **Define Images** dialog appears.

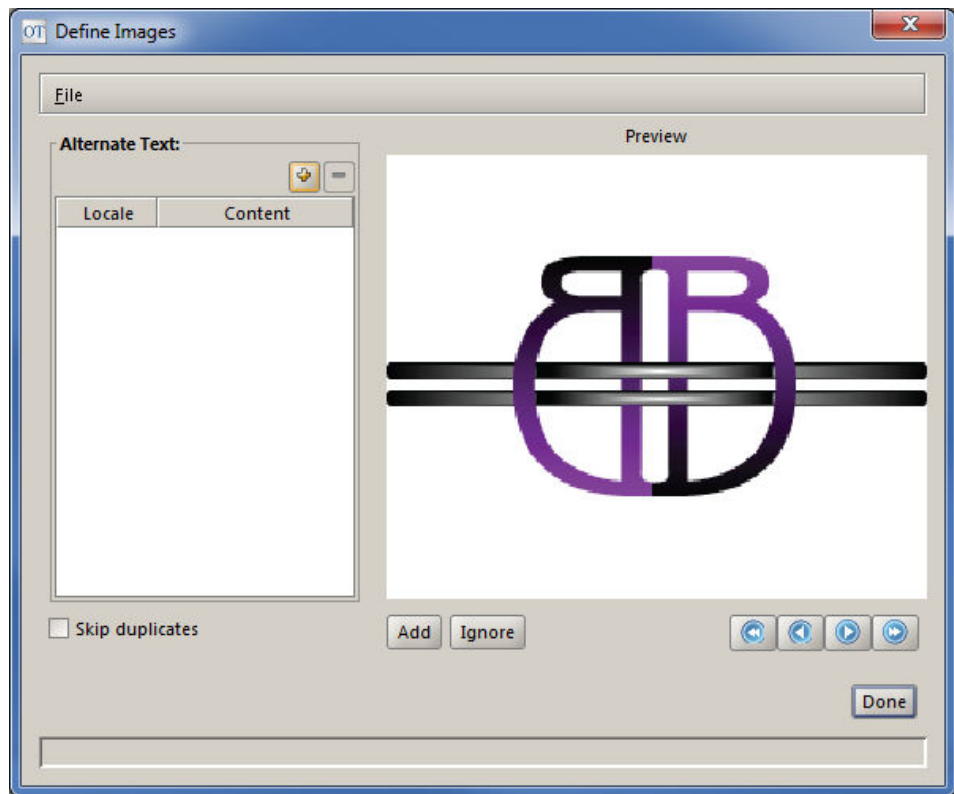


Figure 5-1: Define Images dialog

3. Images contained in the document are displayed one at a time in the **Preview** window. Use the navigation buttons below the Preview window to cycle sequentially through each image in the order that they appear on each page.
 -  Returns to the first image in the document.
 -  Returns to the previous image.
 -  Proceeds to the next image.
 -  Skips to the last image in the document.
4. To add alternate text for the selected image, click **Add a new item** () and enter the following:
 - **Locale.** Determines the language for the alternate text. Select a locale from the drop down list or enter the appropriate language code. This field is mandatory.
 - **Content.** Enter the alternate text to be used by the screen reader. This field accepts up to 2000 alphanumeric characters and/or symbols.



Note: When entering alternate text, be sure to keep the following in mind:

- You can add multiple entries for the same image. This may be useful when defining alternate text for more than one locale.
- Do not define alternate text for non-static images (such as charts and barcodes that are customized for each customer) as the alternate text will not be propagated to the final document.
- The alternate text that you define for an image will automatically apply to all occurrences of the same image in the document.
- To remove an alternate text entry, highlight the entry and click **Remove selected item** ()

5. When you have finished defining alternate text for the image, select the **Skip duplicates** check box to bypass subsequent occurrences of the same image in the document, then click **Add**, .

The added image is denoted by a green check mark in the Preview window. Additionally, a message is displayed at the bottom of the dialog window, confirming that the image and alternate text were added to the package.



Note: If you do not want an image to be added to the package, click **Ignore** () to bypass the image. This button is active only when the **Skip duplicates** check box is selected. An ignored image will not be added to the package, and is denoted by a blue downward arrow in the Preview window.

6. When you are ready to package the completed images, click **File** and select **Save** from the context menu.

The images and alternate text are packaged and saved as an ARP (.arp) file in the location that you define. Additionally, the package's file name and location are now displayed at the top of the **Define Images** window.

7. Click **Done** to exit the Define Images tool. If you have not already saved the package, you will be prompted to do so.

5.1.2.2 Editing an Existing Alternate Text Package

To make changes to an existing alternate text package:

1. In Output Transformation Designer, launch the **Page View** tab from either the parser or the Accessible PDF Generator. For more information, see "[Page View Tab](#)" on page 90.
2. Click **Define Images** () at the top right corner of the Page View toolbar. The **Define Images** dialog appears.
3. Select **File > Open**, browse to the location of the alternate text package you wish to edit, and click **Open**.

You can now edit alternate text entries, or add new images to the package.

4. For each modification you make, click **Add** to add your changes to the package.

When you have finished, do one of the following:

- Select **File > Save** to incorporate your modifications in the package using the same file name. Note that this will overwrite the previous version of the package.
- Select **File > Save As** to save as a separate package under a different file name.

5. Click **Done** to exit the Define Images tool.

5.1.2.3 Testing the Output for Accessibility

Once the Accessible PDF has been generated, you can view the output in Adobe Acrobat.

To check the PDF for accessibility:

1. Open the PDF using Adobe Acrobat.
2. Choose **Advanced > Accessibility > Full Check**.

To check that the reading order you arranged using the Page Markup component is correct:

1. Open the PDF using Adobe Acrobat.
2. Choose **Advanced > Accessibility > TouchUp Reading Order**.

For more information about using Adobe Acrobat, see your Adobe Acrobat documentation.

5.1.3 Preserving Invisible Text in PDF Documents

Occasionally, there are instances where certain content should be available to users of screen reader technologies to improve accessibility, but concealed from sighted users. For example, while sighted users can view the screen and discern visual features, a particular section may require some additional tagged content specifically for visually-impaired users if there isn't sufficient information for them to understand it. This supplementary text can be hidden from sighted users to maintain the document's visual design since the information is visually apparent to them without it.

The **PreserveInvisibleText** parameter, which is also available in the PDF and PDF/A generators, maintains any invisible text from your input documents in your accessible PDF output. When enabled, the **PreserveInvisibleText** parameter preserves all invisible text elements in the output PDF with the text rendering mode set to invisible. If this parameter is disabled, all invisible text elements are ignored by the generator. By default, this parameter is enabled.

Related Topics

- [“PDF Generator” on page 203](#)
- [“PDF/A Generator” on page 217](#)

5.2 AFP Generator



The AFP Generator component allows production of AFP output for indexing or print on AFP printers. It supports the following features:

- IBM MO:DCA (not mixed mode or line data).
- 240, 300, 1440 dpi.
- Customizable options for various IOCA formats.
- Reuse of incoming AFP resources (page segments and overlays).
- Collection and packaging of resources into inline resource groups or external ACIF file.
- Automatic discovery of font properties to determine appropriate encoding and MCF font lists.
- Flexible Open Type font support with Unicode (UCS-2) encoded data.

5.2.1 AFP Generator Parameters

The AFP Generator parameters are used with the AFP Generator.

A default set of parameters and values is used when you add a generator to your project. When you work through the AFP Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your project.

The following AFP Generator parameters are discussed in detail in this guide:

- [“AcifOptions Parameter” on page 181](#)
- [“DefineFeed Parameter” on page 182](#)
- [“GocaOptions Parameter” on page 184](#)
- [“ImageOptions Parameter” on page 186](#)



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the **message area** will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the AFP Generator Wizard, see [“AFP Generator Wizard” on page 189](#).

5.2.1.1 AcifOptions Parameter

This parameter enables you to configure ACIF resource generation as inline or external.

Parameter Elements

BuildAcif	Specifies whether to build an ACIF resource file.  Note: BuildAcif is ignored when BuildInlineResgroup is enabled. A separate ACIF file cannot be constructed when an inline resource group is enabled.
True	Indicates that an ACIF resource file is built during the transform.
False	Indicates that no ACIF resource file is generated.
BuildInlineResgroup	Specifies whether to create an inline resource group (resgroup) for all output documents. Such resource groups must physically occur at the beginning of the output file, before the first page of data. Because of this requirement, the AFP Generator must utilize a series of temporary files to buffer the resource group and the output documents. Once the AFP Generator finishes generating all of the data, it assembles the final output documents and the temporary files are deleted. This impacts the overall transform time, and requires additional physical memory (disk space) and virtual memory (RAM/Java Heap size). This functionality is provided for convenience to produce a single output file. However, if performance is a key concern in your process, you may wish to consider building a separate ACIF file and concatenate them in a post process step.  Note: When BuildInlineResgroup is enabled, the BuildAcif and FdAcifOutput parameters are ignored. A separate ACIF file cannot be constructed when an inline resource group is enabled.

True	Designates that an inline resource group (resgroup) for all output documents is created.
False	Designates that no inline resource group (resgroup) is created.
IncludeFonts	Specifies whether to include fonts in the ACIF resource file.
True	Indicates that fonts are included in the ACIF resource file.
False	Indicates that fonts are not included in the ACIF resource file.
ReblockResources	Enables the re-blocking of records from existing resources (such as overlays or page segments) which are not generated by the AFP Generator, but are imported into the ACIF/Resgroup.
FdAcifOutput	Specifies the filename of the ACIF resource file.  Note: FdAcifOut is ignored when BuildInlineResgroup is enabled. A separate ACIF file cannot be constructed when an inline resource group is enabled.
FormdefName	Specifies the default form name.

5.2.1.2 DefineFeed Parameter

This parameter is used to define settings which map the properties from the incoming XifPage (such as input printer tray name, page size, jog, and plex mode attributes) to output Medium Map names. The AFP Generator will issue IMM (Invoke Medium Map) commands in the print stream for all pages which match a DefineFeed entry's criteria. It is assumed that these Medium Maps are already defined in the existing target form definition that the user wishes to pair with the generated AFP document(s).

Parameter Elements

The **DefineFeed** parameter has the following sub-elements:

IntrayName	Enter the input tray name (such as the feed name from the Metacode Parser, or the paper source from the AFP Parser).

IntraySize	Enter the input paper size (the size of the XiffPage that the AFP Generator receives) which should be matched. If the IgnoreIntraySize option is selected, this setting will be ignored when resolving the output Medium Map.
PageType	Enter the page size. Available options are Letter, Legal, A4, Custom, Ledger, A3, Executive, Monarch, COM10, DL, C5, and B5.
XSize	Enter the width of the page in the units defined by the Units parameter. The default value is 8.5. This value is required only if the PageType option is set to Custom.
YSize	Enter the height of the page in the units defined by the Units parameter. The default value is 11.0. This value is required only if the PageType option is set to Custom.
Units	Enter the units of measurement. This option is used only when PageType is set to Custom. Valid values are Inches or Centimeters.
Jog	Specify whether the input page is physically offset when it comes off the printer. Having the stack of printer paper shift slightly in the output tray makes it easier to differentiate one print job from another.
Plex	Select the input plex mode. Valid values are:
Simplex	Prints only on one side of the page.
Duplex	Prints on both sides of the page.
Tumble	Prints on both sides of the page, but flips the page along the bottom edge so that the text on the opposite side is upside down.
OutMediumMap	Specify the AFP Medium Map to be used for the IntrayName , IntraySize , Jog , and Plex options. Any valid Medium Map name is accepted.
IgnoreIntraySize	Select this check box to ignore the IntraySize value when selecting the output Medium Map. This feature is useful when API components are used to programmatically set the tray name on XiffPages to trigger the correct Medium Map in the output document.

IgnoreJog	Specify whether the Jog setting is ignored during DefineFeed mapping to the Medium Map.

5.2.1.3 GocaOptions Parameter

This parameter allows users to customize how the AFP Generator transforms incoming XifVectorElements into GOCA objects. This customization is often required when the target consumer of the generated AFP (such as a printer or archiving application) does not support the full GOCA specification. Additionally, users may wish to maximize printing speeds by forcing the AFP Generator to generate less complex GOCA in exchange for some unused features (for example, no color).

Parameter Elements

The **GocaOptions** parameter has the following sub-elements:

ColorMode	Controls what types of color commands should be included in all GOCA objects generated. Some AFP printers will not support full color commands, so this parameter can be modified to suit the target device or application.
UseProcessColors	(Default) Indicates that the Set Process Color command should be used to preserve the color information. This option preserves the most detail; however, some older or monochrome printers do not support this command natively.
MapOcaByColorValue	Matches the input color to one of the standard named OCA colors by color value.
MapOcaByGrayValue	Matches the input color to one of the standard named OCA colors by comparing their converted grayscale values. Select this option when the AFP output file is only intended to be printed on a monochrome printer. If you need to view the AFP in color, the MapOcaByColorValue option may be more appropriate.
MapViaMfctGocaTable	Maps the input color to one of the 16 AFP OCA colors and one of 16 GOCA Fill IDs using the AFP GOCA Output Color/Pattern Map Table in the MFCT.

BlackWhiteGrayColorOnly	Specifies that only three standard OCA colors, black, white, and gray, should be used. The AFP Generator will determine which of these three colors to use for all color vector objects it receives. This is useful when converting from formats like PDF or PCL where the target device is a monochrome printer.
NoColorCommands	Indicates that no color commands will be issued.
MapMfctGocaColorTableDefaultMode	Controls what happens when ColorMode is set to MapViaMfctGocaTable but the table does not contain a mapping entry for a particular color or colors. If an AFP pattern symbol already exists in the incoming XifVectorElements, then it will be used as is, regardless of which option is selected below.
UseDefaultTableEntry	Uses the default color and pattern ID in the MFCT table.
MapOcaByColorValue	Uses the standard OCA color that best matches the input color by color value. The pattern ID will be randomly selected.
MapRandomOcaColor	(Default) Randomly chooses one of the 14 visible OCA colors (white, black, and color of medium are excluded), and one of the 14 OCA pattern symbols (white and solid fill are excluded).
PatternMapMode	Controls how non-AFP vector patterns are represented as GOCA patterns in the AFP output.
	 Note: This parameter is ignored when ColorMode is set to MapViaMfctGocaTable , which provides more robust control over the pattern IDs. It is also ignored when doing AFP to AFP transformations, since the vector objects being processed by the AFP Generator will have the original GOCA pattern ID attached to them.
None	(Default) Indicates that no GOCA patterns will be used.
UseDefaultPatternID	Uses the pattern ID specified in the DefaultPatternSymbolID option for all GOCA objects. This may not be useful in scenarios where a variety of patterns is desired.

MapRandomPatternID	Provides a variety of GOCA patterns in the output using a random assignment of pattern IDs to the incoming vectors.
DefaultPatternSymbolID	Specifies which symbol to use from the default pattern set. If None (the default value) is selected and AllowColorCommands is set to True, no pattern command is issued in default cases and the color will still be used. See the Set Pattern Symbol (GSPT) Order command in the IBM GOCA specification for more information.
BezierToLineSteps	Controls how many line segments to create per bezier curve. Because the GOCA architecture does not support native bezier curve operations, the AFP Generator must convert bezier objects (such as those created by the PDF Parser) into simple line draw operations. The default value is 100.

5.2.1.4 ImageOptions Parameter

This parameter enables you to customize how the AFP Generator processes image elements. There is a large variation in the level of support for AFP IOCA specification amongst the various printers and applications that process AFP documents. Additionally, you may wish to optimize AFP output to maximize printing speeds by forcing the AFP Generator to scale and/or convert the color spaces of images.

Example A

XYZ Corp. may be using Output Transformation Engine to generate color AFP statements from existing PDF documents (pdf2afp transformation). The PDF documents contain very large RGB images with a native resolution of 65 DPI. However, XYZ Corp plans to print these documents on an AFP Printer that has CMYK color (not RGB) and while it's capable of handling the 65 DPI images, this is a non-standard AFP resolution so the printing speed is affected. They do know that if the images are in 300 DPI (a standard resolution for their printer), they achieve 25% speed increase in printing. Therefore, they configure the AFP Generator's ImageOptions parameter to tell Output Transformation Engine to convert the RGB images to CMYK and to scale them from 65 DPI to 300 DPI.

Example B

Big Bank Inc. is preparing to perform an afp2afp transformation with their own custom Output Transformation Engine components used to insert cheque images into their customer's statements. They will then print the statements and mail them to their customers. The process that produces the cheque images provides the images as full color JPEG images but their high speed printer is not color ready and refuses to handle color images. Therefore, they need to configure Output

Transformation Engine to convert the images into grayscale IOCA objects in the AFP output.

Parameter Elements

The parameter has the following sub-elements:

PermittedImageFormats	This sub-element is used to specify the IOCA image formats the AFP Generator is able to produce. The purpose of this element is to specify the image formats supported by the target printer or application and force the conversion of any images not in the selected list to the closest match that preserves the most image detail. For example, if you wish all full color images to be converted to grayscale, you would disable FullColorRGB and FullColorCMYK but leave Grayscale enabled. Alternatively, if you disable Grayscale and FullColorRGB, the AFP Generator would convert all grayscale and RGB images into CMYK images. The image formats available are the following:
BlackAndWhite	When set to True, indicates that the AFP Generator is permitted to generate 1 Bit (bi-level) black and white images. When set to False, the AFP Generator will attempt to convert black and white 1 Bit images into another format such as Grayscale or FullColorRGB if those options are available. Note, however, that setting this value to false is NOT recommended as all known printers/application will support bi-level images.
SingleColor	When set to True, indicates that the AFP Generator is permitted to generate single color (1 Bit bi-level) images. When set to False, the AFP Generator will attempt to convert these 1 Bit images into another format that will preserve the color or shading, such as FullColorRGB. If FullColorRGB and FullColorCMYK are unavailable, Grayscale or lastly BlackAndWhite will be used (if permitted).

Grayscale	When set to True , indicates that the AFP Generator is permitted to generate grayscale images. When set to False , the AFP Generator will attempt to convert all grayscale images into another format such as FullColorRGB or FullColorCMYK to preserve detail (if those options are available). If full color is not an option but BlackAndWhite is enabled, grayscale images will be dithered into 1 Bit bi-level images. Note, however, that this may result in severe loss of image detail.
FullColorRGB	When set to True , indicates that the AFP Generator is permitted to generate full color RGB images. When set to False , the AFP Generator will attempt to convert any RGB images it receives by checking for another format option in this order: FullColorCMYK, Grayscale, SingleColor, BlackAndWhite.
FullColorCMYK	When set to True , indicates that the AFP Generator is permitted to generate full color CMYK images. When set to False , the AFP Generator will attempt to convert any CMYK images it receives by checking for another format option in this order: FullColorRGB, Grayscale, SingleColor, BlackAndWhite.
AllowNativeImageScaling	When set to True, this indicates that the AFP Generator can preserve images that have resolutions (DPI values) other than those specified in the ImageDpi parameter. It also enables the AFP Generator to preserve images that have different horizontal and vertical resolutions, such as anamorphic images, which are a common occurrence when converting documents from formats such as PDF. Some AFP printers and applications cannot handle images in these resolutions efficiently or at all. When set to False, the AFP Generator scales the image to a uniform horizontal/vertical DPI resolution based on the values specified in the ImageDpi / PreferredDpi list.
ArbitraryScaling	When set to True , this indicates that the AFP Generator can preserve images that have resolutions (DPI values) other than those that are specified in the ImageDpi parameter.

AnamorphicScaling	When set to True , this allows the AFP Generator to preserve images that have different horizontal and vertical resolutions (i.e. anamorphic images). When set to False , the AFP Generator will scale any images with different horizontal and vertical resolutions to a uniform resolution.

5.2.2 AFP Generator Wizard

5.2.2.1 Launching the AFP Generator Wizard

The AFP Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The AFP Generator wizard launches when you:

- Create a new project that produces AFP output. The **New Component Wizard** launches the AFP Generator Wizard if the **Auto Start Generator Wizard(s)** option box is selected.
- Double-click the AFP Generator on the **Project** tab.



Note: For more information, see “**New Component Wizard**” on page 95 or “**Project Tab**” on page 106.

The AFP Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select AFP output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the AFP Generator Wizard are:

1. Identify Your Generator.
2. Select AFP Output Details.
3. Select Property Settings.

5.2.2.1.1 Identify Your Generator

The following fields are shown on the **AFP Generator Wizard – 1 of 3** screen:

- **Generator name.** Enter a name for your generator or accept the default of AFP Generator.

Since it is possible to have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, for example, AFPGenerator2.

- **Model Generator on AFP Generator in the Project File.** Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting parameters.

Leave this field blank to create a simple project file with default AFP Generator values. You can make changes to any of the parameters later in the **Properties window**.

Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.2.2.1.2 Select AFP Output Details

The following properties must be configured on the **Select AFP Output Details – 2 of 3** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add, Remove, or Edit a Resource Path**

To add or modify a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

- Click **Remove** to remove the resource path.

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.2.2.1.3 Select Property Settings

This final step sets the properties of the output file:

- **Output Page Resolution.** Select the resolution from the drop-down list: 240, 300, 600, or 1440 DPI.
- **Form Mode.** Select the form mode: External or Inline.
- **Translation.** When selected, translation will not occur.
- **Record length.** Enter the maximum record length in the generated AFP output file.

Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

5.3 Data Generator



The Data Generator component permits the running of Data Transformation Engine, which is a data transformation tool that can transform structured data formats, from within an Output Transformation Engine project. Through the Data Generator, the Data Transformation Engine editor is available to accept Document Object Model (DOM) structured input from the XFT Field Technology component and map it to your desired output data format and structure.

5.4 Image Generator



The Image Generator allows production of single-page TIFF, PNG, or JPEG images. The Image Generator supports the following features:

- Single-page TIFF, PNG, or JPEG images, providing near-perfect fidelity but non-searchable output.
- Black and white, grayscale, or full color output.
- Fonts can be correlated to TrueType fonts installed on the system; supported in Java by AWT (Abstract Windows Toolkit).



Note: The Image Generator produces single-page images (PNG, TIFF, or JPEG), while the TIFF Generator will produce single-page or multi-page TIFF images. For more information, see [“TIFF Generator” on page 228](#).

5.4.1 Image Generator Parameters

The Image Generator parameters are used with the Image Generator.

A default set of parameters and values is used when you add a generator to your project. When you work through the Image Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the Image Generator Wizard, see [“Image Generator Wizard” on page 192](#).

5.4.2 Image Generator Wizard

5.4.2.1 Launching the Image Generator Wizard

The Image Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The Image Generator wizard launches when you:

- Create a new project that uses the Image Generator. The New Component Wizard launches the Image Generator Wizard if the **Auto Start Generator Wizard(s)** option box is checked.
- Double-click the Image Generator on the Project tab.

The Image Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the Image Generator Wizard are:

1. Identify Your Generator.
2. Select Image Output Details.
3. Select Property Settings.

5.4.2.1.1 Identify Your Generator

The following fields are shown on the **Image Generator Wizard – 1 of 3** screen:

- **Generator name.** Enter a name for your generator or accept the default of Image Generator.

Since it is possible to have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, for example, ImageGenerator2.

- **Model Generator on Image Generator in the Project File.** Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting parameters.

Leave this field blank to create a simple project file with default Image Generator values. You can make changes to any of the parameters later in the **Properties** window.

Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.4.2.1.2 Select Image Output Details

The following properties must be configured on the **Select Image Output Details – 2 of 3** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add or Edit a Resource Path**

To add or modify a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

- Click **Remove** to remove the resource path.
3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.4.2.1.3 Select Property Settings

This final step sets the output file type and the compression method:

1. **Output Type.** Select the output type from the drop-down list: TIFF, PNG, or JPEG.
2. **Compression.** For TIFF output select the compression method from the drop-down list: None, Group4, LZW, Packbits, or Flate.
3. Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

5.5 Metacode Generator



The **Metacode Generator** produces Metacode/DJDE (Dynamic Job Descriptor Entry) output from the . It provides integration with archival and Electronic Content Management systems.

When combined with the Metacode Parser, the Metacode Generator enables the creation of:

- Normalized Metacode/DJDE output and indexes for archiving as Metacode.
- Indexing and archiving of high volumes of Xerox Metacode/DJDE printstream documents.
- Subsequent dynamic retrieval and conversion to PDF on the fly and for reprint.

For more information, see [“Metacode Parser” on page 155](#).

5.5.1 Applications

The Metacode Generator adds value for archiving and content management applications and for repurposing documents for other e-delivery channels where the customer requirement is to store Metacode/DJDE. The Metacode Generator enables normalizing, indexing and archiving of high volumes of Xerox Metacode/DJDE printstream documents for subsequent dynamic retrieval and conversion to PDF on the fly and for Metacode/DJDE reprint.

5.5.2 Generated Formats

Xerox Metacode/DJDE printstream output.

5.5.3 Supported Features

The **Metacode Generator** supports the following features:

- Facilitates the creation of IORES bundles (.iores) to capture the resources referenced by the generated Metacode printstream. This includes fonts (.fnt), forms (.frm), images (.img), and resources (.jsl). The IORES file created by a Metacode to Metacode transform is very similar to an ACIF file created by the AFP Generator. Useful for IBM Content Manager OnDemand or other ECM archives.



Note: For more information on the AFP Generator, see [“AFP Generator” on page 180](#).

- Starting DJDE can be in ASCII or EBCDIC.
- The output is optimized for printer speed.
- Shading support.
- Ink List/Palette/Catalogue color support.

- FEED command support for mapping input tray names/sizes appropriately.
- Highlight color support.
- Intelligent shade joining for efficiency.

5.5.4 Limitations

Limitations of the **Metacode Generator** are:

- Only the Metacode output document and image (.img) resources are generated. No fonts (.fnt), forms (.frm), or resources (.jsl) are generated.
- An existing JSL must be present to be mapped. The Metacode Generator parameters must be configured to provide information about the JSL (the JDL and JDE name to use).
- Data verification by is essential to ensure data specific requirements are supported.

5.5.5 Metacode Generator Parameters

The Metacode Generator parameters are used with the Metacode Generator.

A default set of parameters and values is used when you add a generator to your project. When you work through the Metacode Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the Metacode Generator Wizard, see [“Metacode Generator Wizard” on page 196](#).

5.5.6 Metacode Generator Wizard

5.5.6.1 Launching the Metacode Generator Wizard

The Metacode Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The Metacode Generator wizard launches when you:

- Create a new project that produces Metacode output. The New Component Wizard launches the Metacode Generator Wizard if the **Auto Start Generator Wizard(s)** option box is checked.
- Double-click the Metacode Generator on the Project tab.

The Metacode Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select Metacode output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the Metacode Generator Wizard are:

1. Identify Your Generator.
2. Select Metacode Output Details.
3. Specify JSL/DJDE Attributes.
4. Specify Printer Attributes.
5. Specify Data Element Options.

5.5.6.1.1 Identify Your Generator

The following fields are shown on the **Metacode Generator Wizard – 1 of 5** screen:

- **Generator name.** Enter a name for your generator or accept the default of Metacode Generator.

Since it is possible to have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, for example, Metacode Generator2.
- **Model Generator on Metacode Generator in the Project File.**
 1. Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to create a simple project file with default Metacode Generator values. You can make changes to any of the parameters later in the **Properties** window.
 2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.5.6.1.2 Select Metacode Output Details

The following fields are shown on the **Select Metacode Output Details – 2 of 5** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add or Edit a Resource Path.**

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

- Click **Remove** to remove the resource path.
3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.5.6.1.3 Specify JSL/DJDE Attributes

This step sets the attributes of the JSL/DJDE:

- **DJDE IDEN string.** Specify a particular string to be used as a DJDE IDEN string, which is prepended to every DJDE record. Default setting is '\$\$DJDE\$'.
- This value should be identical to the IDEN string defined in the JSL that will be associated with the generated Metacode data. The format used for this string is the Xerox PDL/JSL string format, which allows strings to be specified as ASCII or EBCDIC encoded, or in Hexadecimal notation.
- To specify that this string should be ASCII encoded, the string \$\$DJDE\$\$ would be entered as A '\$\$DJDE\$\$'.
- To specify an EBCDIC encoding, the same string would be entered as E '\$\$DJDE\$\$'.
- Alternately, because EBCDIC is the default encoding, simply using \$\$DJDE\$\$ would specify an EBCDIC encoding. In rare occasions when hexadecimal

format is required, you should provide two hex characters for each byte, and wrap it with a `X'` as follows: `X'2424444a44452424'`.

- **RStack string.** Specify a particular string to be used to indicate the end of the Metacode document. The Metacode Generator will include a non-printable record at the end of the document that contains this string.
 - The JSL for this report should define an RSTACK command that uses this string. The format used for this string is the Xerox PDL/JSL string format, which allows strings to be specified as ASCII or EBCDIC encoded, or in Hexadecimal notation.
 - To specify that this string should be ASCII encoded, the string ENDDOCUMENT would be entered as `A'ENDDOCUMENT'`.
 - To specify an EBCDIC encoding, the same string would be entered as `E'ENDDOCUMENT'`.
 - Alternately, because EBCDIC is the default encoding, simply using ENDDOCUMENT would specify an EBCDIC encoding. In rare occasions when hexadecimal format is required, you should provide two hex characters for each byte, and wrap it with a `X'` as follows: `X'2424444a44452424'`.
- **Line draw font name.** Specifies the name of the font to be used for line drawing commands. The value should correspond to the name of this font used by the printer that will print the generated Metacode document(s). Common values are FORMSX and FORMS\$.
- **First DJDE encoding.** Specifies the format of the first DJDE statement. A setting of Default will force EBCDIC encoding if the Online parameter is false, ASCII if the Online Parameter is true.

Click **Next** to proceed to the next step, click **Cancel to exit the Metacode Generator Wizard**, or click **Back** to change any of your previous settings.

5.5.6.1.4 Specify Printer Attributes

This step sets the properties of printer-related attributes:

- **Max Record Size.** Select the maximum size for each record.
- **Printer Mode.** Select the printer mode: **Online** or **Offline**.
- **CCType.** Select the style of Carriage Control bytes that will be used for each data record: **Default**, **ASCII ANSI**, **EBCDIC ANSI**, or **Machine**.

Click **Next** to proceed to the next step, click **Cancel to exit the wizard**, or click **Back** to change any of your previous settings.

5.5.6.1.5 Specify Data Element Options

This final step sets the data element properties of the output file:

- **Form Options.** Select this check box to have forms pass through.
- **Shade Options.** Select this check box to have shades pass through.
- **Image Options.** Select this check box to have images pass through.
- **Image Mode.** Select an image mode: **Inline** or **External**.

Click **Finish** to save the settings or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

5.6 PCL Generator



The **PCL Generator** produces PCL5 output from the XIF Intermediate Format. Output from traditional high-volume data streams may be transformed into PCL suitable for desktop or production printers.

The PCL Generator provides integration with desktop and production PCL printers from legacy printstreams.

5.6.1 Applications

The PCL Generator is often used to drive printers that once used Xerox XES/UDK or Metacode, and have been upgraded to support PCL.

5.6.2 Supported Features

Features of the **PCL Generator** include:

- Support for PCL5 output.
- Support for color image, text, shade and vector.
- Support for macro (forms/overlays from parser are output as macro) to make the output file smaller.
- Point sizes for fonts up to 3 decimal places, such as 10.005.
- Tray size and simplex/duplex settings are passed from parser.
- Edge-to-edge printing. Support for edge-to-edge printing done by slightly shrinking the page image and/or shifting the image to the center of the page.
- Support for PCL fonts stored in multiple directories/datasets.
- Ability to embed fonts in PCL output. To do this, you must specify the external PCL font (or soft font) in the MFCT, and then configure the PCL Generator parameter **FdPclFnt** to point to the location of the soft fonts directory. Example:
`FdPclFnt={d2eAppPath}/SoftFonts/{0}.fnt`



Note: For more information on the MFCT, see “[Master Font Correlation Table](#)” on page 502.

5.6.3 Limitations

Limitations of the **PCL Generator** are:

- No support for PCL5e, PCL5c, or PCL6.
- Data verification is essential to ensure data specific requirements are supported, or to determine whether an enhancement may be needed.

5.6.4 PCL Generator Parameters

The PCL Generator parameters are used with the PCL Generator.

A default set of parameters and values is used when you add a generator to your project. When you work through the PCL Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator’s parameters in the **Properties** window before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the PCL Generator Wizard, see “[PCL Generator Wizard](#)” on page 200.

5.6.5 PCL Generator Wizard

5.6.5.1 Launching the PCL Generator Wizard

The PCL Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The PCL Generator Wizard launches when you:

- Create a new project that produces PCL output. The **New Component Wizard** launches the PCL Generator Wizard if the **Auto Start Generator Wizard(s)** option box is checked.
- Double-click the PCL Generator on the **Project** tab.



Note: For more information, see “[New Component Wizard](#)” on page 95 or “[Project Tab](#)” on page 106.

The PCL Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select PCL output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the PCL Generator Wizard are:

1. Identify Your Generator.
2. Select PCL Output Details.
3. Select Property Settings.



Note: For information on steps 2 and 3, see “[Select PCL Output Details](#)” on page 201 and “[Select Property Settings](#)” on page 202.

5.6.5.1.1 Identify Your Generator

The following fields are shown on the **PCL Generator Wizard – 1 of 3** screen:

- **Generator name.** Enter a name for your generator or accept the default of PCL Generator.

Since you can have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, for example, PCLGenerator2.

- **Model generator on PCL Generator in the project file.**

1. Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to create a simple project file with default PCL Generator values. You can make changes to any of the parameters later in the **Properties** window.
2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.6.5.1.2 Select PCL Output Details

The following options are shown on the **Select PCL Output Details – 2 of 3** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add or Edit a Resource Path.**

To change a resource path:

1. Select the resource by clicking on it in the list.

2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - a. Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - b. Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.



Note: For more information, see “Select Resource Path/File Window” on page 102.

- c. Click **Remove** to remove the resource path.
3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.6.5.1.3 Select Property Settings

The following options are shown on the **Property Settings – 3 of 3** screen and sets the properties of the output file:

- **Size.** Using the **Up** and **Down** arrow button, select a percentage of scale ratio. Since PCL printers tend to have unprintable areas in their margin, it is often required to scale down the page data in order to fit the entire page into the printable area.
- **Output Options.** Using the **Up** and **Down** arrow buttons, select a dots-per-inch (DPI) resolution to be used in the output document's coordinate system. This enables you to ensure the PCL Generator creates documents that match your printer's capabilities.
- **Copies.** Using the **Up** and **Down** arrow buttons, select the **Number** of copies to be printed when the output document(s) are sent to the printer.
- **Shift By.** Enables all data on the page to be shifted by the given amounts in both the horizontal (X) and vertical (Y) directions. The values are in 300 DPI.
- **X.** Horizontal shift at 300 DPI (300 = 1 inch) along the X axis of the page (in Portrait orientation).
- **Y.** Vertical shift at 300 DPI (300 = 1 inch) along the Y axis of the page (in Portrait orientation).
- **Mode.** Select the **color mode** from the drop-down list:
 - **Black and White.** All text is output in black, and color images/shades/vectors will be output as grayscale elements.
 - **Color.** Generates color raster image/shade/vector/color text as required.

Click **Finish** to save the settings or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

5.7 PDF Generator



The **PDF Generator** transforms XIF data into high-fidelity Adobe Portable Document Format (PDF) which is viewable and printable. It supports the following features:

- PDF documents can remain searchable with the appropriate use of font mapping.
- Bookmarking and URL links enabled through the Table of Contents component and the URL Link component defined in the **Project** tab.
- DBCS support available using standard Adobe CJK fonts, enabled through the use of Unicode encoding in XIF.
- Embedded Type 1 fonts.
- Embedded TrueType fonts.
- OpenType CFF (Compact Font Format) font subsetting.
- Dynamic creation and embedding of Type 3 fonts (with searchable text) even for double-byte Adobe CJK fonts.
- Overlays/Forms converted once, not each time they are used.
- Flate compression of text, graphics, and fonts used within a PDF.
- True PDF color-fill shading can be substituted for Bitmap shade patterns in the incoming documents, to improve the look of the output and reduce file sizes.
- Linearization, can be done in-place while generating the output PDF file, which enables better viewing over the Web, in certain circumstances, when using a browser.
- Standard Adobe Acrobat 128-bit encryption for document security and fraud protection.

Related Links

- [“Output Transformation Engine Font Mapping” on page 501](#)
- [“Table of Contents Component” on page 330](#)
- [“Index Writer Component” on page 272](#)
- [“Project Tab” on page 106](#)

5.7.1 PDF Output

Portable Document Format (PDF) is a document architecture and file format developed by Adobe. Adobe Acrobat Reader can be used to view PDF files created by the PDF Generator and Adobe Acrobat Exchange can be used to both view and manipulate PDF files created by the PDF Generator.

The following features are available for PDF generation using the PDF Generator with other components:

- Automatically generate PDF Bookmarks using the Table of Contents component.
- Automatically generate URLs using the URL Link component.
- Add color during processing using the Constant Print component.
- Allow document breaking (multiple PDF files from a single input file) using the Document Breaking component.

Related Links

- [“Table of Contents Component” on page 330](#)
- [“Index Writer Component” on page 272](#)
- [“Constant Print Component” on page 238](#)
- [“Document Breaking Component” on page 262](#)

5.7.2 PDF File Linearization

Linearization is the process of rearranging objects in PDF files such that the file's pages can be displayed to a satisfying extent without reading the entire file in advance. This process requires some knowledge of how a file is displayed in Adobe Acrobat.

When a PDF file is accessed on an HTTP server, the server would send the entire file, by default. Adobe Acrobat would read the first few kilobytes to determine whether it is linearized before aborting the transaction.

Linearization is recommended when the PDF file is archived.

5.7.3 Substituting Colors for Patterns in PDFs with Shade Mapping

From time to time, you may encounter documents with particularly complex patterns or patterns that are predefined with the input stream type such as AFP or Metacode. Upon converting these documents to PDF files, you may discover that the visual fidelity of these patterns do not translate adequately, causing issues like illegible text or distortion inside the patterned areas. In these situations, you can use shade mapping to substitute each pattern with another color, allowing you to select a more suitable arrangement for your output.

To map a pattern to a color in your PDF output files, a few separate parts are involved:

1. Setting the PDF Generator's `ShadeMode` parameter.
2. Acquiring the mapping names for unmapped shades.
3. Configuring the mapping keys for patterns.

5.7.3.1 Step 1: Setting the PDF Generator's `ShadeMode` Parameter

Configure your project with the PDF Generator. Access the PDF Generator's properties, navigate to the **ShadeMode** parameter and ensure it is set to **Map**. The `ShadeMode` parameter determines the shading technique to use, with the `Map` value indicating that the `ShadeMap` parameter, which defines the shade map, should fill in the pattern area with the chosen color command.



Note: For more information, see [“PDF Generator” on page 203](#).

5.7.3.2 Step 2: Acquiring the Mapping Names for Unmapped Shades

You must first identify the input mapping names for the individual patterns you want to map. Mapping names consist of a pattern name/key and sometimes a color key. Pattern names are unique to each pattern in your document and vary in length, but are always a combination of four sets of numbers groups, similar to 1.23.456.7890 or 12.34.567.8901. If your document contains areas that are filled by identical patterns but in different colors, be aware that the pattern name itself remains the same between them, but with the RGB color code appended to the end by way of a period to link the two keys. For example, two matching patterns with one in red and one in blue would be written out as 1.23.456.7890.200_20_0 and 1.23.456.7890.0_100_250.

You can locate these mapping keys through either of the following methods:

- Using the input file you want to apply shade mapping on, run the project. Once the job completes, navigate to the Log sub-tab within the Output tab for your job. Scroll through the log entries until you can locate a section on **PatternName** and the keys are shown in the Message column. (The pattern names are also shown in

the job's output report file so you can open the report file in your preferred text editor if you need to copy and paste the names.)

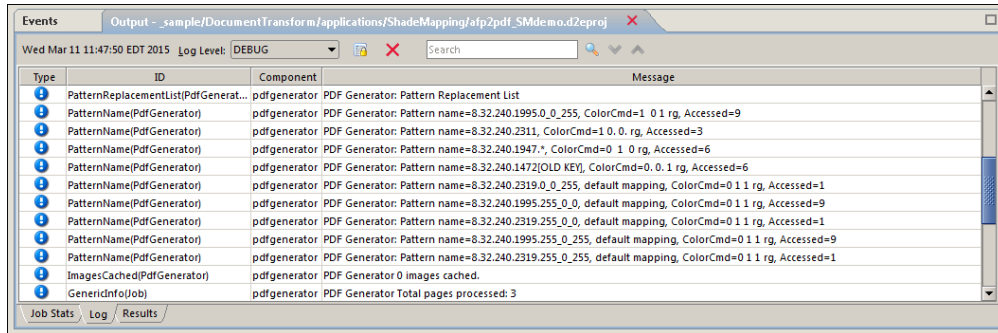


Figure 5-2: Log sub-tab within Output tab showing unique pattern keys

- Open the input file you want to apply shade mapping on in the **Page View** tab. Then in the Summary pane, depending on the category containing your pattern, select either **Shade** or **Vector**, which displays all existing patterns in the Elements pane. As you select the individual pattern elements, the pattern keys are shown in the Properties window as the value for the **XifShadePatternName** parameter. If you need to obtain the color key, it is shown in the **XifShadePatternColorKey** parameter. (When defining the mapping key for matching patterns with different colors, you must include both the pattern and color key by linking them together with a period to create the full mapping key.)

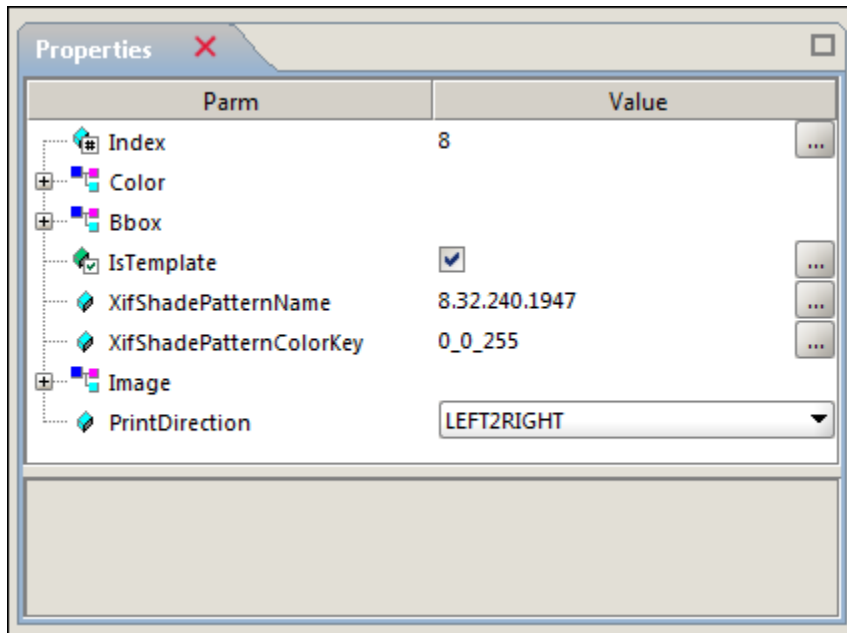


Figure 5-3: Properties tab showing pattern and color keys



Note: If you encounter a pattern key with [OLD KEY] appended to the end, it means that it is an older, deprecated key and is available to maintain backwards compatibility. It is highly recommended you replace any pattern keys with [OLD KEY] in the name with the name as reported through either the report logs or Page View tab.

5.7.3.3 Step 3: Configuring the Mapping Keys for Patterns

With the mapping keys in hand, you can map them to other colors. To specify shade mapping in the PDF Generator's properties:

1. Ensure that the **ShadeMode** parameter is set to **Map**. The Map value indicates to the generator that the values defined in the ShadeMap parameter should be used to fill in shaded areas with a solid color.

2. Right-click on the **ShadeMap** parameter and from the context menu that appears, click **Add**.

A new set of XifShadePatternMappingKey and ColorCmd child parameters appear.

3. Locate the pattern key for the shade pattern to be mapped and enter it as the value for the **XifShadePatternMappingKey** parameter. (The mapping key is automatically set as the name for the particular set of child parameters to help you differentiate between sets.)

If the pattern is reused throughout your document in different colors, since each instance can be mapped to a different color ensure that you are using both the pattern and color keys in your mapping key to identify the specific pattern to perform shade mapping on. However, if you wish to map all instances of the same pattern, which includes all its color variants, to the same color, you can add a **wildcard** (*) to the pattern key (for instance, 12.345.678.9012.*) in the parameter value.

4. In the **ColorCmd** parameter, enter the color you want to use to substitute into your pattern using any valid PDF color command. For example, 0.5 0.5 0.5 rg returns a medium grey color.
5. Repeat steps 2-5 for all pattern keys you wish to apply shade mapping to.
6. Optionally, using the **DefaultMappingColorCmd** and a PDF color command as the value, you can also set a default color to map to all patterns that have an undefined or invalid color. If the value is blank, then the existing pattern is rasterized. By default, the setting is .75 .75 .75 rg, which is a light grey.
7. **Save** your settings and **Execute** the project.

Review your output and the patterns are substituted with the colors they are mapped to.

5.7.4 PDF Generator Parameters

The PDF Generator parameters are used with the PDF Generator. A default set of parameters and values is used when you add a PDF Generator to your project. When you work through the PDF Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your Output Transformation Engine project.

The following PDF Generator parameters are discussed in detail in this guide:

- [“AcrobatOpen Parameters” on page 208](#)



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the PDF Generator Wizard, see [“PDF Generator Wizard” on page 211](#).

5.7.4.1 AcrobatOpen Parameters

The **AcrobatOpen** parameters contain options to configure how PDF files are initially displayed upon opening as well as general document options, such as author and creator metadata.

The following parameters are available:

- **ApplyParams.** Defines how the **AcrobatOpen** parameters are applied to the output document. You can choose from the following options:
 - **Preserve.** Retains all settings from the input document.
 - **OverrideAll.** Overrides all settings from the input document, including any that are not specified, and applies the **AcrobatOpen** settings.
 - **Override.** Overrides only the **AcrobatOpen** settings that are configured. If a particular **AcrobatOpen** parameter value is not specified, then the input document's setting is applied.
- **PdfOpenAct.** Contains display options for when the PDF file is opened.
 - **OpenAct.** Specifies how the initial page is displayed. These options correspond with the Fit type names used in PDF standards. You can choose from the following options:
 - **Fit.** Fits the page to the window. The page is magnified only enough to fit the entire page within the window, both horizontally and vertically.

- **FitB.** Fits the bounding box for the page contents to the window. The page is magnified only enough to fit its bounding box entirely within the window, both horizontally and vertically.
- **FitR.** Fits the rectangle specified by the parameters for the window. The page is magnified only enough to fit the rectangle specified by the `left`, `bottom`, `right`, and `top` coordinates within the window, both horizontally and vertically.
- **FitBH.** Fits the width of the bounding box for the page contents to the window. Specifying a `top` value positions the vertical coordinate at the top edge of the window and the page is magnified only enough to fit the entire width of the bounding box within the window.
- **FitBV.** Fits the height of the bounding box for the page contents to the window. Specifying a `left` value positions the horizontal coordinate at the left edge of the window and the page is magnified only enough to fit the entire height of the bounding box within the window.
- **FitH.** Fits the width of the page to the window size. Specifying a `top` value positions the vertical coordinate at the top edge of the window and the page is magnified only enough to fit the entire width of the page within the window.
- **FitV.** Fits the height of the page to the window size. Specifying a `left` value positions the horizontal coordinate at the left edge of the window and the page is magnified only enough to fit the entire height of the page within the window.
- **XYZ.** Specifies the distance from the start of the page to the top-left corner of the window and a zoom factor using `left`, `top`, and `zoom` values. The zoom setting uses 1 to indicate 100% magnification. If any of these values are set to NULL, the current value for the parameter is preserved. Setting zoom to 0 results in the same behavior as a zoom setting of NULL.
- **Left.** Denotes a value for the left side of a bounding box to use with the associated action.
- **Top.** Denotes a value for the top of a bounding box to use with the associated action.
- **Right.** Denotes a value for the right side of a bounding box to use with the associated action.
- **Bottom.** Denotes a value for the bottom of a bounding box to use with the associated action.
- **Zoom.** Indicates the zoom level ratio for initially displaying the PDF in Acrobat. This parameter is only used when the **OpenAct** parameter is set to **XYZ**. As an example, to specify that a PDF should be displayed at 50% zoom, use 0.5 as the value.
- **ViewerPreferences.** Contains options to specify how the document is displayed on the screen.

- **DisplayDocTitle.** Indicates whether the Windows title bar displays the document title as retrieved from the Title value in the document properties. If false, the title bar displays the name of the PDF file containing the document. By default, this is set to false. However, when working with the Accessible PDF Generator, this parameter is always set to true to conform with PDF/UA requirements.
- **Pagemode.** Defines how the document is displayed when the PDF file is opened in Adobe Acrobat Reader. Users can customize the page mode view by themselves after the file has been loaded. You can choose from the following options:
 - **Document.** Retains the input document's original page mode settings.
 - **Bookmarks.** Displays the pages of the document with the bookmarks panel visible. You can also indicate the bookmark levels to open with the BookmarksLevel parameter.
 - **Thumbnail.** Displays the pages of the document with the page thumbnails panel visible.
 - **FullScreen.** Displays the pages of the document in full screen mode.
 - **None.** Displays the pages of the document with all other panels hidden.
- **PdfAuthor.** Inserts an author name in the PDF file's Document Properties.
- **PdfCreator.** Inserts a creator name in the file's Document Properties.
- **PdfKeywords.** Inserts keywords to the file's Document Properties.
- **PdfSubject.** Inserts a subject in the file's Document Properties.
- **PdfTitle.** Inserts a title in the file's Document Properties.
- **CustomProperties.** Adds custom metadata, such as a company name or document version number, to the generated PDF.
 - **Name.** Designates the custom property name.
 - **Value.** Designates the custom property value.
- **BookmarkLevel.** Denotes the level of bookmarks to open. If you want all bookmarks to be closed, use a value of 0. You must also ensure **Pagemode** is set to **Bookmarks** for this parameter to take effect.

5.7.5 PDF Generator Wizard

5.7.5.1 Launching the PDF Generator Wizard

The PDF Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The PDF Generator wizard launches when you:

- Create a new project that produces PDF output. The **New Component Wizard** launches the PDF Generator Wizard if the **Auto Start Generator Wizard(s)** option box is checked.
- Double-click the PDF Generator on the Project tab. For more information, see [“Project Tab” on page 106](#).

The PDF Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select PDF output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the PDF Generator Wizard are:

1. Identify your Generator.
2. Select PDF Output Details.
3. Select PDF Features.
4. Select Basic Document Options.
5. Select Compression Options.
6. Select Open View Options
7. Select Security and Encryption Options.
8. Specify Authorship.

5.7.5.1.1 Identify Your Generator

The following fields are shown on the **PDF Generator Wizard – 1 of 8** screen:

- **Generator name.** Enter a name for your generator or accept the default of PDF Generator.

Since it is possible to have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, such as PDFGenerator2.

- **Model generator on PDF Generator in the project file.**

1. Click the ellipsis button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to

create a simple project file with default PDF Generator values. You can make changes to any of the parameters later in the Properties window.

2. Click **Next** to proceed to the next step or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.7.5.1.2 Select PDF Output Details

The following options are shown on the **Select PDF Output Details – 2 of 8** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add or Edit a Resource Path.**

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - a. Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - b. Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

- c. Click **Remove** to remove the resource path.
3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.7.5.1.3 Select PDF Features

This step lets you select the attributes of your generated output. Most Output Transformation Engine projects will use only a subset of these.



Note: As you select these options, the number of steps changes dynamically to accommodate the additional screens.

By default, the first three options are selected. If you want additional options, select the check boxes beside your choices:

- **Basic document options.** Refers to page orientation, offset, and scaling of output.
- **Compression required.** The PDF commands are never compressed. Only line data may be compressed.
- **Open view options.** Establishes how the PDF document will look when opened by setting page mode, image placement, and display positioning details.

- **PDF authorship.** Determines the text to put into the Document Info box, including title, subject, author and keywords. This information is accessible in Adobe Acrobat from the file menu **File > Document Properties > Summary**.
- **Encryption required.** Specifies the security which should be applied to the generated document. This information is accessible in Adobe Acrobat from the file menu File Document Properties Security. Also select this option to enable linearization.

Click **Next** to proceed to the next step or click **Back** to change your previous settings. Click **Cancel** to exit the wizard.



Note: If you exit the wizard before specifying the required generator options, the project will load with a simple generator project file. You will need to specify the generator parameters in the Parameters window before you run your project.

For more information, see [“PDF Parser Parameters” on page 164](#).

5.7.5.1.4 Select Basic Document Options



Note: This screen will only appear if **Basic document options** is selected in the previous step. For more information, see [“Select PDF Features” on page 212](#).

These choices determine following basic PDF document options:

- **Shift By:**
 - **X position.** The starting point from the left of the page before any rotation.
 - **Y position.** The starting point from the bottom of the page before any rotation.
- **Size:**
 - **Scale by.** Scales the output by the factor you specify. A value of 100 means no scaling. A value smaller than 100 will reduce the size of the output; a value greater than 100 will enlarge the output.

Click **Next** to proceed to the next step, click **Back** to return to the previous step, or click **Cancel** to exit the wizard.

5.7.5.1.5 Select Compression Options



Note: This screen will only appear if **Compression required** is selected in the PDF Wizard Step 3. For more information, see “[Select PDF Features](#)” on page 212.

On this screen select the parameters for text, image or font compression.

The PDF Generator provides for three areas of compression:

- **Text Compression:**
 - **None.** Specifies no text compression will be used.
 - **Flate.** Specifies that Flate text compression will be used.
- **Image Compression:**
 - **None.** No image compression will be used.
 - **Flate.** Flate image compression will be used.
- **Font Compression:**
 - **None.** No font compression will be used.
 - **Flate.** Flate font compression will be used.

Click **Next** to proceed to the next step, or click **Back** to return to the previous screen. Click **Cancel** to exit the wizard.

5.7.5.1.6 Select Open View Options



Note: This screen will only appear if **Open View Options** is selected in the PDF Wizard Step 3. For more information, see “[Select PDF Features](#)” on page 212.

On this screen, establish how your PDF document will first appear when it is opened in Adobe Acrobat or Adobe Acrobat Reader:

<Text Compression.> Determines the format of the page display. Options are:

- <Bookmarks.> Displays the bookmarks alongside the page in a table of contents fashion.
- <Full screen.> Displays the document in full screen view.
- <Document.> Displays the document view.
- <Thumbnail.> Creates and shows the thumbnail-sized view of the pages, alongside the page.

<Display Positioning.> Defines the initial or default view when the document is opened in Acrobat. Options are:

- *<Viewer default.>* Utilizes the default view settings within the Acrobat Reader. If you choose not to accept the default, you can make one of the selections below.
- *<Fit page.>* Makes the current page fit inside the viewer window.
- *<Fit to width.>* Makes the visible width of the current page fit inside the viewer window.
- *<Fit to height.>* Makes the visible height of the current page fit inside the viewer window.
- *<Fit to rectangle.>* Makes the page fit within a rectangle of the dimensions specified in the Rectangle section.
- *<Fit bounding box of page.>* Makes a bounding box fit the page.
- *<Fit bounding box horizontally.>* Makes a bounding box fit the page horizontally.
- *<Fit bounding box vertically.>* Makes a bounding box fit the page vertically.
- *<Specify top and left corner.>* Specifies a top and left corner where the page view will start.

<Rectangle.> Defines the location and dimensions of a rectangular view space. These values are required when the **Fit to rectangle** display positioning has been selected. Options are:

- *<Top.>* Enter a value in dpi to indicate where the top boundary of the rectangle is anchored.
- *<Bottom.>* Enter a value in dpi to indicate where the bottom boundary of the rectangle is anchored.
- *<Left.>* Enter a value in dpi to indicate where the left boundary of the rectangle is anchored.
- *<Right.>* Enter a value in dpi to indicate where the right boundary of the rectangle is anchored.

<Zoom percentage.> Select from the drop-down box the percentage of zoom required to view the document. The default is 100, meaning the document will appear as actual size, without any zooming either in or out. A value lower than 100, such as 25, will make the document appear smaller; a value greater than 100, such as 125, will make the document appear larger.

Click **Next** to proceed to the next step, or click **Back** to return to the previous screen. Click **Cancel** to exit the wizard.

5.7.5.1.7 Select Security and Encryption Options



Note: This screen will only appear if **Encryption Required** is selected in the PDF Wizard Step 3. For more information, see [“Select PDF Features” on page 212](#).

On this screen, select security options to be applied to the generated document:

Encryption Level. Select the encryption level you wish to use: **None**, **40-bit**, or **128-bit**.

Depending on the encryption level you select, the following user capabilities for users who view the generated PDF document are made available:

- **40-bit Permissions level selected:**
 - **Allow printing.** Allows users to print the generated PDF document.
 - **Allow changes to the document.** Allows users to change the generated PDF document and save the changes.
 - **Allow content copying or extraction, and enable accessibility.** Allows content to be copied or extracted from the generated PDF document, and enables settings which make the document more accessible for visually impaired users.
 - **Allow adding or changes to comments and form fields.** Allows users to add or edit comments and text fields in the generated PDF document.
- **128-bit Permissions level selected:**
 - **Enable content access for the visually impaired.** Enables settings, such as text-to-speech and text reflow, which make the generated PDF document more accessible for visually impaired users.
 - **Allow content copying and extraction.** Allows content to be copied or extracted from the generated PDF document.
 - **Changes allowed.** Specifies the changes, if any, which can be made to the generated PDF document. You can allow changes to content, comments, form fields, or document assembly.
 - **Printing allowed.** Determines whether users can print the generated PDF document.

Password (Optional). Enter the passwords you wish to use with the generated PDF document:

- **User.** The password the user viewing the document must enter to view and perform user capabilities above.
- **Owner.** The password the owner of the document must enter to make changes to the generated PDF document.

Click **Next** to proceed to the next step, or click **Back** to return to the previous screen. Click **Cancel** to exit the wizard.

5.7.5.1.8 Specify Authorship



Note: This screen will only appear if **PDF Authorship** is selected in the PDF Wizard Step 3. For more information, see “[Select PDF Features](#)” on page 212.

These optional details indicate the authorship of the PDF document.

Enter the details you want users to be able to see when viewing the output in Adobe Acrobat. This information is found in Adobe Acrobat's **File > Document Properties > Summary**:

- **Author.** Usually the name of the person who authored the document. It may also be a company name.
- **Creator.** The name of the person who created the output document. May also be a company name or the name of the project.
- **Keywords.** Indicates keywords to enable searching for the document across a network, or for categorizing the document in a repository.
- **Title.** The title of the document, for example, June Statement.
- **Subject.** The subject of the document, for example, Monthly Statement.

Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

5.8 PDF/A Generator



The **PDF/A Generator** converts XIF data into high fidelity PDF documents that are compliant with the PDF/A-1b (ISO 19005-1:2005) standard.

Based on the PDF 1.7 standard, PDF/A defines a format of PDF that is suitable for long term archiving.

The PDF/A Generator will ensure that all documents generated are fully compliant within the PDF/A-1b format. The 1B specification is designed to ensure that the exact visual reproduction of the documents will be preserved over time, regardless of the application version used to view or print the PDF documents. The 1B standard therefore restricts or forbids the usage of some features within the PDF 1.7 specification, as well as requiring the use of others. For example, PDF/A-1b documents are required to use embedded fonts only; no Base 14 font references may be used. This is to ensure that the PDF document provides all required visual resources to the viewer, instead of the viewer substituting its own resources.

The PDF/A Generator supports many of the same features as the PDF Generator, with the following key differences:

- To facilitate font mapping, users can still select fonts from the list of Base 14 fonts. The PDF/A Generator will automatically embed equivalent TrueType fonts into the generated PDF documents (which are provided with the engine).
- Non-Base 14 font mappings must be configured as embedded fonts.
- If user configuration settings prevent the PDF/A Generator from producing PDF/A-1b compliant documents, the transform process will abort and provide the user with corresponding error messages.
- Encryption (password protection) of documents is not allowed.
- Linearization is not allowed.



Note: For more information, see “PDF Generator” on page 203.

5.8.1 PDF/A Output

PDF/A-1b format is a constrained form of Adobe PDF version 1.7 intended to be suitable for long-term preservation of page-oriented documents for which PDF is already being used in practice. The proposed standard is being developed by an ISO working group with representatives from government, industry, and academia and active support from Adobe Systems Incorporated.

PDF/A attempts to maximize:

- Device independence
- Self-containment
- Self-documentation

PDF/A constraints include:

- Audio and video content are forbidden.
- JavaScript and executable file launches are prohibited.
- All fonts must be embedded and also must be legally embeddable for unlimited, universal rendering.
- Colorspaces are specified in a device-independent manner.
- Encryption is disallowed.
- Use of standards-based metadata is mandated.

5.8.1.1 PDF/A Generator Parameters

The PDF/A Generator parameters are used with the PDF/A Generator. A default set of parameters and values is used when you add a PDF/A Generator to your project. When you work through the PDF/A Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the PDF/A Generator Wizard, see ["PDF/A Generator Wizard" on page 219](#).

5.8.1.2 PDF/A Generator Wizard

5.8.1.2.1 Launching the PDF/A Generator Wizard

The PDF/A Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The PDF/A Generator wizard launches when you:

- Create a new project that produces PDF/A output. The **New Component Wizard** launches the PDF/A Generator Wizard if the **Auto Start Generator Wizard(s)** option box is selected.
- Double-click the PDF/A Generator on the **Project** tab.



Note: For more information, see ["New Component Wizard" on page 95](#) or ["Project Tab" on page 106](#).

The PDF/A Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select PDF/A output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the PDF/A Generator Wizard are:

1. ["Identify Your Generator" on page 189](#).
2. ["Select PDF/A Output Details" on page 220](#).
3. ["Select PDF/A Features" on page 221](#).
4. ["Select Basic Document Options" on page 213](#).
5. ["Select Compression Options" on page 214](#).

6. “Select Open View Options” on page 214.
7. “Specify Authorship” on page 217.

5.8.1.2.2 Identify Your Generator

The following options are displayed on the **PDF/A Generator Wizard – 1 of 6** screen:

- **Generator name.** Enter a name for your generator or accept the default of PDF/A Generator.

Since you can have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, such as PDF/A Generator2.

- **Model generator on PDF/A Generator in the project file.**
 1. Click the ellipsis button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to create a simple project file with default PDF/A Generator values. You can make changes to any of the parameters later in the **Properties** window.
 2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.8.1.2.3 Select PDF/A Output Details

The following options are displayed on the **Select PDF/A Output Details – 2 of 6** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add or Edit a Resource Path.**

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.8.1.2.4 Select PDF/A Features

This step lets you select the attributes of your generated output. Most Output Transformation Engine projects will use only a subset of these.



Note: As you select these options, the number of steps changes dynamically to accommodate the additional screens.

By default, the first three options are selected. If you want additional options, select the check boxes beside your choices:

- *<Basic document options.>* Refers to page orientation, offset, and scaling of output.
- *<Compression required.>* The PDF/A commands are never compressed. Only line data may be compressed.
- *<Open view options.>* Establishes how the PDF/A document will look when opened by setting page mode, image placement, and display positioning details.
- *<PDF authorship.>* Determines the text to put into the Document Info box, including title, subject, author and keywords. This information is accessible in Adobe Acrobat from the file menu **File > Document Properties > Summary**.

Click **Next** to proceed to the next step, or click **Back** to return to the previous step. Click **Cancel** to exit the wizard.



Note: If you exit the wizard before specifying the required generator options, the project will load with a simple generator project file. You will need to specify the generator parameters in the **Properties window** before you run your project.

5.8.1.2.5 Select Basic Document Options



Note: This screen will only appear if **Document Options** is selected in the PDF/A Wizard Step 3.

For more information, see “[Select PDF/A Features](#)” on page 221.

These choices determine the following basic PDF/A document options:

- **Shift By:**
 - *<X position.>* The starting point from the left of the page before any rotation.
 - *<Y position.>* The starting point from the bottom of the page before any rotation.
- **Size:**

- *<Scale by.>* Scales the output by the factor you specify. A value of 100 means no scaling. A value smaller than 100 will reduce the size of the output; a value greater than 100 will enlarge the output.

Click **Next** to proceed to the next step, or click **Back** to return to the previous step. Click **Cancel** to exit the wizard.

5.8.1.2.6 Select Compression Options



Note: This screen will only appear if **Compression required** is selected in the PDF/A Wizard Step 3.

For more information, see *“Select PDF/A Features” on page 221.*

On this screen select the parameters for text, image or font compression.

The PDF/A Generator provides for three areas of compression:

Text compression:

- *<None.>* Specifies no text compression will be used.
- *<Flate.>* Specifies that Flate text compression will be used.

Image Compression:

- *<None.>* No image compression will be used.
- *<Flate.>* Flate image compression will be used.

Font Compression:

- *<None.>* No font compression will be used.
- *<Flate.>* Flate font compression will be used.

Click **Next** to proceed to the next step, click **Back** to return to the previous step, or click **Cancel** to exit the wizard.

5.8.1.2.7 Select Open View Options



Note: This screen will only appear if **Open view options** is selected in the PDF/A Wizard Step 3.

For more information, see *“Select PDF/A Features” on page 221.*

On this screen, you establish how your PDF/A output documents will be displayed on the screen.

These options indicate how the PDF/A document will first appear when it is opened in Adobe Acrobat or Adobe Acrobat Reader:

Text Compression. Determines the format the page is displayed as. Options are:

- *<Bookmarks.>* Displays the bookmarks alongside the page in a table of contents fashion.
- *<Full screen.>* Displays the document in full screen view.
- *<Document.>* Displays the document view.
- *<Thumbnail.>* Creates and shows the thumbnail-sized view of the pages, alongside the page.

Display Positioning. Defines the initial or default view when the document is opened in Acrobat. Options are:

- *<Viewer default.>* Utilizes the default view settings within the Acrobat Reader. If you choose not to accept the default, you can make one of the selections below.
- *<Fit page.>* Makes the current page fit inside the viewer window.
- *<Fit to width.>* Makes the visible width of the current page fit inside the viewer window.
- *<Fit to height.>* Makes the visible height of the current page fit inside the viewer window.
- *<Fit to rectangle.>* Makes the page fit within a rectangle of the dimensions specified in the Rectangle section.
- *<Fit bounding box of page.>* Makes a bounding box fit the page.
- *<Fit bounding box horizontally.>* Makes a bounding box fit the page horizontally.
- *<Fit bounding box vertically.>* Makes a bounding box fit the page vertically.
- *<Specify top and left corner.>* Specifies a top and left corner where the page view will start.

Rectangle. Defines the location and dimensions of a rectangular view space. These values are required when the Fit to Rectangle display positioning has been selected. Options are:

- *<Top.>* Enter a value in dpi to indicate where the top boundary of the rectangle is anchored.
- *<Bottom.>* Enter a value in dpi to indicate where the bottom boundary of the rectangle is anchored.
- *<Left>.* Enter a value in dpi to indicate where the left boundary of the rectangle is anchored.
- *<Right.>* Enter a value in dpi to indicate where the right boundary of the rectangle is anchored.

<Zoom Percentage.> Select from the drop-down box the percentage of zoom required to view the document. The default is 100, meaning the document will appear actual size, without any zooming either in or out. A value lower than 100, such as 25, will make the document appear smaller; a value greater than 100, such as 125, will make the document appear larger.

Click **Next** to proceed to the next step, or click **Back** to return to the previous screen. Click **Cancel** to exit the wizard.

5.8.1.2.8 Specify Authorship



Note: This screen will only appear if **PDF/A authorship** is selected in the PDF/A Wizard Step 3.

For more information, see *“Select PDF/A Features” on page 221*.

These optional details indicate the authorship of the PDF/A document.

Enter the details you want users to be able to see when viewing the output in Adobe Acrobat. This information is found in Adobe Acrobat's **File > Document Properties > Summary**:

- **Author.** Usually the name of the person who authored the document. It may also be a company name.
- **Creator.** The name of the person who created the output document. May also be a company name or the name of the project.
- **Keywords.** Indicates keywords to enable searching for the document across a network, or for categorizing the document in a repository.
- **Title.** The title of the document, for example, June Statement.
- **Subject.** The subject of the document, for example, Monthly Statement.

Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to exit the wizard.

5.9 PostScript Generator



The **PostScript Generator** produces PostScript Level 3, suitable for printing on any PostScript Level 3 printer, from the XIF Intermediate Format.

Applications include migration from Metacode and AFP to PostScript enabled printers providing integration with desktop and production PostScript printers from legacy printstreams. In conjunction with other Output Transformation Engine project flow components, the PostScript Generator can selectively output and split document input allowing routing and migration from legacy printstreams.

5.9.1 Applications

This component is often used to drive printers that support PostScript or applications which allow PostScript viewing.

5.9.2 Supported Features

- Supports PostScript Level 3.
- Point sizes for fonts up to 3 decimal places, such as 10.005.
- Tray size and simplex/duplex settings are passed from parser.
- Support for color output.
- Group 4 (CCITT) Image compression.
- Edge-to-edge printing. Support for edge-to-edge printing done by slightly shrinking the page image and/or shifting the image to the center of the page.
- Support for Document Structuring Conventions (DSC) PostScript.

5.9.3 Limitations

Limitations of the **PostScript Generator** are:

- Cannot embed fonts into output.
- Data verification is essential to ensure data specific requirements are supported, or to determine whether an enhancement may be needed.

5.9.4 PostScript Generator Parameters

The PostScript Generator parameters are used with the PostScript Generator.

A default set of parameters and values is used when you add a PostScript Generator to your project. When you work through the PostScript Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the PostScript Generator Wizard, see [“PostScript Generator Wizard” on page 226](#).

5.9.5 PostScript Generator Wizard

5.9.5.1 Launching the PostScript Generator Wizard

The PostScript Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The PostScript Generator Wizard launches when you:

- Create a new project that produces PostScript output. The **New Component Wizard** launches the PostScript Generator Wizard if the **Auto Start Generator Wizard(s)** option box is selected.
- Double-click the PostScript Generator on the **Project** tab.



Note: For more information, see “*New Component Wizard*” on page 95 or “*Project Tab*” on page 106.

The PostScript Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select PostScript output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

The steps of the PostScript Generator Wizard are:

1. Identify Your Generator.
2. Select PostScript Output Details.
3. Select Property Settings.

5.9.5.1.1 Identify Your Generator

The following options are available on the **Postscript Generator Wizard – 1 of 3** screen:

- **Generator name.** Enter a name for your generator or accept the default of PostScript Generator.
Since you can have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, for example, PostScript Generator2.
- **Model generator on PostScript Generator in the project file.**
 1. Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to create a simple project file with default PostScript Generator values. You can make changes to any of the parameters later in the **Properties window**.
 2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.9.5.1.2 Select PostScript Output Details

The following options are available on the **Select Postscript Output Details – 2 of 3**:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.

- **Add or Edit a Resource Path**

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.9.5.1.3 Select Property Settings

This step sets the document attributes:

Compression. Select your desired compression options:

- **Enable Compression.** Select this check box to turn on compression of the output file.
- **Record Length.** Using the **Up** and **Down** arrow button, select a maximum record length for the output file.

Pages Per Sheet. Using the **Up** and **Down** arrow button, select the number of pages per sheet. It determines whether multiple pages are scaled to fit on a single output page (N-up mode). If set to a value greater than 1, the specified number of logical pages will be placed on one output page.

Shift By. Enables all data on the page to be shifted by the given amounts in both the horizontal, X, and vertical, Y, directions. The values are in 300 DPI.

- **X.** Horizontal shift at 300 dpi (300 = 1 inch) along the X axis of the page (in Portrait orientation).

- **Y.** Vertical shift at 300 dpi (300 = 1 inch) along the Y axis of the page (in Portrait orientation).

Size. Using the **Up** and **Down** arrow button, select a percentage of scale ratio. Since PostScript printers tend to have unprintable areas in their margin, it is often required to scale down the page data in order to fit the entire page into the printable area.

Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

5.10 TIFF Generator



The TIFF Generator allows production of single- or multiple-page TIFF images. It is typically used for applications such as fax output or for loading archive systems with individual output. The TIFF Generator supports the following features:

- Industry standard TIFF format, providing near-perfect fidelity but non-searchable output.
- TIFF Class B format (black-and-white, 1-bit), Class G format (grayscale, 8-bit), and Class P format (full-color RGB, 24-bit) images.
- Compression options: CCITT Group4 or None for black-and-white images; LZW, Packbits, Flate or None for grayscale and color images.
- Any resolution (DPI) can be generated.
- Fonts can be correlated to TrueType fonts installed on the system; supported in Java by AWT (Abstract Windows Toolkit).
- Ability to create single-page Xerox Image files (.IMG) or AFP Page Segments.
- Ability to resize pages to apply uniform image sizes.

5.10.1 TIFF Generator Parameters

The TIFF Generator parameters are used with the TIFF Generator.

A default set of parameters and values is used when you add a TIFF Generator to your project. When you work through the TIFF Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the TIFF Generator Wizard, see [“TIFF Generator Wizard” on page 229](#).

5.10.2 TIFF Generator Wizard

5.10.2.1 Launching the TIFF Generator Wizard

The TIFF Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The TIFF Generator Wizard launches when you:

- Create a new project that produces TIFF output. The **New Component Wizard** launches the TIFF Generator Wizard if the **Auto Start Generator Wizard(s)** option box is selected.
- Double-click the TIFF Generator on the **Project** tab.



Note: For more information, see [“New Component Wizard” on page 95](#) or [“Project Tab” on page 106](#).

The TIFF Generator Wizard guides you through the steps to identify the output file and its characteristics, the generator project file, and any document breaking options. You then select TIFF output options that are written to your project file. In this way, helps you build the generator parameter file as you create a project.

The steps of the TIFF Generator Wizard are:

1. [“Identify Your Generator” on page 189](#).
2. [“Select TIFF Output Details” on page 230](#).
3. [“Specify Image Settings” on page 230](#).

5.10.2.1.1 Identify Your Generator

The following options are available on the **TIFF Generator Wizard – 1 of 3** screen:

- **Generator name.** Enter a name for your generator or accept the default of TIFF Generator. This name given to the generator is used as the handle for that generator.

Since you can have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle.

- **Model generator on TIFF Generator in the project file.**

To change a resource path:

1. Click the ellipsis button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to create a simple project file with default TIFF Generator values. You can make changes to any of the parameters later in the **Properties window**.

2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.10.2.1.2 Select TIFF Output Details

The following options are available on the **Select TIFF Output Details – 2 of 3** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.
- **Add or Edit a Resource Path.**

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see [“Select Resource Path/File Window” on page 102.](#)

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.10.2.1.3 Specify Image Settings

Specify the compression and output settings for your output.

The available settings are:

Compression. Enter additional output options:

- **Data Compression.** Specify the data compression from the drop-down list.
- **Output Options.** Enter additional output options:
- **Record Length.** Select a final DPI for your output, or accept the default of 300 DPI.

Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.



Note: If you exit the wizard before specifying the required generator options, the project will load with a simple generator project file. You will need to specify the generator parameters in the Parameters window before you run your project.

For more information, see [“PDF Parser Parameters” on page 164.](#)

5.11 Web Generator



The Web Generator enables the generation of HTML output with CSS to provide a near-fidelity version of the original printstream document for viewing in a CSS compliant web browser without requiring plug-ins or software to be installed on the client desktop.

5.11.1 Web Generator Features

Features of the Output Transformation Engine Web Generator include:

- Ability to generate HTML renditions of printed documents that have a similar look and feel as paper documents without requiring browser plug-ins.
- Font mapping to available browser fonts using the MFCT.
- Images are displayed using linked PNG files.
- Substitution of black and white images with a different image, using the ImageReplace parameter. Image substitution can be used to replace images and improve the look of a document on the web. A black and white bitmap image, used for printing, can be replaced by a color image (JPEG, GIF, or PNG) for web presentation. The replacement image must be the same size and in the same orientation as the image that is being replaced.
- Vector graphics are displayed using linked JPEG files.
- Support for portrait and landscape documents.
- Simple navigation frame set to allow access to the different pages of the output.
- URL handling, in conjunction with the URL Link component. For more information, see [“URL Link Component” on page 335.](#)
- Shade mapping, automatically mapped to a gray scale based on the shade pattern.
- Support for Double-Byte Character Set fonts.
- Separate HTML files are generated for each page in the original document.
- An HTML frameset file is generated to permit easier navigation between pages.

5.11.2 Known Limitations

- Documents where fonts are shared in multiple orientations may produce unacceptable results due to the need to rasterize text that is rotated. A single PNG is created for each orientation (except portrait) in which there is text.
- Multi-color images are not currently supported.

5.11.3 Web Generator Parameters

The Web Generator parameters are used with the Web Generator.

A default set of parameters and values is used when you add a Web Generator to your project. When you work through the Web Generator Wizard, you develop parameters that are specific to your project.

To ensure that you have all of the settings you need, we recommend you review the generator's parameters in the **Properties** window before running your Output Transformation Engine project.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties** window and the message area will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the message window.

For more information about working with the Web Generator Wizard, see [“Web Generator Wizard” on page 233](#).

5.11.4 Using the Image Replacement Feature

To use the image replacement feature:

1. Set up a project using the Web Generator.
2. For the `fdImageOutput` parameter, select a basic filename, such as `{d2eOutputPath}{0}.png`
3. Run your project.
4. Open the generated web page and identify any images that need to be replaced.
5. Using Windows Explorer, do the following
 - a. Navigate to your output directory.
 - b. View the images in the output directory looking for the image(s) you identified previously in Step 4.
 - c. Take note of the image's filename. The PNG files will have names such as, `66664360402.png`.

6. Select or create a new image to replace the image you identified in Step 5.



Note: This image should be the same size and rotation as the generated one.

7. Copy the replacement image to a location on disk.



Note: Using the location of the input resources is a good idea.

8. In the **Properties** window, expand the **ImageReplace** node and do the following:

- For the **EnableImageReplacement** parameter, select **True**.
- For **FdReplacementImages**, add a new file definition to point to the location and the type of replacement image.

For example, if you want to replace 66664360402.png with 66664360402.jpg create a file definition like this:

```
{d2eAppPath}\resources\{0}.jpg
```

You can have multiple file definitions which can indicate multiple file types for different images. For example, you may want to replace some images with GIF images and some with JPEG images. In this case you would need two file definitions:

```
{d2eAppPath}\resources\{0}.jpg
```

```
{d2eAppPath}\resources\{0}.gif
```

9. Run the project and the output will contain the replaced images.



Note: You can change the `fdImageOutput` parameter at any time, but be aware that all images generated will contain at least the numeric name. The image intended for replacement should only contain the numeric name.

5.11.5 Web Generator Wizard

The Web Generator Wizard guides you through the steps to identify the output file and its characteristics. You then select web output options that are written to your project file. In this way, helps you build the generator project file as you create a project.

Launching the Web Generator Wizard

The Web Generator has a wizard associated with it to help you assemble the resources and establish the parameters for your project.

The Web Generator Wizard launches when you:

- Create a new project that produces web output. The **New Component Wizard** launches the Web Generator Wizard if the **Auto Start Generator Wizard(s)** option box is selected.

- Double-click the Web Generator on the **Project** tab.



Note: For more information, see “New Component Wizard” on page 95 or “Project Tab” on page 106.

The steps of the Web Generator Wizard are:

1. Identify your Generator.
2. Select Web Output Details.
3. Select Property Settings.

5.11.5.1 Identify Your Generator

The following options are available on the **Web Generator Wizard – 1 of 3** screen:

- **Generator name.** Enter a name for your generator or accept the default of **Web Generator**.

Since it is possible to have more than one generator running together in the same project, each generator must have a different name so that each has a unique handle. If you add generators with generic names, they will be incremented by one for each generator you add, such as Web Generator2.

- **Model generator on Web Generator in the project file.**

To change a resource path:

1. Click the ellipse button to browse to an existing project file that you want to use as a model to set your starting parameters. Leave this field blank to create a simple project file with default Web Generator values. You can make changes to any of the parameters later in the Parameters window.
2. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.



Note: If you exit the wizard at this point, and this is a new generator, you will need to enter your generator parameters before you can run your project.

5.11.5.2 Select Web Output Details

The following options are available on the **Select Web Output Details – 2 of 3** screen:

- **Output File.** Enter the path for your output directory. All of the output files will be written to this directory. If you are modelling your generator on an existing project file, that file may already contain the file path for output. If so, you do not need to specify an output directory.
- **Add or Edit a Resource Path.**

To change a resource path:

1. Select the resource by clicking on it in the list.
2. Click the **Add**, **Edit**, or **Remove** button, depending on what you wish to do:
 - Click **Add** to open the Select Resource Path/File window, where you can select the resource.
 - Click **Edit** to open the Select Resource Path/File window, where you can edit the resource.
 - Click **Remove** to remove the resource path.



Note: For more information, see “[Select Resource Path/File Window](#)” on page 102.

3. Click **Next** to proceed to the next step, or click **Cancel** to exit the wizard.

5.11.5.3 Select Property Settings

This final step sets the output and resolution details:

<Page Settings>	
<Web page name>	Enter the title of the HTML page, which is displayed in the title bar of the browser when the page is displayed.
<Page Size>	
<Width>	Enter the width of the page in pixels.
<Height>	Enter the width of the page in pixels.
<Page Settings>	
<Output DPI>	Enter resolution, in dots per inch, of the output page.
<Shade>	Select the technique to use for shading from the drop-down list:
<Native>	Uses the incoming Bitmap pattern.
<Map>	Uses the shade maps defined by the ShadeMap parameter.
<Automatic>	Attempts to calculate a grey level based on the percentage of “r;on” bits in the source shade.

Click **Finish** to save the settings, or click **Back** to change any of your previous settings. Click **Cancel** to close the wizard without saving your settings.

Chapter 6

Components

This section discusses Output Transformation Engine components.



Note: For more information about some basics about components, see *“Getting Started with Projects and Components”* on page 73.

6.1 Best Viewed Rotation Component



Best Viewed Rotation is a component that sets the rotation of a page so that the majority of text/items are displayed horizontally or in whatever manner is easiest to read. You can use either a constant rotation (when all the pages are the same), or a dynamic rotation based on the orientation of text elements or of individual characters.

6.1.1 Best Viewed Rotation Wizard

In the Best Viewed Rotation component wizard, there are two parameters to be configured:

Rotation. This dropdown menu sets the rotation of the page. The menu's values are:

- **None.** No rotation. (This is the default value.)
- **Right90.** Rotates to the right by 90°.
- **Right180.** Rotates to the right by 180°.
- **Right270.** Rotates to the right by 270°.
- **AsPerTextElement.** Rotates the page based on the most common text element orientation.
- **AsPerCharacterCount.** Rotates the page based on the most common character orientation.

Include form data. Indicates whether text elements in forms are included when finding the most common orientation. If the setting is **True**, then data from the form is included. If it is set to **False**, then the text elements will not be included. By default, this property is set to False.

6.2 Constant Print Component



The **Constant Print** component enables the following capabilities in specified locations on a page or selection of pages:

- Conditional placement of text, 1 bit-per-pixel Bitmaps or color JPEGs, backgrounds, grids, forms, other projects, or PDFs.
- Ability to emulate preprinted paper stock or add recurring elements to an existing document such as logos, forms, images of barcodes or OMR marks, text stamps or watermarks and gradients, strips of color or lines to the foreground or background of a page.

The above functionality is implemented by setting the Constant Print conditions and then using the following Constant Print operations:

- Constant Text
- Constant Image
- Constant Background Gradient
- Constant Grid
- Constant Form
- Constant Project
- Constant PDF

Insertion Types

You can place the following types of print objects using Constant Print:

- **Text.** Text stamps or watermarks, and other text that you want to have repeated on every page, like company name, date/time, or words such as *Draft* or *Confidential*.
- **Images.** Graphic files, such as logos, forms, and other recurring images.
- **Background or Foreground.** Rectangular color filled areas as strips or gradients of color, lines.
- **Grid.** A grid over the entire document. The default size is 1 inch.
- **Form.** Placement of a form in the document.
- **Project.** A single page of data from another project.
- **PDF.** A page from a Portable Document Format (PDF) file.



Note: Constant Print objects can be placed in either the background or the foreground.

Related Links

- [“Constant Text” on page 248](#)
- [“Constant Image” on page 250](#)
- [“Constant Background Gradient” on page 253](#)
- [“Constant Grid” on page 255](#)
- [“Constant Form” on page 256](#)
- [“Constant Project” on page 257](#)
- [“Constant PDF” on page 259](#)

6.2.1 Constant Print Requirements and Limitations

There are some requirements and limitations that you must be aware of before working with the Constant Print component. These requirements and limitations are catalogued below.

Requirements

In order to use the Constant Print component, you must have a license deployed that permits you use of the component. Some insertion types require further conditions be met before you can use their core functionality, see the table below:

Constant Print Insertion Type	Licensing Requirements	Component Requirements
Text	None	None
Image	None	None
Background	None	None
Grid	None	None
Form	Requires AFP Parser component	Forms must be preloaded from AFP Parser
Project	Requires components used in subproject to be licensed	Projects must be predefined (i.e. references an existing project)
PDF	None	None

Limitations

When using the Constant Print component, be aware that only some multicolor overlays are supported. Currently supported are multiple single color lines, shades, images, text, and vectors, which can be incorporated using the Constant Print Form option.

The following insertion type-specific limitations also exist:

Constant Print Insertion Type	Limitations
Text	- Text can only be on a single line (no line breaks). - Static text only (dynamic text is not supported). - Text cannot be placed on an angle.
Image	- Currently only JPEG, PNG, and BMP image formats supported. - BMP format can be either color or black and white. - JPEG and PNG formats supported in black and white only. - When using JPEG as an input format, the following Constant Image features are not supported: - Raster output, such as TIFF, PNG, and Page Viewer, including Constant Print Preview Pane. - Image scaling and rotation. - Can be used in conjunction with the Web Generator component, but only single color images are supported.
Background	PDF, PDF/A, Accessible PDF, Postscript, and TIFF support only.
Grid	None.
Form	Can only be used with the AFP Parser.
Project	Only uses the first page of the subproject output.
PDF	Only applicable to PDF output (PDF or PDF/A).

In addition, not all Constant Print insertion types are compatible with all generators. See the following table for more information.

Constant Print Insertion Type	Generator Type										
	AFP	Metacode	PCL	PostScript	TIFF	Web	Image (PNG)	Image (JPEG)	PDF	PDF/A	Accessible PDF
Text	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Image (BMP)	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes (Not readable)
Image (JPEG)	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes (Not readable)
Image (PNG)	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes (Not readable)
Background	No	No	No	Yes	Yes	No	No	No	Yes	Yes	Yes (Not readable)
Grid	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes (Not readable)
Form	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes
Project	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
PDF	No	No	No	No	No	No	No	No	Yes	Yes	Yes (Not readable)



Note: For Accessible PDFs, **Not readable** indicates that while Constant Print text can be added to the output, only text included in the original input document is readable by screen reader applications.

6.2.2 Constant Print Window

The **Constant Print** window consists of three areas:

- **Constant Print Window toolbar.** This toolbar provides control over the page preview features of the Constant Print window. For more information, see [“Constant Print Toolbar” on page 243](#).
- **Constant Print Preview pane.** This area previews your document. Refreshing the view, using the Refresh button on the toolbar, will allow you to preview your constant print job before running your project.
- **Constant Print Operations table.** This table lists the constant print operations. Some areas of the table are hidden so you can use the scroll bar at the bottom to see the full scope of the table, and resize the columns by dragging the column dividers left or right.

The columns represent the conditions that may be set for **Constant Print** objects:

- **Name.** Indicates the Constant Print condition name.
- **When.** Specifies when to place constant print.
- **Placement.** Indicates where the constant print text/image will appear; Background or Foreground.
- **Text.** Specifies if text is to be printed; True or False.
- **Image.** Specifies if an image is to be printed; True or False.
- **Background.** Specifies if a background gradient is to be printed; True or False.
- **Grid.** Specifies if a grid is to be printed; True or False.
- **Project.** Specifies if data from another project is to be printed; True or False.
- **PDF.** Specifies if a PDF page is to be printed; True or False.

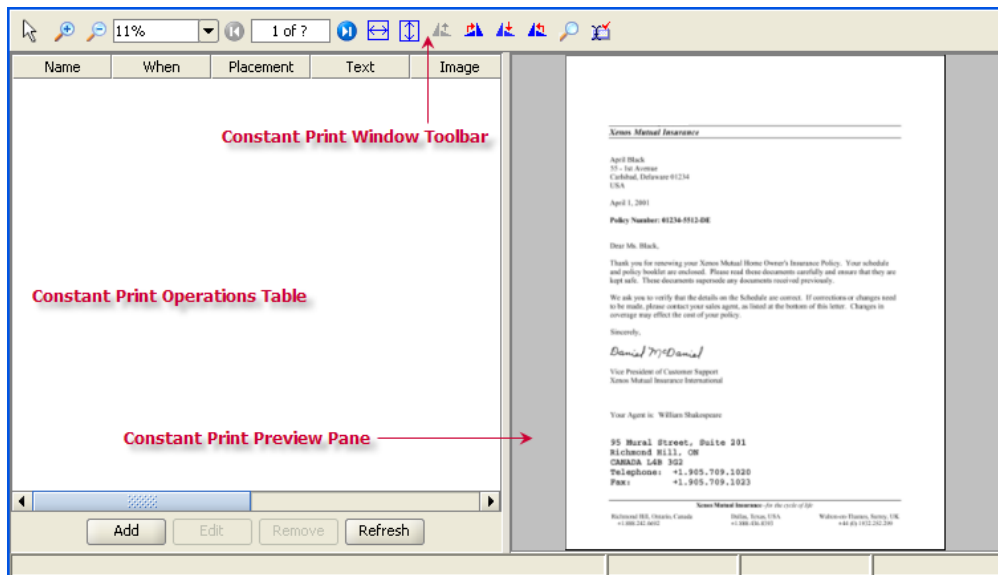


Figure 6-1: Constant Print window

6.2.2.1 Constant Print Toolbar

You can customize the way you preview the document pages in the Preview pane using the **Constant Print** toolbar.

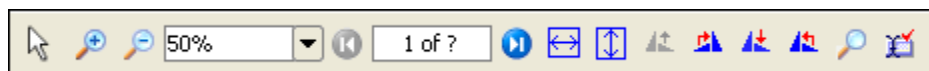


Figure 6-2: Constant Print toolbar

	Description
	Select Tool. Restores the cursor for work within the Preview pane.
	Zoom In. Changes the cursor to the Zoom In tool, which increases the document magnification view by 5% with each click in the Preview Pane.
	Zoom Out. Changes the cursor to the Zoom Out tool, which decreases the document magnification view by 5% with each click in the Preview Pane.
	Magnification. Specify the magnification at which to view the page in the Preview pane.
	Previous Page. Switches the view to the previous page.
	Go to Page. Enter a number to specify the page number to skip to.
	Next Page. Switches the view to the next page.
	Fit Page Width. Adjusts the magnification in the Preview pane so that the full width of the page can be seen.
	Fit Page Height. Adjusts the magnification in the Preview pane so that the full height of the page can be seen.
	Rotate 0° Up. Reorients the preview image to the original 0°, upright view.
	Rotate 90° Right. Reorients the preview image from 0° to the right by 90°.
	Rotate 180° Down. Reorients the preview image down by 180°.
	Rotate 90° Left. Reorients the preview image from 0° to the left by 90°.

	Find. Opens the Find dialog which allows you to search for specific text on the page.
	Page View Filter. Opens the Page View Filter dialog where you can select which element types to display on the page. Elements that can be hidden include Text, Image, Lines, Barcodes, Shading, Forms, and Vectors.

6.2.3 Constant Print Control

Adding Constant Print Operations

To implement the Constant Print functionality, a condition must be met before the Constant Print Operation is executed.

To add a Constant Print operation, click the **Add** button at the bottom of the Constant Print window. The Constant Print Operation Designer displays, which has separate tabs for each operation type.

- *<Condition.>* This is where you set the conditions that must be met to place the constant print object.
- *<Text.>* This is the tab where you set the constant text attributes.
- *<Images.>* This is the tab where you set the constant image attributes.
- *<Background.>* This is the tab where you set the constant background attributes.
- *<Grid.>* This is the tab where you set the grid attributes.
- *<Form.>* This is the tab where you set the form attributes.
- *<Project.>* This is the tab where you set the project attributes used to extract a single page of data from another project and place it into the current project.
- *<PDF.>* This is the tab where you set the PDF attributes.

6.2.3.1 Constant Print Operation Designer

The **Constant Print Operation Designer** can be displayed by selecting either **Add** or **Edit** from the Constant Print window. There are eight tabs and when opened, the **Condition** tab is shown by default.

Condition Tab

On the **Condition** tab, set the following options for the constant print:

- **Condition Name.** Type a name for this constant text condition.
- **Placement.** Select where you wish the constant text to appear:
 - *<Foreground.>* Appears in the foreground (in front of other objects).

- *<Background.>* Appears in the background (behind other objects).
- **Index Entries.** Provides configuration for conditions to be based on the content of Indexer fields. For more information, see [“Index Entries” on page 245](#).
- **Apply to Pages.** Select, from the listed options, the page(s) where you wish the constant text to appear:
 - **Every Page.** Constant Print will appear on every page of the generated document.
 - **First Page.** Constant Print will appear on only the first page of the generated document.
 - **First Odd Page.** Constant Print will appear only on the first odd number page of the generated document.
 - **Odd Pages.** Constant Print will appear on all odd number pages (1,3, 5, 7, ...) of the generated document.
 - **Even Pages.** Constant Print will appear on all even number pages (2, 4, 6, 8, ...) of the generated document.
 - **On all pages except the first.** Constant Print will appear on every page of the generated document except the first page.
 - **Last Page.** Constant Print will only appear on the last page of the generated document.
- **Apply When.** Used to insert pages with constant print items on them:
 - **On Selected Pages(s).** Constant Print items will be applied to the pages specified above.
 - **Before Selected Pages(s).** Constant Print items will be applied to a new page that is inserted in the output before the pages specified above.
 - **After Selected Pages(s).** Constant Print items will be applied to a new page that is inserted in the output after the pages specified above.
- **Page Range.** Constant print will appear on every page of the defined page range of the generated document. Select the Start (From) and End (To) pages.

6.2.3.1.1 Index Entries

The **Constant Print Index Entries** provide configuration for conditions to be based on the content of Indexer fields. The Indexer fields are defined in an upstream Indexer Component. Any field defined in an Indexer Component can be used for this condition type. When this, or any, condition evaluates to true, the Constant Print operation associated with the condition is applied. For more information, see [“Indexer Component” on page 281](#).

Content added through the use of an Index Entries based condition can be placed in:

- The foreground or the background on the page where the condition was processed.

- Before or after the page where the condition was processed.



Note: Index Entries based conditions are processed in addition to the **Apply to Pages** condition. For example, if **Apply to Pages** is set to **Odd Pages**, only odd numbered pages are processed by the Index Entries condition.

To enable the use of Index Entries, choose **Enable Index Entry Expression** from the Condition tab in the **Constant Print Operation** dialog.

To add, remove, or edit Index Entry Expressions, select the **Edit Expressions** button.

The **Index Entry Match Expressions** dialog allows you to add one or more **Index Entry Expressions** to validate one or more Index Entries.

Index Entry Match Definitions

To add, remove, or edit Index Entry Expressions, select the **Edit Expressions** button.

The **Index Entry Match Expressions** dialog window allows you to add one or more **Index Entry Expressions** to validate one or more index entries.

Index Entry Types

Index type:

- **Page.** Select this option if your Indexer Component is creating Page Index entries
- **Document.** Select this option if your Indexer Component is creating Document Index entries.

Match type:

- **MatchAny.** If any Index Expression defined as a match definition is evaluated to be true, the condition is processed. If they are all evaluated as false, the condition is not processed.
- **MatchAll.** All the Index Expressions defined as match definitions must be evaluated as true for the condition to be processed. If any Index Expression is evaluated to be false, then the condition is not processed.

Match Definitions

- **Invert Match.** If true, an expression that would normally evaluate to true evaluates to false. An expression that would normally evaluate to false, evaluates to true.
- **Entry Name.** The name of the Index Entry as defined within an Indexer Component.
- **Expression.** A Regular Expression to be applied to the Entry Name's value.

Regular Expressions

The use of regular expressions in the **Constant Print Component** is a powerful tool that enables you to process complex logic without having to write script or Java code.

A regular expression is a sequence of predefined characters used to match a pattern in a given word, phrase, paragraph, or document. Regular Expressions are commonly used to search documents for particular patterns of characters such as dates, phone numbers and zip codes.

Regular expressions vary in complexity from simple matching string literals to processing complex patterns.

For example, the expression 01234 - 5512-DE will only match the string 01234 - 5512 - DE. The same data can be processed using the expression `([0-9]{5}) - ([0-9]{4}) - ([A-Z]{2})` which will also match the input data.

The difference is the first expression will only match the given string; whereas, the second will match any string that follows the pattern: 5 digits followed by a dash followed by four digits followed by a dash followed by two uppercase letters.



Note: There are several books and online resources dedicated to regular expression usage. You should be aware that there are three general types of regular expression: DFA, NFA/POSIX, and NFA Standard. The type used in Output Transformation Engine is **NFA Standard**. Keep this in mind when looking for examples.

6.2.3.1.2 Creating a New Match Definition

To create a new match definition:

1. On the **Index Entry Match Expressions** dialog, select the **Invert match** type from the dropdown menu.
2. Enter a name in the **Entry name** field.
3. Enter a regular expression in the **Expression** field.
4. Click **New**.

The Match Definition is added to the list. Any number of Match Definitions can be added this way.

6.2.3.1.3 Editing a Match Definition

To edit an existing match definition:

1. On the Index Entry Match Expressions dialog, select a Match Definition from the list.
2. Modify the definition properties.
3. Click **Modify**.

The definition is updated in the list.

6.2.3.1.4 Removing a Match Definition

To remove a match definition:

1. Select a Match Definition from the list on the Index Entry Match Expressions dialog.
2. Click **Remove**.

The definition is removed.



Caution

There is no delete confirmation dialog or undo for this operation, so be extremely careful when deleting match definition.

6.2.3.2 Constant Text

The Constant Text operation enables the emulation of preprinted paper stock or the addition of recurring text to an existing document.

Adding and Editing Constant Text

Prior to configuring Constant Text, you must ensure that you have set the **Constant Print Conditions** on the **Conditions** tab. To add or edit constant text:

1. In the **Constant Print Operation Designer**, select the **Text** tab.
2. Do one of the following:
 - To add new text, click the **Add** button. The **Constant Print - Text Window** appears.
 - To edit existing text, select the text from the Text Items list, and click the **Edit** button. The **Constant Print - Text Window** appears.
3. Set your Constant Text options, and click **OK**. (For more information on the Constant Print Text options, please see “**Constant Print - Text Window**” on page 249.)

The entry is added to the **Text Items** list.

Removing Constant Text

To remove Constant Text:

1. In the **Constant Print Operation Designer**, select the **Text** tab.
2. Select the text you want to delete from the Text Items list, and click the **Remove** button.
3. You must confirm the removal of your Constant Print Text entry on the **Delete Constant Print Text** confirmation dialog.

If you click **Yes**, the constant text is removed from the list.

6.2.3.2.1 Constant Print - Text Window

To add some constant print text:

1. In the **Constant Print** window, click **Add** and the Constant Print Operation Designer dialog appears.
2. Switch to the **Text** tab.
3. Set the following:

- **Text to Appear on Page.** Type the constant text to appear.



Note: To enter multiple lines of constant text, you must add each line of constant text as a separate constant print object.

- **Font.** Set the following font information:
 - **Font name.** Select an output font from the drop-down list.
 - **Point size.** Select a point size by clicking the arrows or typing a numeric point size.
 - **Color.** Select a color from the drop-down list. Select **(New...)** to create a new color.
- From the **Add Color** dialog window, set the following:
 - a. Enter a **name** for your color.
 - b. Click **Edit** in the **Color box** to select your color from the Color Picker window.
 - c. Click **OK** to add the color, or click **Cancel** to exit without adding a color.
- **Position.** Set the following positioning information:
 - **X-position.** The number of DPI from the left edge of the page.
 - **Y-position.** The number of DPI from the top edge of the page.
 - **Dots per inch.** Set the DPI by clicking the arrows or typing a numeric DPI. This is the resolution used by the X and Y positions. The default is 300 DPI.

- **Orientation.** Select from the preset orientation settings. Select a **Print Direction**. The sample box will display a sample of the direction and orientation of the text depending on the selection you make.
4. Click **OK** to accept the settings or **Cancel** to discard them and return to the Constant Print window.

Related Links

- [“Output Fonts” on page 535](#)
- [“Adding a Color to a Color Table” on page 550](#)
- [“Color Picker Window” on page 261](#)

6.2.3.2.2 Constant Print – Text Properties

A default set of parameters and values is used when you add this component to your project.

Only a limited number of parameters are discussed here so to ensure that you have all of the settings you need, we recommend you review the component’s parameters in the **Properties** window before running your project.

- **FontNameType.** Defines what the FontName parameter refers to in the MFCT font mappings. You can choose from **Output** or **OpenType**.
If using Output, the FontName value should match the name of one of the Output Font mappings in the MFCT.
If using OpenType, the FontName value should match the name of one of the OpenType Input Font mappings in the MFCT.
- **FontName.** Defines the font name to use when creating text elements. This value must match the name of an Output Font or OpenType Input Font in the MFCT parameter file based
If you are using an OpenType Input Font name, you must ensure the OpenType source font is defined in the Font Environment Map and there is a corresponding OpenType Input Font mapping defined in the MFCT.

6.2.3.3 Constant Image

The Constant Image operation enables conditional placement of 1 bit-per-pixel Bitmaps or color JPEGs in specified locations on a page or selections of pages. This is often used to emulate preprinted paper stock or add recurring images to an existing document.

Adding a Constant Image

Prior to configuring Constant Images, you must ensure that you have set the **Constant Print Conditions** on the Conditions tab. To add or edit a constant image:

1. In the **Constant Print** window, click **Add**. Then, when the Constant Print Operation Designer appears, select the **Images** tab.

2. The path must be set to the location of your images. Under the **Image Paths** section, perform one of the following:
 - To add a new image path, select the **Constant Image Paths** icon, and then the **Add** button. The **Select Resource Path/File** dialog opens.
 - To edit an existing image path, select the path from the **Image Paths** list, and click the **Edit** button. The **Select Resource Path/File** opens.
3. In the Images section, perform one of the following:
 - To add a new image, select an image from the **Image Paths** list at the bottom to enable the **Add** button in the Images section. Click **Add**, and the **Constant Print – Image window** opens.
 - To edit an existing image, select an existing path and click the **Edit** button to change the selected image. Change the settings as required on the **Constant Print – Image window**.
4. Click **OK** to accept the settings, or **Cancel** to discard them and return to the Constant Print window.

Removing a Constant Image

To remove a constant image:

1. In the **Constant Print Operation Designer**, select the **Images** tab.
2. Select the file path of the image you want to delete from the **Image Paths** list, and click **Remove**.
3. Confirm the removal of your Constant Print Image entry.
If you click **Yes**, the constant image is removed from the list.

6.2.3.3.1 Constant Print - Image Window

On the **Images** tab of the Constant Print Operation Designer, you can view the settings available by selecting an image from the **Image Paths** section and then clicking **Add** in the **Images** section. In the **Constant Print - Image dialog**, you may set the following image settings:

- **Image**. Set the image details:
 - **File name**. The file name of the constant image to appear.
 - **Type**. Select the type of image from the drop-down list:
 - **BMP**. Monochrome Bitmap images.
 - **JPG**. JPEG images. **Limitations** are:
 - No support for raster output such as: TIFF, PNG, and the Page Viewer, including the Constant Print Preview Pane.

- No scale.
- No rotation.
- **Scale by.** Select the percentage by which you want to scale the image by clicking the arrows or typing a value.
- **Color.** Select a color from the drop-down list. Select **(new...)** to create a new color.
- **DOM Properties**
 - **Alternate Text.** Defines the alternate text to use for the Constant Print image. To enable this parameter, you must set the alternate text source under the XFT Object or Auto Detect properties to **IMAGE_SOURCE**.
 - **Language.** Indicates the language code as defined by ISO-639 for the alternate text value.

From the **Add Color** dialog window, set the following:

- Enter a **name** for your color.
- Click the **Edit** button in the **Color box** to select your color from the Color Picker window.
- Click **OK** to add the color, or click **Cancel** to exit without adding a color.
- **Rotation.** Select a rotation degree and direction.
- **Position.** Set the following positioning information:
 - **X-position.** The number of DPI from the left edge of the page.
 - **Y-position.** The number of DPI from the top edge of the page.
 - **Dots per inch.** Set the DPI by clicking the arrows or typing a numeric DPI. This is the resolution used by the X and Y positions. The default is 300 DPI.

Related Links

- [“Constant Print Operation Designer” on page 244](#)
- [“Page View Tab” on page 90](#)
- [“Adding a Color to a Color Table” on page 550](#)
- [“Color Picker Window” on page 261](#)

6.2.3.4 Constant Background Gradient

The Constant Background operation enables the emulation of preprinted paper stock by adding a color gradient to an existing document.

Adding a Constant Background Gradient

Prior to configuring Constant Background Gradients, you must ensure:

- You have set the **Constant Print Conditions** on the Conditions tab.
- On the **Background Tab**, you have selected the **Apply background settings** check box to enable background settings.

To add a constant background gradient:



Note: Your settings will be saved if you set the parameters and then decide to turn off this option.

1. In the **Constant Print** window, click **Add** and the Constant Print Operation Designer dialog appears.
2. Switch to the **Background tab** and select **Apply background settings** to enable background settings, if necessary.
3. Set the following; as you set each setting the preview box at the bottom of the window will show you what the gradient will look like:
 - **Start Color.** Select a color for the start of the gradient by setting the RGB value or using the color picker by clicking the **Edit** button.
 - **End Color.** Select a color for the end of the gradient by setting the RGB value or using the color picker by clicking the **Edit** button. Select the same color as start for a solid background color.
 - **Location.** Set the position of the background gradient by setting the location, in inches, from the four sides of the gradient. This is similar to setting margins on a page.
 - **Appearance.** Set the appearance of the background gradient by setting the following:
 - **No. of Steps.** Set the number of steps from the start to the end. The higher the number the smoother the gradient.
 - **Style.** Hue or Lum (luminance).
 - **Direction.** Set the direction of the color gradient's start to end points.
4. Click **OK** to accept the settings or **Cancel** to discard them and return to the Constant Print window.

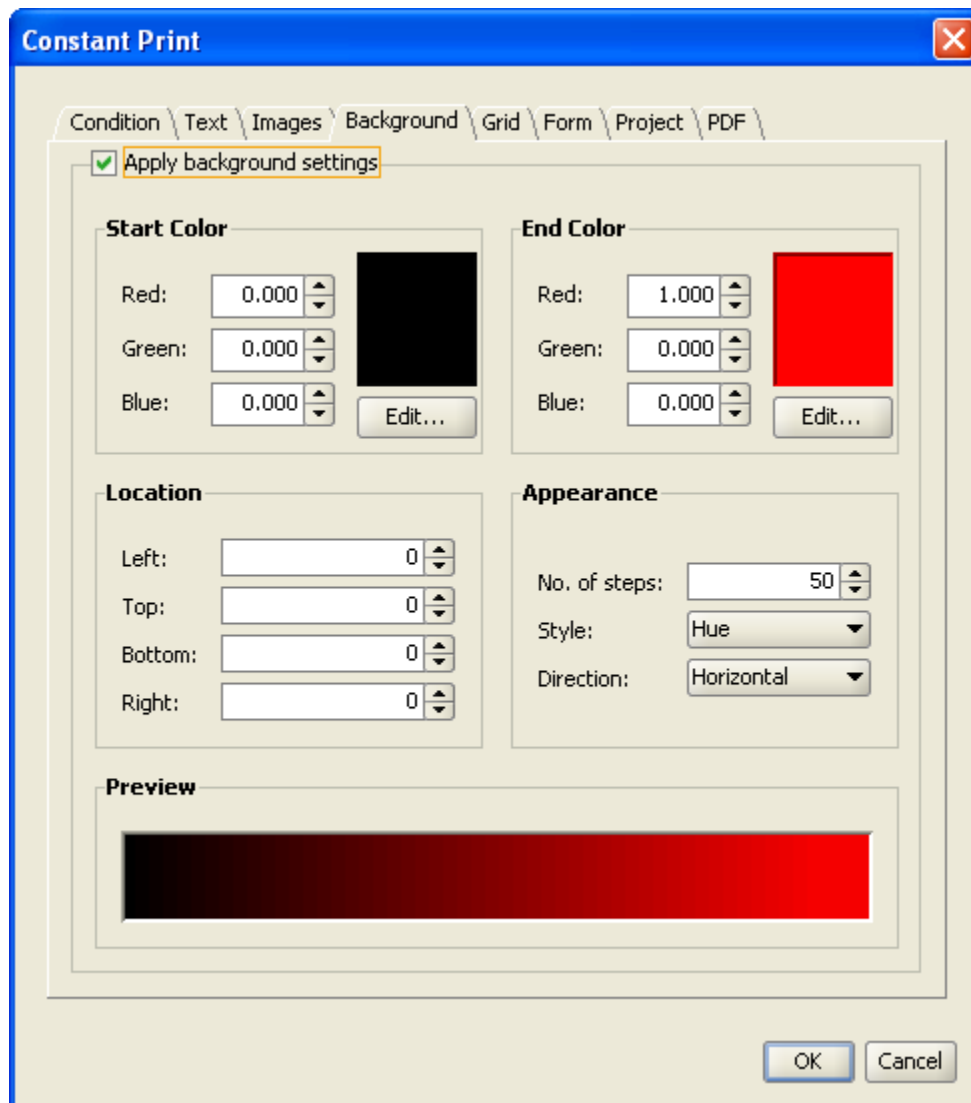


Figure 6-3: Constant Print dialog

Related Links

- [“Color Picker Window” on page 261](#)
- [“Constant Print Window” on page 241](#)

6.2.3.5 Constant Grid

The Constant Grid operation enables the emulation of preprinted paper stock by adding a grid to an existing document.

Adding a Constant Grid

Prior to configuring Constant Grids, you must ensure that you have set the **Constant Print Conditions** on the Conditions tab, and on the **Grid Tab**, select the **Apply grid settings** check box to enable grid settings. To add a constant grid:



Note: The settings you configure will be saved if you set the parameters and then decide to turn off this option.

1. In the **Constant Print** window, click the **Add**. Then, when the Constant Print Operation Designer appears, select the **Grid** tab.
2. On the **Constant Print Grid tab**, set the following:
 - **Grid Start Point.** Select where the grid will start, aligning the grid with the bottom edge or the top edge of the page.
 - **Color.** Select a color from the drop-down list. Select (**New...**) to create a new color.
From the **Add Color** dialog window, set the following:
 - a. Enter a **name** for your color.
 - b. Click the **Edit** button in the **Color box** to select your color from the Color Picker window.
 - c. Click **OK** to add the color, or click **Cancel** to exit without adding a color.
 - **Units.** Select the units used to size the grid.
 - **Grid Size in Units.** Select a grid size by clicking the arrows or typing a numeric grid size. The units used are set under the units section.
3. Click **OK** to accept the settings or **Cancel** to discard them and return to the Constant Print window.

Related Links

- [“Adding a Color to a Color Table” on page 550](#)
- [“Color Picker Window” on page 261](#)
- [“Constant Print Window” on page 241](#)

6.2.3.6 Constant Form

The **Constant Form** operation enables the emulation of preprinted paper stock by adding a form to the background of an existing document.

Adding a Constant Form

Prior to configuring constant forms:

- Ensure you have set the **Constant Print Conditions** on the Conditions tab.
- On the **Form Tab**, select the **Apply form settings** check box to enable form settings.

To add a constant form:

1. In the **Constant Print** window, click **Add**. Then, when the Constant Print Operation Designer appears, select the **Form** tab.
2. On the Constant Print **Form tab**, set the following:
 - **Form**. Type the name of the form in the text box. This name must be the same as one of the overlays specified in the AFP Parser's External Resources Parameter.
 - **New Page Size**. Select a page width and height by clicking the arrows or typing a size.



Note: Setting the page size width and height to 0 will result in a new page defaulting to the current page's width and height.

- **Rotation**. Select a rotation degree and direction.
 - **Position**. Set the following positioning information:
 - **X-position**. The number of DPI from the left edge of the page.
 - **Y-position**. The number of DPI from the top edge of the page.
 - **Dots per Inch**. Set the DPI by clicking the arrows or typing a numeric DPI. This is the resolution used by the X and Y positions. The default is 300 DPI.
3. Click **OK** to accept the settings or **Cancel** to discard them and return to the Constant Print window. For more information, see [“Constant Print Window” on page 241](#).

6.2.3.7 Constant Project

The **Constant Project** is used to extract a single page of data from another project and place it into the current project. This is typically done to insert a watermark, banner page or custom form into an existing printstream.

Adding a Constant Project

Prior to configuring constant forms:

- Ensure you have set the **Constant Print Conditions** on the Conditions tab.
- On the **Project** tab, select the **Apply project settings** check box to enable project settings.

To add a constant project:



Note: The settings you configure will be saved if you set the parameters and then decide to turn off this option.

1. In the **Constant Print** window, click **Add** and the Constant Print Operation Designer dialog appears.
2. Switch to the **Project** tab and select **Apply project settings** to enable the settings, if necessary.
3. Then, set the following:
 - **Project.** Select one of the buttons on the right to set the location of the project you wish to use as constant content:
 - **Add.** Selects a new project location.
 - **Edit.** Modifies the location of an existing project.
 - **Remove.** Deletes the location of a selected project.



Note: Only one location can be specified. To use multiple projects, define as many Constant Project Print Operations as you need.

- **New Page Size.** Defines the size of the page that is created to contain the constant content from the selected project. Select a page width and height by clicking the arrows or typing a size.

If the constant content is to be inserted:

- **As a new page,** the page size is defined by the width and height chosen.
- **Into an existing page,** the width and height define the size of the form used to hold the constant content.



Note: Depending on the output type, it may not be possible to arbitrarily choose the page size.

- **Rotation.** Select a rotation degree and direction. Rotations pivot on the top left corner of the inserted content. The four possible rotations are:
 - **None**
 - **90° to the left**
 - **90° to the right**
 - **180°**

If a rotation other than none is chosen, it is likely that a position other than (0,0) will have to be applied to get the content to the desired position.

- **Position.** The coordinate of the top left corner of the image before rotation is applied. Set the following positioning information:
 - **X-position.** The number of DPI from the left edge of the page.
 - **Y-position.** The number of DPI from the top edge of the page.
 - **Dots per inch.** Set the DPI by clicking the arrows or typing a numeric DPI. This is the resolution used by the X and Y positions. The default is 300 DPI.

The **DPI** parameter allows you to choose a coordinate system that is convenient for your calculations. For example:

- If you are working in inches, a DPI of 1 can be chosen. All coordinates will represent whole inches.
- If a DPI of 2 is chosen all coordinates will represent half inches.



Note: The DPI parameter affects both the Position parameter and the PageSize parameter.

4. Click **OK** to accept the settings or **Cancel** to discard them and return to the Constant Print window. For more information, see “**Constant Print Window**” on page 241.

Special Considerations

Constant Print uses the first page it comes across from the chosen project. If you want to use a page other than the first page, the best way to do this is to set the **StartPage** parameter of the project you are using to match the page you wish to use as constant content.

To assist with DOM merging for Constant Print projects, the **MergeDOM** parameter indicates when DOM from a Constant Project PDF is automatically merged with a target PDF DOM in Accessible PDF Generator outputs. By default, this is set to true. If false, DOM merging must be manually performed using the Merge DOM API, which can be implemented in the Custom or Script components.



Note: If there is no DOM present in the Constant Print item to add to the page, then this parameter is ignored.

Furthermore, when performing DOM merging of a Constant Print project and a document, be aware the Constant Project's Stop Page is automatically dynamically overwritten to be the same as the Start Page during processing. This is done to prevent an exception due to the DOM from multiple pages not finding the corresponding elements on the single merge page from being thrown by the component.

Setting Up a Project To Use with Constant Project

Any fully functioning project can be used with Constant Project. Set up this project the same way as any other project. When this project is used as Constant Print, only the parser is considered; all other components in the project are ignored.

6.2.3.8 Constant PDF

The **Constant PDF** operation enables the emulation of preprinted paper stock by adding a page from a Portable Document Format (PDF) file to the background of an existing document. It also enables the insertion of a range of pages from a PDF file into the output PDF file as new pages.

Adding a Constant PDF

Prior to configuring constant forms:

- Ensure you have set the **Constant Print Conditions** on the Conditions tab.
- On the PDF tab, select the **Apply PDF settings** check box to enable project settings.

To add a constant PDF:

1. In the **Constant Print** window, click **Add**. Then, when the Constant Print Operation Designer appears, select the **PDF** tab.
2. On the **PDF** tab, set the following:
 - **PDF**. Select one of the buttons on the right-hand-side to set the location of the PDF file you wish to use as constant content:
 - **Add**. Selects a new PDF file location.
 - **Edit**. Modifies the location of an existing PDF file.
 - **Remove**. Deletes the location of a selected PDF file.



Note: Only one location can be specified. To use multiple projects, define as many Constant PDF Print Operations as you need.

- **Options**.

- **Scale By Percentage.** Scales the PDF page on the output document by the specified percentage. A value of:
 - **100.** Keeps the original size.
 - **Greater than 100.** Increases the size of the PDF in the output document.
 - **Less than 100.** Reduces the size of the PDF in the output document.
- **Start page.** Indicates the first page number in a page range from the input PDF to use for extracting constant PDF pages. The numeric value must be:
 - greater than or equal to **1**, and
 - less than or equal to **Stop page**

A page number greater than the total number of pages in the PDF will result in the last page of the PDF being used.
- **Stop page.** Indicates the last page in a page range from the input PDF to use for extracting constant PDF pages. The numeric value must be:
 - greater than or equal to **Start page**, or
 - equal to **-1** (the default value) or **0**, meaning all pages starting from, and including, **Start page** are processed.

A page number greater than the total number of pages in the PDF will result in the last page of the PDF being used.
- **Rotate in place.** Ensures the position you have chosen takes any rotation and scale into consideration. If the PDF page you have chosen is larger than it appears, there is a crop box for example, this may not produce the desired location on the page. In a case like that, turn off **Rotate in place** and update the position to account for rotations and scale.
- **Rotation.** Select a rotation degree and direction. Rotations pivot on the top left corner of the inserted content. The four possible rotations are:
 - **None**
 - **90° to the left**
 - **90° to the right**
 - **180°**

If a rotation other than **None** is chosen, it is likely that a position other than (0,0) will have to be applied to get the content to the desired position.
- **Position.** The coordinate of the top left corner of the image before rotation is applied. Set the following positioning information:
 - **X-position.** The number of DPI from the left edge of the page.
 - **Y-position.** The number of DPI from the top edge of the page.

- **Dots per inch.** Set the DPI by clicking the arrows or typing a numeric DPI. This is the resolution used by the X and Y positions. The default is 300 DPI.

The **DPI** parameter allows you to choose a coordinate system that is convenient for your calculations. For example:

- if a DPI of **2** is chosen all coordinates will represent half inches.
if you are working in inches, a DPI of **1** can be chosen. All coordinates will represent whole inches.



Note: The DPI parameter affects both the **Position** parameter and the **PageSize** parameter.

3. Click **OK** to accept the settings or **Cancel** to discard them and return to the Constant Print window. For more information, see “**Constant Print Window**” on page 241.

6.2.3.9 Color Picker Window

Select a Color

You must select the color parameters to define a color to use in the Color Table. The Output Color function is available when adding or editing an entry in the Color Table. To pick a color, do **one** of the following:

- On the **Swatches** tab, select the color from the predefined color swatches.
- On the **HSB** (Hue, Saturation, and Brightness) tab, move the slider on the **Luminosity** bar, and click on an area of the **Color Chart** to choose a new color. This method offers a more accurate color selection process.
- On the **RGB** tab, use either the sliders for each color or enter a value into the field to complete a set of RGB values.
- The **Preview** area at the bottom will change dynamically to show you an example of the currently selected color.

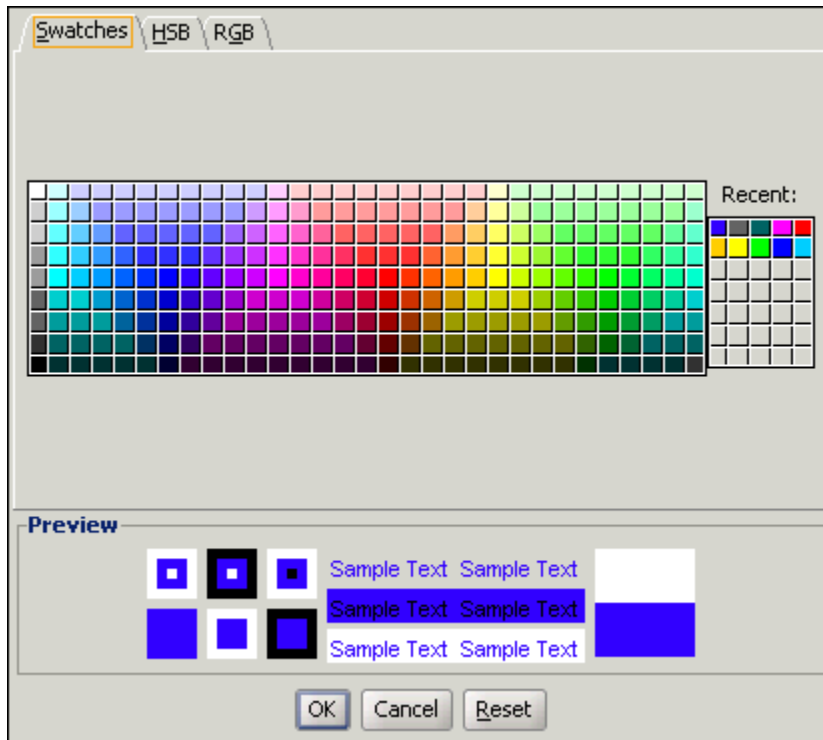


Figure 6-4: Color Picker

6.3 Document Breaking Component



The **Document Breaking** component separates the printstream into individual documents based on logic conditions specified by configuring the parameters:

- Page count
- Sections (triggers, headings, documents, etc.)
- Field information (i.e., new account number)



Note: The Document Breaking Component controls document breaking and naming, which can also be done using the Indexer Component. For more information, see *“Indexer Component” on page 281*.

6.3.1 Document Breaking Wizard

The **Document Breaking Wizard** assists in the customization of its parameters, and has two tabs:

- Document Breaking tab
- Document Naming tab



Note: Document breaking and naming can also be done using the Indexer Component. For more information, see [“Indexer Component” on page 281](#)

6.3.1.1 Document Breaking Tab

The options on the Document Breaking tab allow you to set the mode of document breaking to use. There are several modes which can be set:

- **Page count.** A document break is triggered after the specified number of pages set in the **Number of Pages** field.
- **Page groups.** Denotes a list of regular expressions for the Named Page Groups to match. By default, document breaking occurs for every page group, however, if you want to perform document breaking for only specific page group names then you can add regular expressions matching the desired page group names to this list. If you want to match all page groups, leave the list empty.
- **XFT document change.** A document break is triggered when the XFT document changes.
- **XFT section change.** A document break is triggered when the XFT section changes.
- **XFT field change.** A document break is triggered when the XFT field changes value. You must identify the field that triggers the break.
- **XFT field equals.** A document break is triggered when the specified XFT field equals a certain value. You must identify both the field and the value that triggers the break.

Identifying Document Break Trigger Fields

To identify the field that triggers the document break:

1. Click on the **XFT field change** ellipsis button, and the **Field** dialog appears showing all available fields in the document.
2. Select the field name from the list, and click **OK**.
The name appears in the **Field name** box.

Identifying Document Break Trigger Fields and Values

To identify the field and value that triggers the document break:

1. Click on the **XFT field equals** ellipsis button, and the **Field** dialog appears showing all available fields in the document.
2. Select the field name from the list, and click **OK**.
3. You are brought back to the Document Breaking tab, and the name appears in the **Field name** box.
4. In the **Value** field, type the value that the field must match in the order for the document break to occur.
 - **Indexer match.** A document break is triggered when the indexer condition matches.
 - **Indexer entry change.** A document break is triggered when indexer entries change.



Note: Document breaking and naming can also be done using the Indexer Component. For more information, see “[Indexer Component](#)” on page 281.

6.3.1.2 Document Naming Tab

On Document Naming tab, you set how the document will be named. You can mix text with any of the following variables to produce a document file name:

- **{PAGENO}** - The page number within the printstream.
- **{DOCNO}** - The sequential document number.
- **{SECTION}** - The name of the current XFT section.
- **{DOC}** - The name of the current XFT document.
- **{FLD:xxx}** - The value of the XFT field 'xxx'. If your project has XFT fields created by the XFT Component, then they will appear in this category.

To generate a document file name:

1. Click on the variable type you want to use from the list, and click the **Insert** button to add the variable type to the **Document name** field.
2. More than one variable type can be added, so repeat this process as necessary.

Ensure that the document names generated will be unique for each document. If this is blank then {DOCNO} will be used by default.

Preserving the Logical Document Break

Selecting the **Include source and logical document breaks** check box causes the logical document breaks in the source file to be preserved along with the document breaks added by your configuration of this component.



Note: Document breaking and naming can also be done using the Indexer Component. For more information, see “[Indexer Component](#)” on page 281.

6.4 Field Modifier Component

The **Field Modifier** component alters either text-based or graphical XIF objects such as text, images, lines, shades, vectors, and barcodes. These types of objects can either be moved or completely deleted during processing.

6.4.1 Field Modifier Window

The Field Modifier window consists of five areas:

- **Toolbar.** Provides button shortcuts to menu items, as well as tools to help you customize the view of the page. For more information, see “[Field Modifier Toolbar](#)” on page 267.
- **View Point Selection list.** Allows you to quickly switch between the viewers for the components in your project flow. Components that do not have a custom viewer will open in the Page View tab instead.
- **Design pane.** Displays the field modifier entries you currently have configured.
- **Preview pane.** Presents a graphical representation of the file, on which you can draw the sections, fields, and locators. Refreshing the view, using the **Refresh** button on the toolbar in the design pane, allows you to preview your job before running your project.
- **Status bar.** Displays the cursor coordinates (X, Y), representing the cursor location relative to the coordinates of the image rather than the screen. As such, each pixel on the screen may represent several pixels in the image. Also, the current rotation factor being shown is presented.

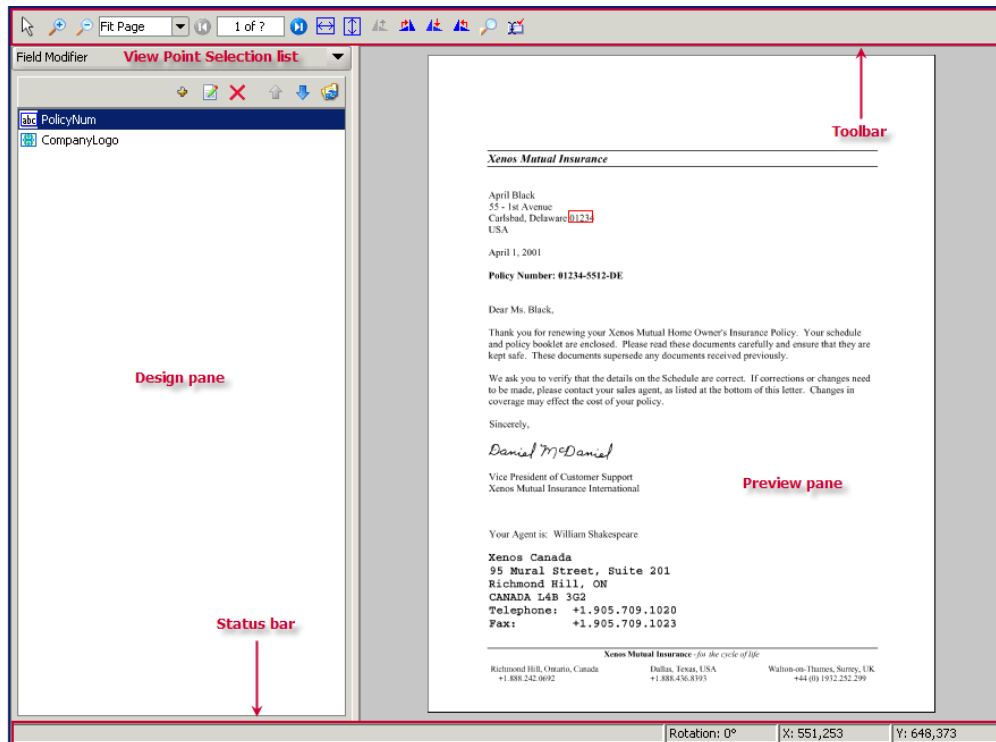


Figure 6-5: Field Modifier window

6.4.1.1 Field Modifier Design Pane

The Design pane is where the bulk of the work for the Field Modifier component is performed. Any Field Modifier entries you currently have configured, regardless of whether they are active or not, will appear in the list located in this pane. If you hover the cursor over the name of an entry, a dialog box appears showing a summary of the entry's current settings.

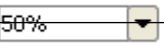
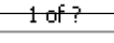
There are several buttons located across the top:

- **Add.** Adds a new Field Modifier entry to the list. For more information, see [“Adding a Field Modifier Entry” on page 269](#).
- **Edit.** Edits an existing Field Modifier entry.
- **Delete.** Removes an existing Field Modifier entry.
- **Up.** Moves the selected entry higher up in the list order.
- **Down.** Moves the selected entry down in the list order. Setting a particular order to the modifications performed provides the advantage of being able to achieve multiple adjustments to a single field during the same job run.
- **Refresh.** Refreshes the image in the Preview pane with the latest version.

 **Tip:** You can also trigger a refresh of the current page view by saving the project. Either click **Save** or **Save All** on the toolbar, or simply type **Ctrl + S**.

6.4.1.2 Field Modifier Toolbar

You can customize the way you preview the document pages in the Preview pane using the **Field Modifier** toolbar.

	Description
	Select Tool. Restores the cursor for work within the Preview pane.
	Zoom In. Changes the cursor to the Zoom In tool, which increases the document magnification view by 5% with each click in the Preview pane.
	Zoom Out. Changes the cursor to the Zoom Out tool, which decreases the document magnification view by 5% with each click in the Preview pane.
	Magnification. Specify the magnification at which to view the page in the Preview pane.
	Previous Page. Switches the view to the previous page.
	Go to Page. Enter a number to specify the page number to skip to.
	Next Page. Switches the view to the next page.
	Fit Page Width. Adjusts the magnification in the Preview pane so that the full width of the page can be seen.
	Fit Page Height. Adjusts the magnification in the Preview pane so that the full height of the page can be seen.
	Rotate 0° Up. Reorients the preview image to the original 0°, upright view.
	Rotate 90° Right. Reorients the preview image from 0° to the right by 90°.
	Rotate 180° Down. Reorients the preview image down by 180°.
	Rotate 90° Left. Reorients the preview image from 0° to the left by 90°.
	Find. Opens the Find dialog which allows you to search for specific text on the page.

	Page View Filter. Opens the Page View Filter dialog where you can select which element types to display on the page. Elements that can be hidden include Text, Image, Lines, Barcodes, Shading, Forms, and Vectors.

6.4.2 Field Modifier Operations

As a prerequisite to configuring the **Field Modifier** component, you must place it in a project flow downstream from the **XFT Field Technology** component with either **text** or **object fields** properly setup for the elements you want to change with the Field Modifier component.



Note: For more information about setting up text or object fields in the XFT Field Technology component, see [“Adding a Text Field” on page 393](#) or [“Adding an Object Field” on page 401](#).

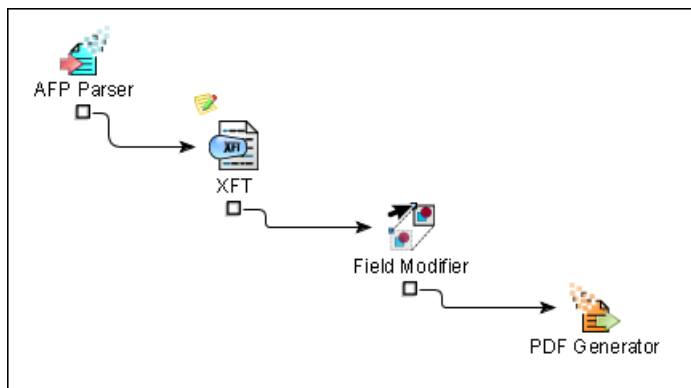


Figure 6-6: Sample project with Field Modifier component present

When working with the Field Modifier component, you can add, edit, or remove a field modifier, and set the field modifier entry list run order.

Risks and Limitations

With the XFT Field Technology component, which is where the Field Modifier component will retrieve the field information, you have the option to configure text and object fields to capture page objects that are located inside forms used by the page. This can be performed by selecting the **Include Form Data** option when configuring your XFT text and/or object fields. However, you must be aware of the risks and limitations that exist when you use Field Modifier to edit the contents of a form. Primarily, it is important to understand that when elements in a form are changed, every page that uses that form is also likely to be affected by the change. In configurations that use multi-threaded and/or concurrent jobs, careful analysis, caution, and sufficient testing should be exercised to ensure that your executed projects will always behave as intended.

6.4.2.1 Adding a Field Modifier Entry

To add a new text or object field to the modifier list:



Note: You must ensure that you have your text or object fields properly established with the component before working with the Field Modifier, otherwise no fields can be located for conversion. For more information about setting up text or object fields in the component, see [“Adding a Text Field” on page 393](#) or [“Adding an Object Field” on page 401](#).

1. In the Development window, double-click the **Field Modifier** component in your project.
The **Field Modifier** screen appears.
2. You must add the text or object element that you want to make changes to. In the Design pane, click the **Add** button.
The **Add Field Modifier Entry** dialog appears.
3. You must enter information in the following fields:
 - **Entry name.** Indicates a unique name with which to identify the field.
 - **Field type.** Specifies the class of the field you want to change. Using the dropdown menu, select either **Text** or **Object**.
 - **Document, Section, and Field.** Indicates the field you configured in the component that you want to modify. Using these three dropdown menus, locate the specific field.
 - **Is RegEx.** Specifies whether the page, section, and field names are processed as regular expressions instead of literal strings.
4. In the **Action Type** dropdown menu, select the operation to perform on the field. You can select from **Move**, **Delete**, **Redact**, or **NOP** (No Operation).
5. Depending on the action type you have selected, the fields in the **Action Information** section change. You must set your action options accordingly:
 - **Move.**
 - **Enabled.** Indicates whether or not the particular field modifier entry is active. The primary purpose of this flag is to allow you to preserve the properties of an action, while disabling it at runtime. Disabling an entry can be convenient during the project development phase. Enabling and disabling an action can also be modified at runtime using the API, which is useful when implementing custom business logic.
 - **Move Type.** Specifies the type of movement the field will undergo. **Relative** indicates that the moves are calculated in relation to the field’s current position, whereas **Absolute** indicates that the X and Y coordinates mark the object’s specific location on the page.
 - **Reference Point.** Controls the starting point for the Absolute move type. You can select from either **TopLeft** or **BottomLeft** as your points of origin

from which Absolute moves are calculated. Using a BottomLeft absolute position is useful for situations where you want to reposition a text field, like an address block, with a bottom justified location.

- **Position X.** Differs depending on the **Move Type** selected. For **Relative**, this field indicates how many pixels the object should be moved up or down. For **Absolute**, this field indicates the number of DPI from the location you selected in Reference Point.
 - **Position Y.** Differs depending on the **Move Type** selected. For **Relative**, this field indicates how many pixels the object should be moved left or right. For **Absolute**, this field indicates the number of DPI from the location you selected in Reference Point.
 - **Delete.**
 - **Enabled.** Indicates whether or not the particular field modifier entry is implemented. The primary purpose of this flag is to allow you to preserve the properties of an action, while disabling it at runtime. Disabling an entry can be convenient during the project development phase. Enabling and disabling an action can also be modified at runtime using the API, which is useful when implementing custom business logic.
 - **Delete Field Event.** Specifies whether the XFT field event will also be deleted. If it is deleted, it will not be passed onto components that are downstream in the project flow. By default, this parameter is **True**.
 - **Enabled.** Indicates whether or not the particular field modifier entry is active. The primary purpose of this flag is to allow you to preserve the properties of an action, while disabling it at runtime. Disabling an entry can be convenient during the project development phase. Enabling and disabling an action can also be modified at runtime using the API, which is useful when implementing custom business logic.
 - **Delete Field Event.** Specifies whether the XFT field event will also be deleted. If it is deleted, it will not be passed onto components that are downstream in the project flow. By default, this parameter is **True**.
- NOP.** Indicates **No Operation**, or rather, that no action should be taken. The NOP action is most useful for development and debugging purposes.
- **Enabled.** Indicates whether or not the particular field modifier entry is implemented.
6. When you have completed setting the configuration, click **OK**.
The main **Field Modifier** screen appears.
 7. **Save** your settings.

6.4.2.2 Editing a Field Modifier Entry

To edit an existing Field Modifier entry:

1. In the Design pane of the Field Modifier window, select the entry you want to change and click the **Edit** button on the toolbar.

The **Edit Field Modifier Entry** dialog appears.

2. Edit the options for the Field Modifier entry as required. For more information on each setting's function, see [“Adding a Field Modifier Entry” on page 269](#).



Note: All options can be changed with the exception of the Entry Name, which is permanent once it is set during the Field Modifier entry's creation.

3. When you are finished, click **OK**.

The Field Modifier window displays.

6.4.2.3 Removing a Field Modifier Entry

To remove a Field Modifier entry:

1. In the Design pane of the Field Modifier window, select the entry you want to delete.

2. Click the **Remove** button.

The Field Modifier entry is removed.



Caution

There is no delete confirmation dialog or undo for this operation, so be extremely cautious when deleting a Field Modifier entry.

6.4.2.4 Setting the Field Modifier Entry List Run Order

Field Modifier entries are carried out in succession, in the order they are shown in the Design pane. Setting a particular order to the modifications performed provides the advantage of being able to achieve multiple adjustments to overlapping fields or even a single field during the same job run. For instance, you could delete a text field and then subsequently move an entire block of text from elsewhere in the document into the same area where the text field appeared.

To reposition a Field Modifier entry:

1. In the Design pane of the Field Modifier window, select the entry you want to move.
2. Click the **Up** or **Down** buttons accordingly, until you have put the entry in a logical place in the list.

3. **Save** your settings.

6.5 Index Writer Component



The **Index Writer** capability is implemented through a number of Index Writers. Index Writers write out the contents of the Indexer in various formats. The following Index Writers are provided:

- A standard/generic Index Writer, which writes out indexes in a standard format.
- IBM Content Manager OnDemand (CMOD) Index Writer, which writes out a file in IBM CMOD format.
- XML Index Writer, which writes out indexes as an XML file using a fixed format.
- FileNet Index Writer, which creates FileNet index files for FileNet archives.
- Custom Index Writer, which allows the ability for OpenText Professional Services or your developers to create a writer to suit your needs.

6.5.1 Index Writer Wizard

The Index Writer Wizard will assist in configuration of the name of the indexer, and the type of index to write. Once these are specified, the remainder of the parameters can be set in the Properties tab, or by finishing the wizard and working in the Properties window.

For parameter definitions, refer to the following sections on the types of indexes:

- “[IBM CMOD Index Writer](#)” on page 272
- “[XML Index Writer](#)” on page 278
- “[FileNet Index Writer](#)” on page 278

6.5.2 IBM CMOD Index Writer

The IBM CMOD Index Writer supports the creation of IBM Content Manager OnDemand index files.

IBM CMOD Index Writer Parameters



Note: The **IndexWriterType** parameter must be set to **ONDEMAND** for the Index Writer component to recognize these parameters.

The IBM CMOD Index Writer parameters are located under the **OnDemandOptions** parameter. The following parameters are available:

OutputFile	Indicates the file name to write the index file to.

StackedOutput	Specifies whether the index refers to a stacked output file or not. By default, it is set to False .
Codepage	Indicates the Codepage value to be recorded in the index file.
JavaCodepageName	Designates the Java name for the codepage used when converting Java strings into index file output bytes. For instance, for codepage 1255 the value is Cp1255.
IncludePathInFilename	Specifies whether or not the full path name should be provided in the FILENAME field(s) of the index file. By default, it is True and full path name information is included.

For details on how to configure an Output Transformation Engine project to create stacked output files for loading to IBM CMOD, see [“Creating Stacked Output for CMOD Loading” on page 273](#).

6.5.2.1 Creating Stacked Output for CMOD Loading

When loading stacked output to IBM Content Manager OnDemand (IBM CMOD), there are specific bytes that need to be added to the stacked file. The StackedOutputHandlerFactory is specific for loading files into IBM CMOD. This procedure details how to create stacked files that can be loaded into IBM CMOD.

1. Create a project with the following components as illustrated:

A parser (AFP Parser illustrated), component, Indexer component, Index Writer component, a generator (usually PDF or AFP, but may be any type of data that can be input to IBM CMOD; PDF is shown).

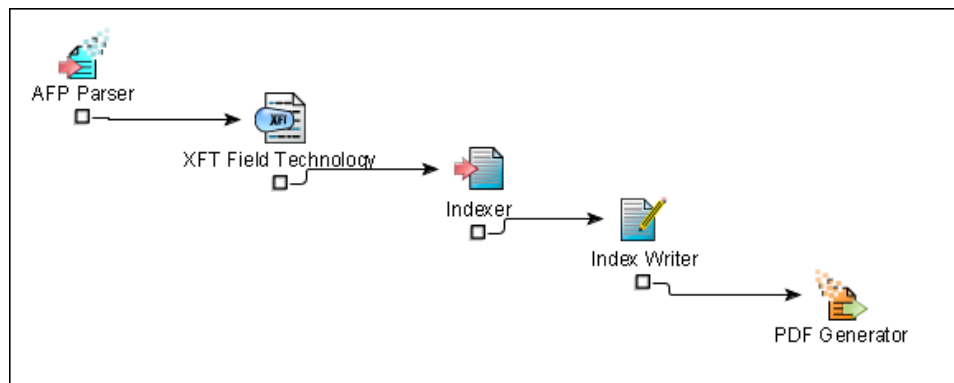


Figure 6-7: Example AFP to PDF project to create stacked output

2. Set up your indexing requirements and data mining criteria in the (XFT) component. This information and configuration is dependent on your unique project. (For more information about the XFT component, see [“XFT Field Technology” on page 341](#).)

3. Set up your document breaking criteria in the Indexer Component Properties.
4. Ensure document breaking and XFT indexing are both enabled on the Indexing tab.

Your Document Breaking tab settings are dependent on your unique project.

For more information about indexing, see [“Indexer Component”](#) on page 281.

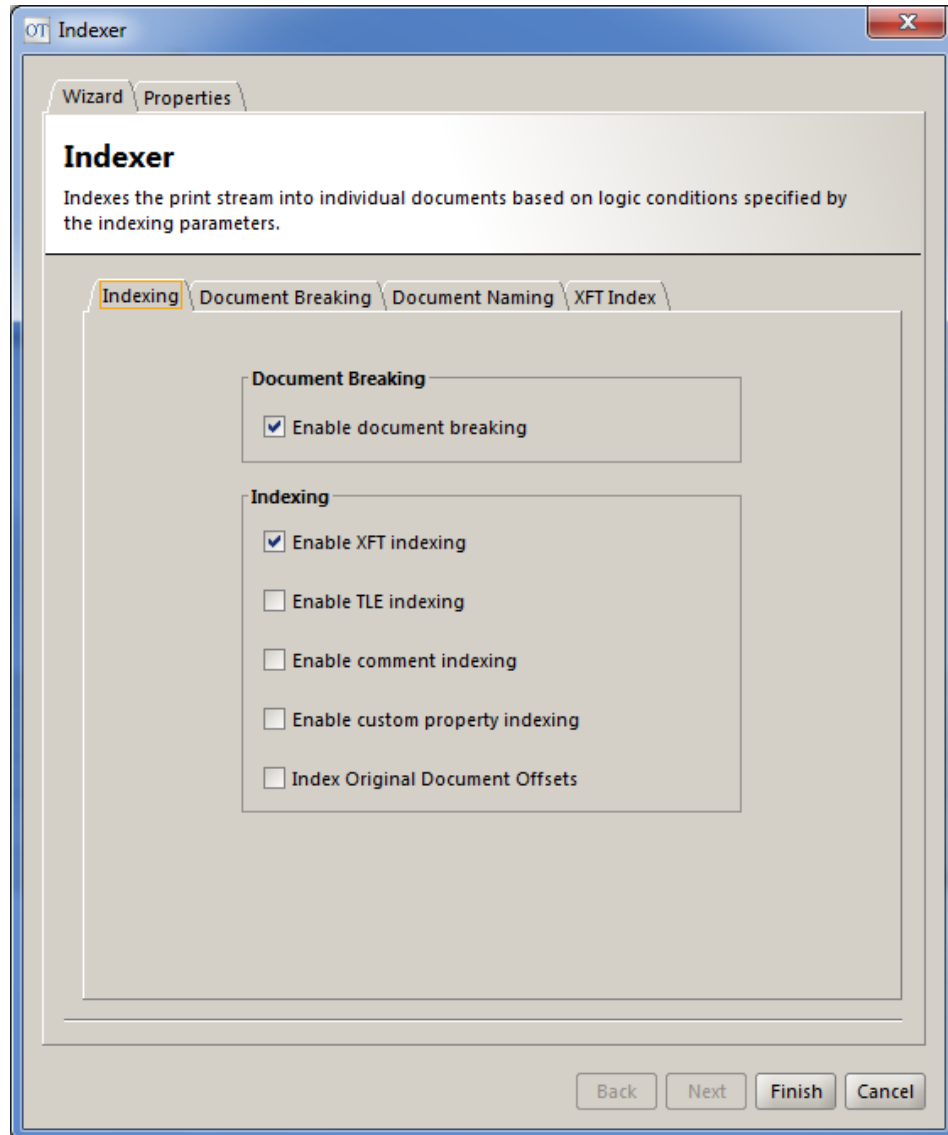


Figure 6-8: Doc breaking and indexing options

5. Test to ensure you are getting the results you expect.

6. Set up the Index Writer Component for proper OnDemand indexing (stacked output):
 - a. Choose ONDEMAND as the IndexWriterType.
 - b. Set **OnDemandOptions**. (For more information on these options, see [“Index Writer Component”](#) on page 272.

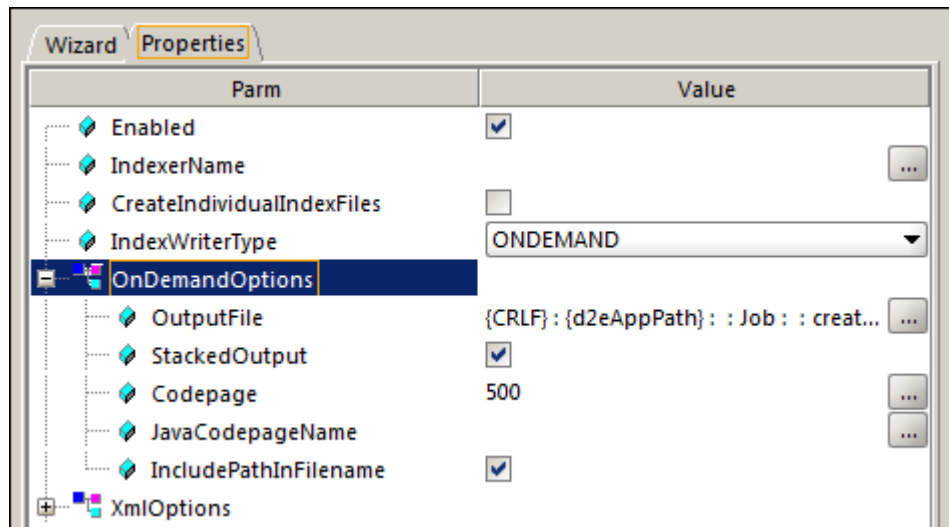


Figure 6-9: IndexWriter OnDemand options

This will create the GROUP_OFFSET and GROUP_LENGTH as required.

7. Configure the I/O Handler for the project.
 - a. In the Project pane on the left, right-click the **IO Handlers** folder and choose **Add**.
8. Enter the instance name STACKED.

In the Properties:

- Set the class name as `com.xenos.d2e.xdc.component.indexer.StackedOutputHandlerFactory`
- Add the following parameter: Name=addLengthField, Value=False

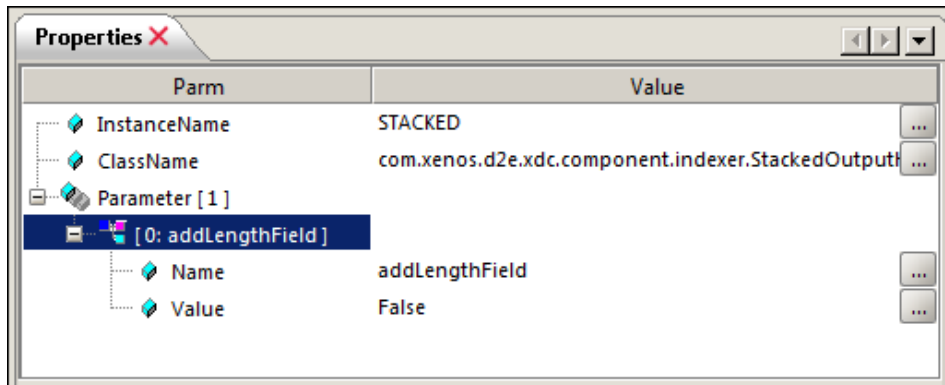


Figure 6-10: Stacked properties

9. Open the generator Properties and set the FdOutput parameter's I/O Handler to {Application}STACKED.

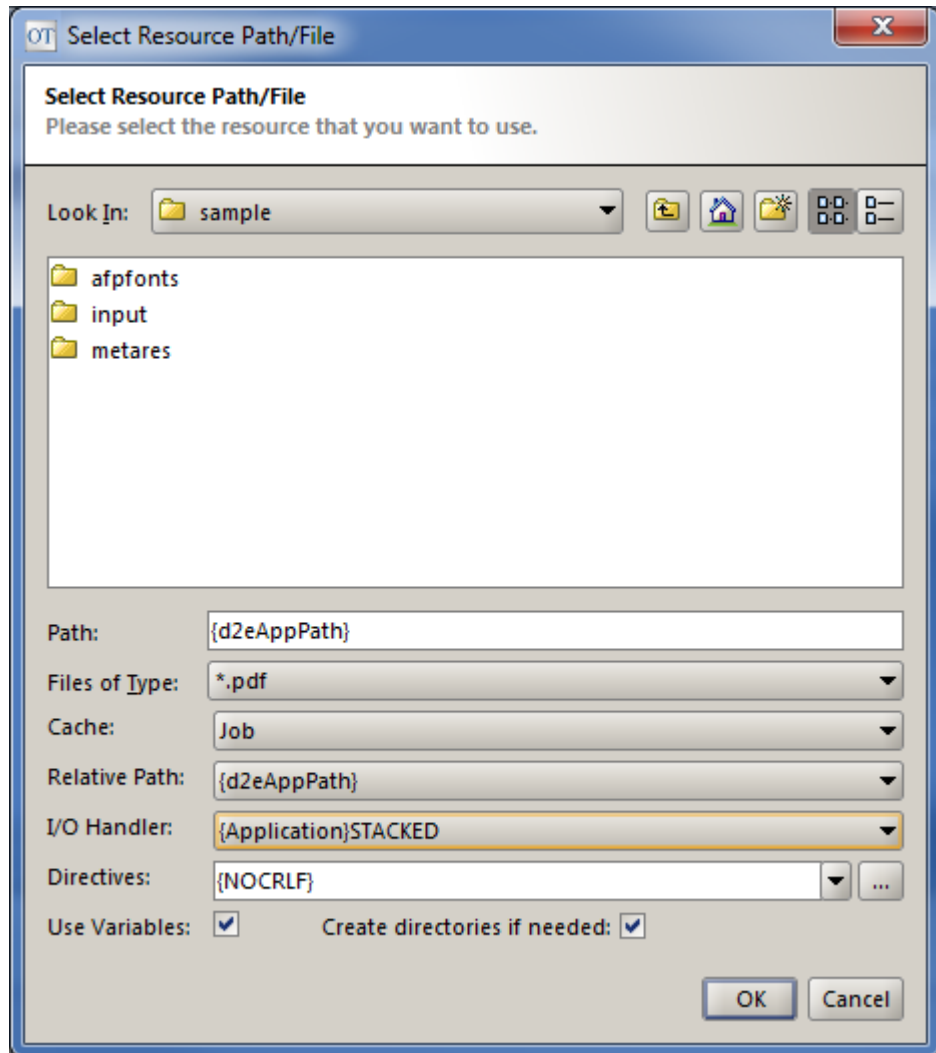


Figure 6-11: Setting the IO Handler

10. Re-run the project and check both the generated index file and stacked file to validate that it is correct. Open the output file in a text editor and validate it against the GROUP_OFFSET and GROUP_LENGTH in the index file to ensure that they match.



Note: When creating stacked output with an AFP Generator, you must build an external ACIF resource file instead of using inline resources since they would be replicated for each generated AFP document. Furthermore, instead of using a stacked I/O handler with the IncludeLengthFields option turned off for the AFP Generator, it is recommended you use the APPEND I/O directive on the generated AFP to simplify configuration.

6.5.3 XML Index Writer

The XML Index Writer supports the creation of indexes as XML files, using a fixed format.

XML Index Writer Parameters

The XML Index Writer parameters are located under the XmlOptions parameter. The following parameters are available:

IncludeFileInfo	Designates whether the XML includes information about the offset, length, and file name. By default, this is turned off.
IncludeEntryTags	Designates whether the XML includes <entry> tags to separate index fields coming from separate pages in the same document instance. By default, this is turned off.
OutputFile	Indicates the file name to write the index file to.
Encoding	Specifies the type of encoding to use. You can select from US-ASCII , ISO-8859-1 , or UTF-8 .



Note: The **IndexWriterType** parameter must be set to **XML** for the Index Writer component to recognize these parameters.

6.5.4 FileNet Index Writer

The FileNet Index Writer enables support for creating FileNet index files for FileNet archives. The FileNet Index Writer achieves this using the Output Transformation Engine Indexer Component.

FileNet index entry records can have any format as long as the records consist of comma-separated values and the field names are defined at the top if the index file before record entries begin.

Field Names:	"Field_name1", "Field_name2", "Field Name 3"
Record entry 1:	Field1_value, Field2_value, Field3_value
Record entry 2:	Field1_value, Field2_value, Field3_value
Record entry 3:	Field1_value, Field2_value, Field3_value

The field names and values can include other information such as: the output file name, the FileNet folder to which the document will be loaded, and the size.

The FileNet Index Writer uses a parameter with a format string as a value, to define the structure of the records. It will accept XFT and TLE names, job variables, and

index fields as parameters in the format string and perform substitutions creating the index record and writing it to the index file. Each indexer record line is terminated with carriage return/line-feed characters ('\r\n').

6.5.4.1 FileNet Index Writer Parameters



Note: The **IndexWriterType** parameter must be set to **FILENET** for the Index Writer component to recognize these parameters.

<i>OutputFile</i>	Specifies the filename to which the index file is written. This can be easily set by double-clicking the value field, which opens the Resource dialog window , and selecting Add/Edit , which opens the file name that can be selected.
IndexHeaders	Indicates a string consisting of headers for the columns that are to be written to the FileNet Index File. No variable substitution will take place. For example: "Policy Number", "Project Name", "Document Output".
IndexFormat	The IndexFormat string is made up of the fields that will be extracted from the data and inserted into the index. They can be made up of: • Strings • XFT fields • TLEs • Inserted data for indexing purposes, such as the output filename. Job variable substitution is supported.
WriteHeaderComment	Indicates whether or not to display the header comment. The default value is True .



Note: Document offset and document size index variables have not been implemented; they have been reserved for future releases.

Everything that is to be inserted into the index that has been extracted from the data **must** be enclosed in curly braces, { and }.

To include constant text, define it in your format string outside of the curly braces, for example:

```
{xftfield1},TLE-{tlename1},outputfile-{d2eOutput}
```

- **Document_Output_Key Variable.** There is a predefined index variable, DOCUMENT_OUTPUT_KEY. This variable is set by the generator to be the file to which the output is being written. You can use this in your index format string by inserting the following variable:

```
{job:DOCUMENT_OUTPUT_KEY}
```

6.5.4.2 Example FileNet Index File

The following is an example of an Index File written by the FileNet Index Writer with the FileNet Index Writer parameters set as in the figure below:

FileNetOptions	
OutputFile	{d2eOutputPath}filenet-index{0}.txt
StackedOutput	False
IndexHeaders	"Policy Number", "Project Name", "Document Output"
IndexFormat	"POLICY-{policy_number}", {d2eProjectName}, "{job:DOCUMENT_OUTPUT_KEY}"

Figure 6-12: Example FileNet options parameter values

// FileNet Index FileFormat
// This File Has Been Generated By The Xenos Process
// Started Mar 21, 2005 10:36:21 AM
"Policy Number", "Project Name", "Document Output"
"POLICY-01234 - 5512 - DE", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-1.pdf"
"POLICY-34567 - 7734 - GA", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-2.pdf"
"POLICY-45678 - 9956 - IN", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-3.pdf"
"POLICY-78901 - 2278 - KS", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-4.pdf"
"POLICY-67890 - 4489 - MS", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-5.pdf"
"POLICY-89012 - 0611 - OK", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-6.pdf"
"POLICY-12345 - 8813 - RI", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-7.pdf"
"POLICY-77777 - 1015 - TX", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-8.pdf"
"POLICY-90123 - 3317 - WY", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-9.pdf"
"POLICY-23456 - 1219 - AL", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-10.pdf"
"POLICY-09876 - 6721 - CT", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-11.pdf"
"POLICY-42234 - 1023 - UK", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-12.pdf"
"POLICY-76543 - 2324 - DL", sample-afp2filenet2pdf, "c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-13.pdf"


```
"POLICY-04551 - 8810 - BR",sample-afp2filenet2pdf,"c:\xenos\d2e-vision\Output\sample-afp2filenet2pdf-14.pdf"
```

6.5.5 Custom Index Writers

Customer-specific Index Writers can be self-written or by OpenText Professional Services. For more information, see these sections of the *OpenText Output Transformation Engine Developer Guide* explains:

- *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-PTE)*
- *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-PTE)*
- *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-PTE)*



Note: The Indexer component exposes an `IXifIndexFilter` that developers can use to modify the values that the Indexer receives. For more information on the `IXifIndexFilter` interface, see *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-PTE)*.

6.6 Indexer Component



The **Indexer** component includes the Indexer and several Index Writer components with the capability for other Index Writers to be written using the **API**. For more information on Index Writers, see *"Index Writer Component" on page 272*.

The Indexer provides the following functionality:

- Gathering information from the following sources:
 - XFT Fields
 - Tagged Logical Elements from AFP files
 - XIF Comments from AFP NOPs, Metacode Comment records, or TIFF tags if the **ASCIITagsAsComments** parameter is set
- Providing a filtering callback for every value added to the index

explains the following aspects of the Indexer:

- Integrating the Indexer
- Writing new generic Index Writers
- Developing Index Writers unique to your requirements



Note: The Indexer also exposes an `IXifIndexerFilter` that developers can use to modify the values that the Indexer receives.

Filters

User-written filters can be supplied to the Indexer, which may be used to update, validate, and add to the index information. This filter is defined in the Indexer's parameter class, and is loaded through the Java Reflection API. A set of key value pairs will also be available through the Indexer's parameters and will be passed to the filter on initialization. This enables the use of Index-specific configurations.

Document Breaking

If basic document breaking is required, the recommended method is to use the Document Breaking functionality supplied by the Indexer Component. However, if you require specialized document breaking logic, you can use the Document Breaking Component, but it must be placed before the Indexer in the workflow. The building of cross-document indexes can only be performed if the document breaking is achieved using the Indexer component.



Note: If you are using the Page Markup component to create accessible PDFs, any document breaking must be done using the Indexer component, and the Indexer component must precede the Page Markup component in the project flow.

Related Links

- [“XFT Indexing tab” on page 285](#)
- [“TLE Indexing tab” on page 286](#)
- [“Comment Index tab” on page 290](#)
- [“Document Breaking Component” on page 262](#)
- [“Page Markup Component” on page 295](#)

6.6.1 Indexer Wizard

The Indexing window has several tabs that enable you to easily set the indexing options for your project. Some will only appear when the different formats of indexing information are selected on the Indexing tab. The topics in this section describe each tab.



Note: For information on Indexing parameters, see [“Indexer parameters” on page 293](#).

6.6.1.1 Indexing tab

The Indexer component's Indexing tab appears by default when you open the Indexer wizard. If you require document breaking in your project, select the **Enable document breaking** check box. Selecting this option displays the **Document Breaking** and **Output Document Naming** tabs.

If you require indexing, select the type of information that you want to use for indexing. You can enable XFT Indexing, TLE Indexing, Comment Indexing, or Custom Property Indexing. Selecting any of these options displays their respective tab.

6.6.1.2 Document Breaking tab

The **Document Breaking** tab only appears after selecting **Enable input breaking** on the Indexing tab. The options on this tab allow you to set the mode of document breaking to use. There are several modes which can be set:

- **Page count.** A document break is triggered after the specified number of pages set in the **Number of Pages** field.
- **Page groups.** Denotes a list of regular expressions for the Named Page Groups to match. By default, document breaking occurs for every page group, however, if you want to perform document breaking for only specific page group names then you can add regular expressions matching the desired page group names to this list. If you want to match all page groups, leave the list empty.
- **XFT document change.** A document break is triggered when the XFT document changes.
- **XFT section change.** A document break is triggered when the XFT section changes.
- **XFT field change.** A document break is triggered when the XFT field changes value. You must identify the field that triggers the break.
- **XFT field equals.** A document break is triggered when the specified XFT field equals a certain value. You must identify both the field and the value that triggers the break.
- **Indexer match.** A document break is triggered when the indexer condition matches.
- **Indexer entry change.** A document break is triggered when indexer entries change.

Identifying document break trigger fields

To identify the field that triggers the document break:

1. Click on the **XFT field change** ellipsis button, and the **Field** dialog appears showing all available fields in the document.

2. Select the field name from the list, and click **OK**.

The name appears in the **Field name** box.

Identifying document break trigger fields and values

To identify the field and value that triggers the document break:

1. Click on the **XFT field equals** ellipsis button, and the **Field** dialog appears showing all available fields in the document.
2. Select the field name from the list, and click **OK**.
3. You are brought back to the Document Breaking tab, and the name appears in the **Field name** box.
4. In the **Value** field, type the value that the field must match in the order for the document break to occur.

6.6.1.3 Output Document Naming tab

The **Output Document Naming** tab only appears after selecting **Enable input breaking** on the Indexing tab. On this tab you set how the output file is named.

To generate a document file name, click on the variable type you want to use from the list, and click the **Insert** button to add the variable type to the **Document name** field. Then continue adding variables until you construct the desired output document name.

You can mix text with any of the following variables to produce a document file name:

- {**@**} – Indicates alphabetic characters (A-Z).
- {**#**} – Indicates numeric characters (0-9).
- {**PAGENO**} – The page number within the printstream.
- {**DOCNO**} – The sequential document number.
- {**MERGEDDOCNO**} – The sequential document number after document merging.
- {**SECTION**} – The name of the current XFT section.
- {**DOC**} – The name of the current XFT document.
- {**FLD:xxx**} – The value of the XFT field 'xxx'. If your project has XFT fields created by the XFT Component, then they will appear in this category.

Ensure that the document names generated are unique for each document. If this is blank, then {**DOCNO**} is used by default.

The following options also can be applied to document naming:

- **Include source and logical document breaks.** Specifies that the logical document breaks in the source file are preserved along with the document breaks added by your configuration of the Indexer component.
- **Sequential naming.** Determines whether the names of each document include an incrementing, sequential pattern. Two special characters can be used to create the sequential pattern: at symbol (@) and pound sign (#). The at symbol indicates the use of alphabetic characters (A-Z) and the pound sign indicates the use of numeric characters (0-9). You can chain together multiple variables to create more complex names. For example, a pattern of @@@### produces names like AAA000, AAA001 and through to AAA999, AAB000.

6.6.1.4 Document Merging tab

The **Document Merging** tab only appears after selecting **Enable output merging** on the Indexing tab. On this tab you set rules for how individual output documents are merged into a single output document.

The following merging options are available:

- **Merge all.** Denotes that all separate documents are merged into one document. Indexes are also kept and based on the documents prior to merging.
- **Page count.** Denotes that documents are merged according to the specified **Page count threshold**. Bear in mind that the page count threshold is a reference value for when documents are merged and that the actual page count of the output document may exceed the specified number.

6.6.1.5 XFT Indexing tab

The **XFT Index** tab only appears after selecting **Enable XFT indexing** on the Indexing tab. This tab allows you to control how XFT fields are used to gather indexing information:

- **Input field name.** The field name you want to include in the list of XFT fields to include in the index. Type the XFT field name, and click **Add**.



Note: XFT values must be entered manually, and they must match the existing XFT fields in your project.

Specify an asterisk (*) to process all fields, including all tables. Fields not named in this list will not be processed.



Note: The asterisk character (*) is only a wildcard to include all fields, and cannot be used for partial field names. For example, "Account*" will only match "Account*", and not "AccountNumber" or any other deviations.

- **Field names.** The list of XFT field names to include in the index.

Once added to the list of field names, you can edit the field name by selecting a name from the list and clicking the Edit button. Also, you can remove a field name by selecting the name from the list and clicking the Remove button.

Select the **Suppress duplicate entries** option to restrain duplicate consecutive index entries; only the first of a series of consecutive identical entries will be added to the index.



Note: Use the Table of Contents Component to create TOC entries based on this field. For more information, see “[Table of Contents Component](#)” on page 330 and “[Entry Properties Tab](#)” on page 333.

6.6.1.6 TLE Indexing tab

The **TLE Index** tab only appears after selecting **Enable TLE indexing** on the Indexing tab. The options laid out on this tab allow you to control how Tagged Logical Elements (TLE) are used to gather index information.

- **Input page group name.** The group name you want to include in the list of TLEs to include in the index. Type the group name, and click **Add**. Specify an asterisk (*) to process all page groups. Fields not named in this list will not be processed.
- **Page group names.** The list of page group names TLEs to include in the index.
- **TLE names.** Denotes the list of TLE names to include in the index.

Once added to the list of page group or TLE names, you can edit the page group or TLE name by selecting a name from the list and clicking the **Edit** button. Also, you can remove a page group or TLE name by selecting the name from the list and clicking the **Remove** button.

You can also opt to apply the following options to the Tagged Logical Elements:

- Select the **Suppress duplicate entries** option to restrain duplicate consecutive index entries; only the first of a series of consecutive identical entries will be added to the index.
- Select the **Include page-level TLE entries** option to allow the Indexer to search for TLE entries within the page, which ensures that all relevant TLEs are included in the input file offsets when indexing original AFP documents.

6.6.1.7 Comment Index tab

The **Comment Index** tab only appears after selecting **Enable comment indexing** on the Indexing tab. On this tab you can manage how comments are used to define index entries.

The **Comment Indexing** table displays the regular expressions and related indexing information used to evaluate the comments. If you want to evaluate all comments, then leave the list empty. When a regular expression is configured and it matches a comment, then the comment is processed. Comments that do not match the configured regular expression are ignored. Regular expressions can be inserted into the table by clicking **Add**. In the **Comment Entry** box that appears, the following options are available to analyze the comment entries:

- **Match expression.** Indicates the regular expression for comments to match. Any matching comments are placed in the index.

- **Index Record Name.** Contains options related to the source of the index entry names. You must select one of the following sources:
 - **Default.** The comment field name is generated by the Indexer component.
 - **Specified.** The comment field name is set to the value specified by **Entry name**.
 - **Substring.** The comment field name is a substring of the comment field value. You must specify the **Begin index** and **End index** values for the substring. When the end index is set to -1, the substring extends to the end of the comment string.
- **Index Value.** Contains options related to the source of the index entry values. You must select one of the following entry values:
 - **Entire record.** The comment field value is the original value from the input stream.
 - **Substring.** The comment field value is a substring of the comment field value. You must specify the **Begin index** and **End index** values for the substring. When the end index is set to -1, the substring extends to the end of the comment string.

Select the **Suppress Duplicate Entries** check box if you want only the first in a series of consecutive identical comment entries to be added to the index. By default, this is cleared.

As a sample use case, assume the following comments exist in your records:

```
"CUSTOMER_NUMBER 123456789"
"CLIENT_NAME John Smith"
```

To retrieve information from the first comment, a valid match expression could be:

```
CUSTOMER.*
```

To retrieve information from an entry name, you can use `cust_num` or begin and end index values of 0 and 15, respectively, to capture `CUSTOMER_NUMBER`.

To retrieve information from the index value, you can set up a substring retrieval of everything following `CUSTOMER_NUMBER`. For this, you would use begin and end index values of 16 and -1, respectively.

6.6.1.8 Custom Properties Index tab

The **Custom Properties** tab only appears after selecting **Enable custom property indexing** on the Indexing tab. This tab allows you to index the print stream into individual documents based on logic conditions specified by custom parameters.

To add a property for custom property indexing:

1. At Properties, click **Add**.
2. Enter a **Name** in the Property Definition Name field.
3. Tab to the **Value** field, or click inside the Value field.
4. Choose a defined XFT field from the bottom pane and click **Insert**. The XFT field name will be entered in the **Value** field.
5. Click **Apply** to add to the Properties.
6. Once you have added the properties you want, you can reorder them by selecting an item and clicking **Up** or **Down**, or you can remove the property by clicking **Remove**.
7. When complete, click **Finish**.
8. Enable **IndexCustomProperties** in the Indexer Properties.

6.6.1.9 Original Document Indexing tab

The **Original Document Indexing** tab only appears after selecting **Index Original Document** on the Indexing tab. On this tab you can specify options for how to index the original input document.

To establish an output file name, click on the variable type you want to use from the list, and click the **Insert** button to add the variable type to the **Document name** field. Then continue adding variables until you construct the desired output document name.

You can mix text with any of the following variables to establish the name used for the output file:

- **{@}** – Indicates alphabetic characters (A-Z).
- **{#}** – Indicates numeric characters (0-9).
- **{PAGENO}** – The page number within the printstream.
- **{DOCNO}** – The sequential document number after the input document breaking.
- **{MERGEDDOCNO}** – The sequential document number after document merging.
- **{SECTION}** – The name of the current XFT section.

- **{DOC}** – The name of the current XFT document.
- **{FLD:xxx}** – The value of the XFT field 'xxx'. If your project has XFT fields created by the XFT Component, then they will appear in this category.

You must stipulate how the input document is indexed, by **Page Numbers** or **Byte Offsets**. Page numbers are used for PDF documents while AFP and other non-PDF file types use byte offsets. When setting byte offsets, you can also denote whether to **Include all TLEs** for indexing AFP input. Typically, only TLE and NOPs from the start of the first page of a sub-document to the end of the last page of the sub-document are included. However, TLEs and NOPs can exist before the first page (between BNG and BPG) or after the last page (between EPG and ENG) of the sub-document. When all TLEs are included, the sub-document starts at BNG and ends at ENG.

6.6.1.10 XFT Doc Index Filter tab

The **XFT Doc Index Filter** tab only appears after selecting **Enable XFT Document filtering** on the Indexing tab. On this tab, you can configure filter conditions for XFT documents to filter or restrict index entries in an index file to only include entries that are associated with specifically-named XFT documents. Document indexing is based on matching XFT document names using literal prefix strings to match XFT document names by prefix or using regular expression strings to match the full XFT document name.

Be aware that filtering takes place when the end of a document is processed to remove unwanted entries. When the index filtering feature is used in conjunction with a custom index filter, XFT filtering occurs before any custom filters are executed.

For each index being filtered, the Condition entries are processed in the order in which they are defined. Additionally, if an include or exclude condition match is met, the remaining conditions will not be checked. Therefore, you must pay special attention to the order of your include/exclude rules since it can greatly affect the results.

To add a new index filter condition:

1. On the **XFT Doc Index Filter** tab, in the **Conditions** pane click **Add**.
2. In the **Skip Filter Entry Mode** list, select the mode of filtering to use. You can choose from the following options:
 - **Include**. Indicates that matching strings in the XFT document name are included in the results. Matches for include entries can be interpreted as Document1 AND Document2.
 - **Exclude**. Indicates that matching strings in the XFT document name are excluded in the results. Matches for exclude entries can be interpreted as Document1 OR Document2.
 - **InDoc**. Denotes that matches on XFT documents that contain a name that is not empty or null so that the XFT document is defined for the current

document index. Furthermore, to use this setting you must also enable the **DocBreaking > BreakOnFillerPages** parameter so that any non-document content, such as filler pages, is split off from the XFT document content.

- **Off.** Indicates that this entry is not enabled.



Note: BreakOnFillerPages is off by default for backwards compatibility. When the document breaking **Mode** is set to **documentchange**, BreakOnFillerPages adds a document break if existing XFT document ends without a new XFT document starting immediately. This functionality can be useful for breaking on blank filler pages between documents or for splitting off banner pages at the start of the document from the first actual page of the document.

3. In the **String Kind** list, select the type of string matching to use. You can choose from the following options:
 - **LiteralPrefix.** Denotes matches on the XFT document name.
 - **RegExToMatch.** Denotes matches on the entire XFT document name.
4. In the **String Value** box, type the string value for the condition to query against.
5. Click **Apply**.

The new condition entry is added to the **Conditions** list.
6. You can continue to add and remove conditions to set up your index filtering criteria. You must also make sure that the conditions are placed in an appropriate order since they are processed in the order in which they appear in the list.
7. When you are finished, click **Finish**.

6.6.1.11 Comment Index tab

The **Comment Index** tab only appears after selecting **Enable comment indexing** on the Indexing tab. The Comment Index tab enables you to set options to control how comments are used to gather index information.

Comment Indexing List. The Comment Indexing List is used to evaluate the comment contained in the input file. If you want all comments to be processed, leave the Comment Indexing list empty.

6.6.1.11.1 Adding and editing comment entries

To **add** or **edit** a comment entry:

1. To add an entry, click the **Add** button to the right of the **Comment Indexing** list.
To edit an entry, select the entry in the list and click the **Edit** button to the right of the **Comment Indexing** list.
The **Comment Entry** dialog window appears.
2. Complete the desired options in the three sections of the Comment Entry dialog window:

Criteria

The Criteria section contains one option, *<Match expression>*, which specifies how comment records should be identified in order to qualify as index records. There may be many comment records in the input document, but if only some records contain the desired indexing information you can use the match criteria to filter the records accordingly.

The value of the Match Expression should be a regular expression. Any standard regular expression may be used.

Examples of some common regular expressions are shown in the following table.

Expression	Description
. *	Matches any string. There are two parts (tokens) to this expression, the dot (.) character and the asterisk (*) character. The dot character is the regular expression wildcard, which means match any character. In regular expressions, the asterisk indicates the previous token may be repeated one or more times.
. {5,10}	Matches any string that is between 5 and 10 characters in length. Again, the dot character indicates that any character is a valid match. The second part, {5,10}, indicates the repeating factor of the previous token. It is in the format of {min,max}.
[0-9] *	Matches any string that only contains numbers. The hyphen character (-) is used to specify the range of valid characters.
[A-Z] *	Matches any string that only contains uppercase letters.
[a-zA-Z] *	Matches any string that only contains lowercase or uppercase letters.

<code>^[0-9]*</code>	Matches any string except ones that contain numbers. In this context, the caret, <code>^</code> , operator indicates the following item should be negated.
<code>abc def xyz</code>	Matches <code>abc</code> , <code>def</code> or <code>xyz</code> . The pipe, <code> </code> , character is an or operator, meaning a match is made if the token on the left side matches, or the token on the right side matches.
<code>abc?</code>	Matches <code>ab</code> or <code>abc</code> . The question mark makes the preceding token in the regular expression optional.



Tip: Regular expressions are commonly used in software applications and scripting languages and are extremely flexible. They can range from very basic expression to very complex expressions. If you are unfamiliar with regular expressions, there are many Web sites and free tutorials available on the Internet that provide in-depth detail on regular expressions.

Index record name

This section identifies how the generated index record should be named. The requirement to provide a specific name for comment records is common, since index records are essentially Name/Value pairs, and unlike other indexing record types, such as those based on XFT fields, they do not have their own specific names defined.

There are three choices to specify the index record name:

- *<Default.>* The comment field name is generated automatically by the indexer component. This is typically descriptive of the source of the comment record.
- *<Specified.>* This allows you to specify a desired string to use as the index name. Type the desired name into the Entry Name text field.
- *<Substring.>* This allows you to use a portion of the comment string as the name. You must provide the **Begin Index** and the **End Index** to identify which portion of the string you wish to use. The index is zero based; therefore to indicate the:
 - Start of the string, use a value of **0** for Begin Index.
 - Substring starts at the second character, use a value of **1** for Begin Index.
 - End of the substring is the end of the comment string, use a value of **-1** for End Index.

Index value

This section specifies what the index value should be. There are two choices:

- *<Entire Record.>* This option indicates that the entire comment record should be used as the index value.

- *<Substring.>* This allows you to use a portion of the comment string as the index value. You must provide the **Begin Index** and the **End Index** to identify which portion of the string you wish to use. The index is zero based; therefore to indicate the:
 - Start of the string, use a value of **0** for Begin Index.
 - Substring starts at the second character, use a value of **1** for Begin Index.
 - End of the substring is the end of the comment string, use a value of **-1** for End Index.

Click **OK** to save the new or edited entry.

6.6.2 Indexer parameters

The Indexer component indexes the printstream into individual documents based on logic conditions specified by the Indexing parameters.

A default set of parameters and values is used when you add the Indexer to your project.

To ensure that you have all the settings you need, we recommend you review the Indexing parameters in the Properties window before running your Output Transformation Engine project.

6.7 Large Print Component



The **Large Print** component reformats content from an input document into large print for your visually-impaired users. A high level of fine tuning is available for output documents through control of page layout, font color, size, or type, and the appearance of tables and lists, which can help you design outputs that support visually-impaired users in perceiving these page elements more capably. Alternate text can also be provided for elements to maximize accessibility for your users. All output content is reflowed and repaginated to boost readability.

When configuring the Large Print component in your flow, it must be placed downstream from the XFT Field Technology component with DOM configuration enabled, since the order of content and tag data is collected from these DOM settings, and ahead of a Generator. Any untagged content or content tagged as artifacts will be ignored for your output.



Note: For more information, see “XFT Field Technology Component” on page 340.

6.7.1 Getting Started with the Large Print Component

The Large Print component contains parameters that can tailor the appearance of images, tables, and lists to minimize their impact on readers with visual impairments as these page elements may interfere with a reader's ability to recognize the text content in your documents.

You can review the descriptions shown in the Properties window for each of the Large Print component's parameters to ensure you have all the settings needed for your project.

Figures

The parameters based around Figure control the processing of images. These options can manage the resolution, raster coloring, layout and also whether the images should be replaced by a string of alternate text.

Be aware of the following limitations:

- Images cannot have text elements overlaid on top of them.
- If the dimensions of the figure are bigger than the page size, then the image is automatically scaled down to fit on the page.

Tables

When working with tables within your documents, font settings for header and body text within tables can be individually customized. Also, table cell margins and rows with alternating pastel colors can be configured so that visually-impaired users can comprehend a table's layout and contents more easily in your accessible documents.

Using the specifications from the XFT Field Technology component, the table column widths ratio is preserved from your input document while also considering any ignored columns. If the column width data is not specified, then the column widths ratio is calculated based on the column content's bounding boxes. Furthermore, if a complete table row cannot be added to a page then the row can be split to span onto the next page or started on the following page.

Lists

All entries within a list processed by the Large Print component can be preceded with numeric or bullet label so that readers can easily identify the contents as a list. Spacing of list item indentations can also be customized as another method of aiding readability. Additionally, a list and sublist may be customized with separate configurations.

URL Links

The Large Print component cannot create hyperlinks for accessible output by itself, but it can be paired with the URL Link component to create functional hyperlinks

for users. For these situations, when setting up your project ensure that the URL Link component is downstream from the Large Print component so that the link tag can be properly inserted in the accessible output.



Note: For more information, see [“URL Link Component” on page 335](#).

6.8 Page Markup Component



The Page Markup component is an essential piece in building user accessible PDFs. User accessible PDFs are created in such a way that a screen reader software program can read the document in a logical way so it is easily understood by the listening user.

Accessible PDFs are built by an Output Transformation Engine project which includes any parser, an XFT Field Technology component, a Page Markup component and an Accessible PDF generator.

The **Page Markup component** determines the reading order of the document data defined in the XFT Field Technology component. It is also used to define for the screen reader how static or mined data is formatted for logical reading. This means that not only can you define that a customer's name and account number are read on a statement before the statement due date, but that the due date is read in a way that makes sense to the customer listening to the information.



Note: As of version 16.0, the Page Markup component has been deprecated and replaced with the DOM Properties within the XFT Field Technology component. Users who have designed projects using the Page Markup component and its accessible PDF features can continue to use them as the component is backwards compatible. However, going forward it is highly recommended that all projects with accessible PDF requirements utilize the XFT Field Technology component instead as all accessibility tags can be configured through the DOM properties.

For more information, see [“XFT Field Technology” on page 341](#).

6.8.1 Configuring the Page Markup Component

The Page Markup component allows you to configure which document fields are read by the screen reader and in what order they are read. Each section and field defined in the [“XFT Field Technology” on page 341](#) can be tagged and ordered in a logical way.



Note: The Page Markup component must follow and be linked from a fully configured component in a project flow. Additionally, if you are using the Document Breaking feature of the Indexer component, this must be completed before the Page Markup component in the project flow. For more information about working with the Indexer component, see [“Indexer Component” on page 281](#).

After adding the Page Markup component after the component in your project, double-click the Page Markup component to open the document in page-view. If this component is not linked from a configured component, you will receive an error.

When a new Page Markup component is created, there are no sections or fields defined. The summary pane contains no information, and the page view displays the document's first page with no markup information.

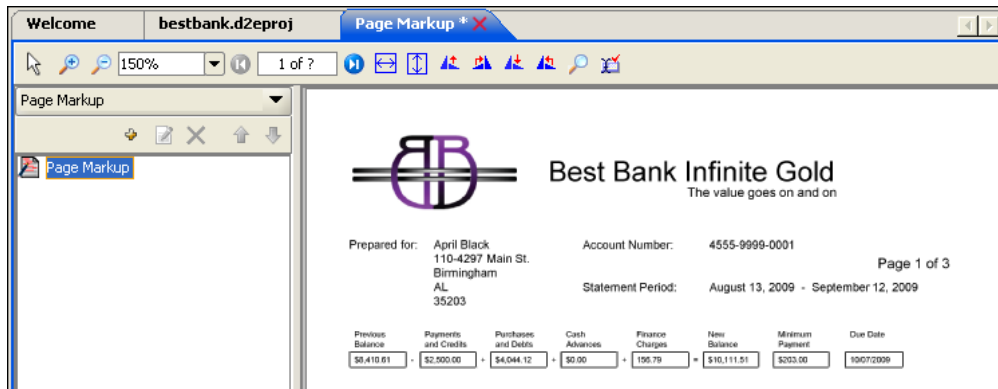


Figure 6-13: Unconfigured Page Markup page view

Defined XFT sections and fields must be imported from the component using the Synch Page Markup tool. The tool is also used to create the reading order of these fields for the Accessible PDF Generator. (For more information about working with the Accessible PDF Generator, see [“Accessible PDF Generator” on page 174](#).

Double-click **Page Markup** in the page view summary pane to access the **Synch Page Markup** dialog.

The defined XFT sections and fields are displayed in the left pane, while the right pane displays the items that have been imported into the Page Markup component. Expand the tree to see the embedded items. At this stage, there will be no imported XFT items in the right pane.

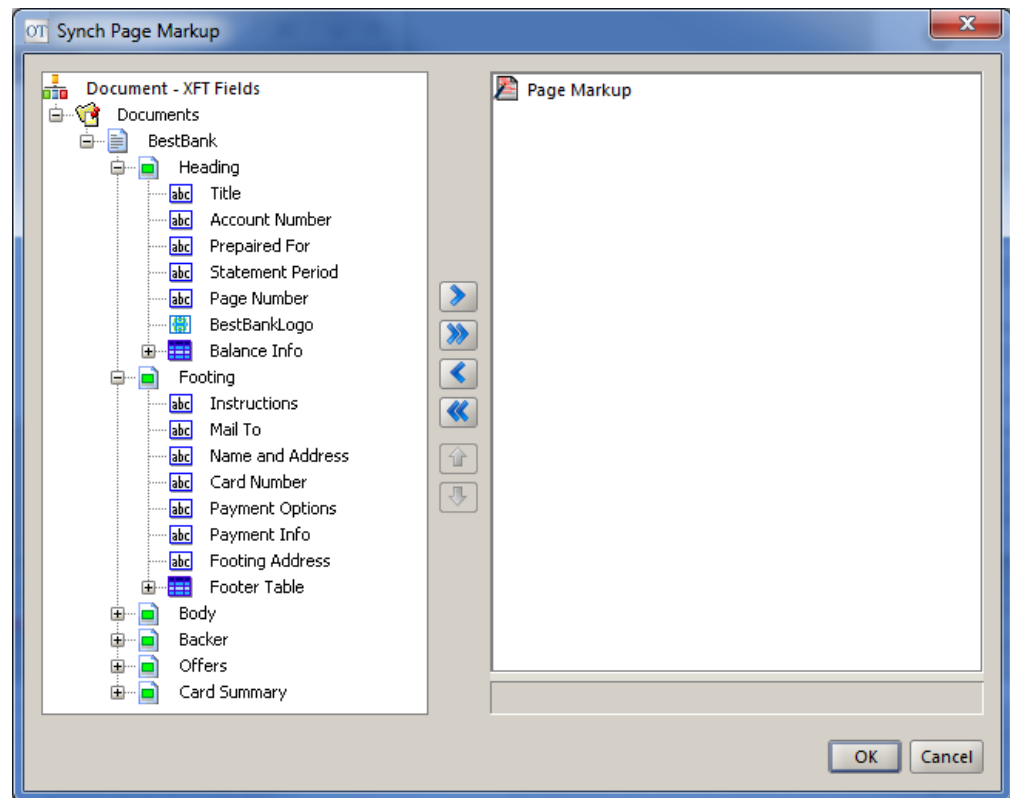


Figure 6-14: Unconfigured Synch Page Markup dialog

For a new page markup, import all the sections and fields at once by clicking **Add all**, . (If you do not want to add all XFT field definitions, you can add an individual field definition by selecting it in the left pane and clicking, **Add selected**, .)

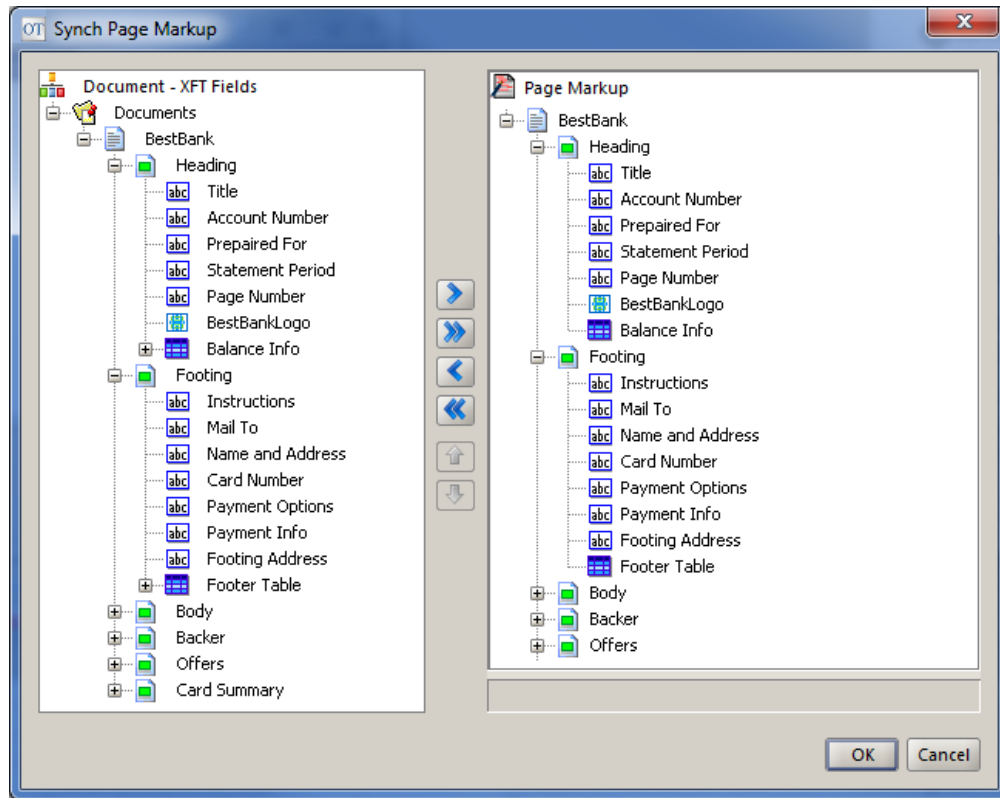


Figure 6-15: Configured Synch Page Markup dialog



Note: When using the **Add all** or **Remove all** options, it doesn't matter what item is selected in the list, as all items in the document will be added or removed from the Page Markup list.

Click a section to expand it and see the fields within. The page in the page view pane is marked up with the fields numbered in the order they are listed beneath the section title. This is the order in which they will be read.

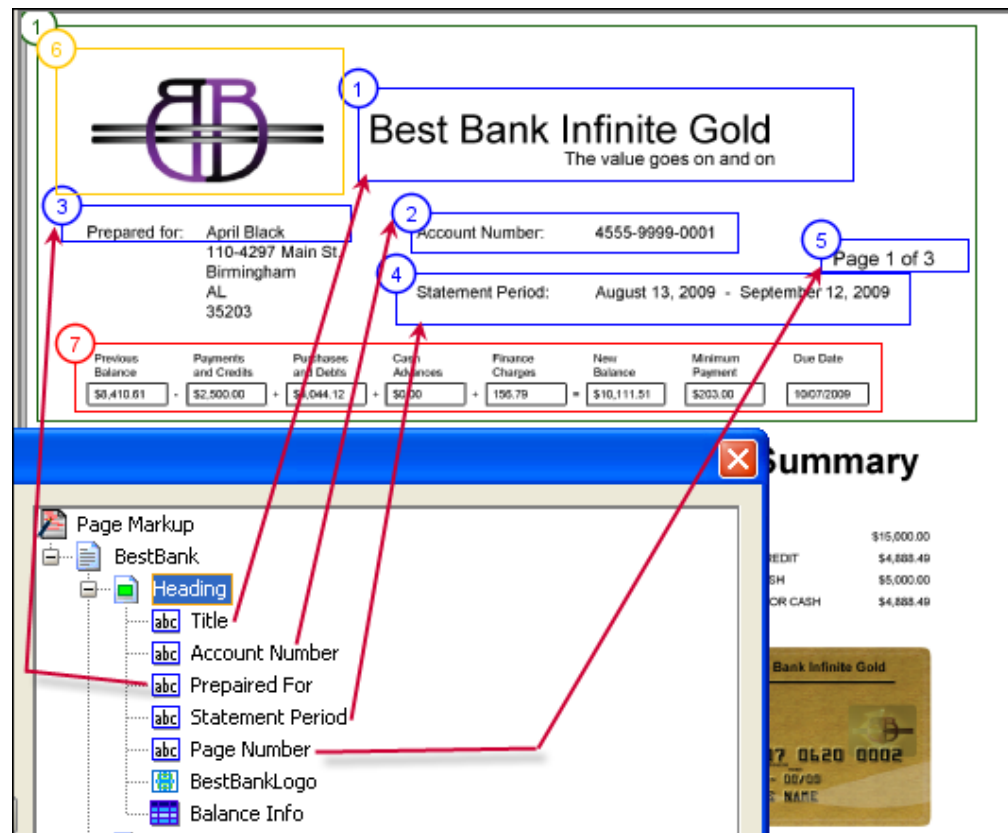


Figure 6-16: Reading order

To change the reading order, select the field in the list and click **Move up**, , or **Move down**, . When the reading order is changed, the new order is reflected in the page view pane.

To remove a field from the reading order, select the field and click **Remove selected**, . The field markup is removed so it is not read by the screen reader, however the element remains on the printed page.

To remove all fields from the markup, click **Remove all**, .

6.8.1.1 Tagging Elements as Artifacts

When transforming an input document to an accessible PDF document, all real content must be properly tagged. This includes content such as lines, borders, or images since they often exist in documents to visually distinguish a section or simply be decorative. These elements must also be tagged because when they are disclosed to accessibility users through the tag structure in the PDFs, they can be distracting and prevent users from being able to fully comprehend their documents. By explicitly tagging elements as an artifact, assistive technologies will ignore these elements, which help ensure accessibility users only receive content that is relevant to them. Basically, any content that does not provide meaningful information to a user requiring assistive technology such as decorative pictures, duplicate information (like headers or footers on every page), or unmarked content should be designated as artifacts prior to generating an output PDF to reduce confusion. The artifact tag is available for manual tagging in section, text field, object field, and table field properties. Any content that is not handled by the Page Markup component is automatically tagged as an artifact by your project's generator.

To help confirm that all content in the input documents is properly tagged, the Accessible PDF Generator's **ReportUntaggedContent** parameter can be enabled to identify and report all untagged content in the error log. You can then use the untagged content data from the error log to adjust your configuration settings in the Page Markup component to accommodate the missing tags. (For more information about this parameter, see [“Using the Accessible PDF Generator” on page 174.](#))



Note: When configuring the XIF option parameters, setting **XifDrawingOptions** to **Optimized** retains all elements tagged as artifacts on the output page. Furthermore, the drawing order for Optimized groups elements of the same type together in the output document structure but can be specified by the Page Markup component so tagging elements as artifacts can help to control the drawing order. (For more information about working with this parameter, see [“XifDrawingOptions” on page 128.](#))

Related Links

- [“Modifying Section Properties” on page 302](#)
- [“Page Markup Text Field Properties” on page 304](#)
- [“Page Markup Object Field Properties” on page 309](#)
- [“Page Markup Table Field Properties” on page 313](#)

6.8.1.2 Modifying Page Markup Properties

The following properties apply to the Page Markup component and need only be set once. Configure these settings in the Properties window of the project.

Documents	A collection of documents that are defined in the component. See Modifying Document Properties below.
TextProviderInstances	A collection of Alternate Text Provider instances. These are referenced by each MarkupItem's AlternateText Parameter by the instance name. • ClassName. The full name of a class which extends <code>com.xenos.d2e.xdc.component.pagemarkup.AlternateTextProvider</code> • Properties. Custom properties which if present are passed into the providers as arguments. These properties are persistent for the lifetime of this instance. #. Custom properties which if present are passed into the providers as arguments. These properties are persistent for the lifetime of this instance. Name. Indicates the custom properties file name. Value. Indicates the full path to the custom properties file. • Persistence. Defines the persistence of an instance of this class. SYSTEM means there is only a single instance of this provider for the lifetime of the engine. JOB means each job gets it's own instance. PAGE means that each XIF page has its own instance. NONE means that each call has its own instance. • InstanceName. The name of this instance of the provider. Used to distinguish between different instances of the same class.
LanguageOverride	The default language for the entire document as defined by RFC4646 Language subtag registry.

UseOriginalOrder	If True, the properties defined for this document are ignored and the order defined by the component is used. If selected, the tags are created, but no custom order, language overrides or alternate text is created. This is a quick way to design simple documents, but may not provide the best possible results.
UsePageTag	If True, a page tag is created for each page. This option is enabled by default.
ReportUID	If True, MarkupItems which have CalculateUID set to true will calculate and report the Unique Identifier (UID) for each XFT Image Object associated with a marked up object field. These UIDs are written to the report log in the format: UID for field {fieldName}=UID UIDs can be used by an alternate text provider to retrieve image information at runtime.

6.8.1.3 Modifying Document Properties

Each document has the following parameters:

Name	Indicates the name of the document. This value is automatically populated from the component.
Language	Specifies the language to use for this document as defined by ISO 639.
SectionOrder	Defines the order in which XFT sections should be processed. Each section contains a collection of Markup items which contain text, graphic objects, or tables.

6.8.1.4 Modifying Section Properties

Each section has the following parameters:

TagOnlyOnce	Select this check box to tag a repeating section only once on its first appearance, and then artifact any subsequent occurrences of the section. As a result, only the first instance of the selected section will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses which may repeat on multiple pages of billing statements. By default, this option is disabled.
TagType	Defines an appropriate tag type selection for this element.
StandardTagType	Designates the tag type for the element. Marking an item with any of the artifact tag types causes the item to be ignored by assistive technologies so that only meaningful content is exposed to accessibility users. You can choose from the following options: • SECT. Indicates the element is a section. • ARTIFACT. Indicates the section is a general type of artifact. If you mark a section as an artifact, all of its sub-elements are also considered artifacts. • HEADER_ARTIFACT. Indicates the element is a header-specific artifact. • FOOTER_ARTIFACT. Indicates the element is a footer-specific artifact. • WMARK_ARTIFACT. Indicates the element is a watermark-type artifact. • LAYOUT_ARTIFACT. Indicates the element is a layout-specific artifact. • BACKGR_ARTIFACT. Indicates the element is a background-type artifact. • PAGE_ARTIFACT. Indicates the element is a page-type artifact.
Name	Indicates the name of the XFT section.
MarkupItems	Displays a list of Markup items representing XFT text fields, object fields, and tables. These will be processed in the order that they are defined in this list.
Language	Indicates the language to use for the selected section as defined by ISO 639. This is useful for multi-language documents.

6.8.1.5 Modifying Field Properties

Each field has an associated icon, which indicates its field type:



. Indicates a text field.



. Indicates an object field.



. Indicates a table field.

Each field type has associated properties, which must be configured for the respective field types. See:

- “Page Markup Text Field Properties” on page 304
- “Page Markup Object Field Properties” on page 309
- “Page Markup Table Field Properties” on page 313

6.8.1.6 Page Markup Text Field Properties

If the XFT field type is Text, the following parameters can be configured. The basic parameters for text field types appear in the Synch Page Markup tool or at the bottom left side of the Page Markup tab, but the full set of parameters appear in the Properties tab.

TagOnlyOnce	Select this check box to tag a repeating element only once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. By default, this option is disabled.
TagType	Defines an appropriate tag type selection for this element.

StandardTagType	Designates the tag type for the element. Marking an item with any of the artifact tag types causes the item to be ignored by assistive technologies so that only meaningful content is exposed to accessibility users. You can choose from the following options: • P. Indicates the element is a paragraph. • H. Indicates the element is a general heading. • H1. Indicates the element is a level 1 heading. • H2. Indicates the element is a level 2 heading. • H3. Indicates the element is a level 3 heading. • H4. Indicates the element is a level 4 heading. • H5. Indicates the element is a level 5 heading. • H6. Indicates the element is a level 6 heading. • ARTIFACT. Indicates the element is a general type of artifact. • HEADER_ARTIFACT. Indicates the element is a header-specific artifact. • FOOTER_ARTIFACT. Indicates the element is a footer-specific artifact. • WMARK_ARTIFACT. Indicates the element is a watermark-type artifact. • LAYOUT_ARTIFACT. Indicates the element is a layout-specific artifact. • BACKGR_ARTIFACT. Indicates the element is a background-type artifact. • PAGE_ARTIFACT. Indicates the element is a page-type artifact.
Name	Indicates the name of the XFT Field that this Markup Item references. This is filled in automatically from the component. If you modify the name, the link between the field name and the field will be broken and the field will not be read by the screen reader. A broken link is identified in the list view like this:  Title x .

Language	Specifies the language to use for this item as defined by the RFC4646 Language tag registry. Do not enter a value if the language is the same as the main document language. If, however, a language needs to be set, any value specified here overrides the document for this item only.
DetectColumns	Determines whether columns are automatically detected within your documents. By detecting columns, a more logical reading order of column by column is established instead of a reading order of row by row. If true, the Accessible PDF Generator automatically scans the area between text elements for empty spaces that are wider than two (2) characters to determine what elements should be recognized as columns. If false, the generator does not attempt to identify the presence of columns. By default, this is set to True .
AlternateText	Defines the alternate text or formatting for this Item. This is used only if you want to substitute the existing text, or add special formatting instructions.
Type	Specifies the appropriate alternate text type or formatting to use from the drop-down list when substituting existing text or adding special formatting instructions. You can select from: • Static is set text that is substituted directly for the contents of this field. • Formatter applies specific formatting to the numeric field contents. • Provider associates the field with a classpath to an external source. Set the provider reference. • Disabled indicates that the alternate text is not used for this field. • CS_ALT_TEXT_MANAGER retrieves alternate text resources from a running Alternate Text Manager service. If you are setting this property on the Page Markup tab, you can click the ellipsis beside the Alternate Text field to set the alternate text, otherwise you must add them manually to the proper parameters in the Properties tab.

Static	Designates the static alternate text string to use. This parameter is only used if the Type is set to STATIC.
Formatter	Indicates the base format to use for this item. You can choose from the following options: • ReplaceCharacterParm. Replaces the specified character with another character. If no replacement character is specified, the input character is simply removed. • FormatDateParm. Formats all dates based on an input and output mask, which specifies how dates are read aloud by screen readers. For example, a due date on a statement may read, "11/08/09", indicating the 11th day of August, 2009, but the screen reader not knowing the semantics of the string, would read it as "eleven slash zero eight slash zero nine". By setting the formatter to FormatDate and setting an input mask and output mask, the screen reader can identify the string as a date field and read it in the appropriate manner. See InputMask and OutputMask for more details. • SpaceNumbersParm. Appends spaces in between all numbers. By adding spaces, text strings like account numbers are read more clearly by screen readers. Selecting this format causes all Locale settings to be ignored.
Locale	Contains locale settings to help structure the formatter parameter.
Language	Indicates the language code for this item as defined by ISO-639 standards.
Country	Indicates the country code for this item as defined by ISO-3166 standards.
IndexRange	Contains options for incorporating the index into the XFT results. If your results returns multiple lines, this index points to the lines containing the text to be formatted. Do not confuse this property with ApplyToCellRange, which is used for tables only.
Start	Designates the starting point of the index range. A negative number used as the value for the start results in a relative position from the end of the range while a zero (0) value represents the last item in the field.

End	Designates the ending point of the index range. A positive number represents an absolute position in the range while a negative number represents an offset from the last item. A zero (0) value represents the last item in the field.
InputMask	Designates the mask for dates in the item, which describes how the input date from the original data is read. This parameter only appears when the Formatter property is set to FormatDateParm. The following meta characters are available with an example appearing in the parentheses: • yy = 2 digit year (14) • yyyy = 4 digit year (2014) • M = Single digit month (1) • MM = 2 digit month (01) • MMM = Short form month name (Jan) • MMMM = Long form month name (January) • d = Single digit day of the month (9) • dd = 2 digit day of the month (09) • E = Short form day of the week (Wed) • EEEE = Long form day of the week (Wednesday) Thus, a sample mask for 09/01/2014 would be dd/MM/yyyy.
OutputMask	Designates the mask for dates in the item, which describes the format of the date to be returned from the formatter. This parameter only appears when Formatter is set to FormatDateParm.
CharacterToReplace	Specifies the character to be replaced or removed. This parameter only appears when Formatter is set to ReplaceCharacterParm.
ReplacementCharacter	Denotes the replacement character to use for the one specified in CharacterToReplace. If this value is left empty, the CharacterToReplace is removed and left blank. This parameter only appears when Formatter is set to ReplaceCharacterParm.

ProviderReference	Contains settings for specifying a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection. This is useful when reusing large chunks of static text across all documents in a large job, as it allows you to insert the section as an object, saving time and disk space in compiling the output PDFs, such as those used for customer billing. An object is more efficient to insert into a PDF, but then it is not readable by screen reader technologies. By using a TextProviderReference to instruct screen readers where the text content is located, a screen reader can read the section from the source text without impacting the structure and efficiency of the PDF. The parameters in this section are only active if the AlternateText Type is set to PROVIDER.
InstanceName	References the InstanceName of an AlternateTextProvider as defined in the TextProviders collection. On the Page Markup  you can select the ellipsis button,  , beside the Alternate Text field to bring up the Alternate Text dialog where you can configure the provider settings. For more information, see “Managing Alternate Text Providers” on page 321 .
Properties	Controls custom properties which, if present, are passed into the providers as arguments. The properties defined here are specific to the MarkupItem associated with this ProviderInstance reference.
HeadingInfo	Indicates that this text item is a heading with a level of one through six. By default, this parameter is set to NONE, indicating that the text item is not a heading.

6.8.1.7 Page Markup Object Field Properties

If the XFT field type is Object, the following parameters can be configured. The basic parameters for object field types appear in the Synch Page Markup tool or at the bottom left side of the Page Markup tab, but the full set of parameters appear in the Properties tab.

TagOnlyOnce	Select this check box to tag a repeating element only once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. By default, this option is disabled.
TagType	Defines an appropriate tag type selection for this element.
StandardTagType	Designates the tag type for the element. Marking an item with any of the artifact tag types causes the item to be ignored by assistive technologies so that only meaningful content is exposed to accessibility users. You can choose from the following options: • FIGURE. Indicates the element is a figure. • ARTIFACT. Indicates the element is a general type of artifact. • HEADER_ARTIFACT. Indicates the element is a header-specific artifact. • FOOTER_ARTIFACT. Indicates the element is a footer-specific artifact. • WMARK_ARTIFACT. Indicates the element is a watermark-type artifact. • LAYOUT_ARTIFACT. Indicates the element is a layout-specific artifact. • BACKGR_ARTIFACT. Indicates the element is a background-type artifact. • PAGE_ARTIFACT. Indicates the element is a page-type artifact.
Name	Indicates the name of the XFT Field that this Markup Item references. This is filled in automatically from the component. If you modify the name, the link between the field name and the field will be broken and the field will not be read by the screen reader. A broken link is identified in the list view like this:  Title x .

Language	Specifies the language to use for this item as defined by the RFC4646 Language tag registry. Do not enter a value if the language is the same as the main document language. If, however, a language needs to be set, any value specified here overrides the document for this item only.
URL	Associates a string with this object item as a URL, which users can use to go from the object directly to the URL. For example, if the object is a company logo, the object can be configured to launch the company website upon a user clicking on the logo.
AlternateText	Defines the alternate text or formatting for this Item. This is used only if you want to substitute the existing text, or add special formatting instructions.
Type	Specifies the appropriate alternate text type or formatting to use from the drop-down list when substituting existing text or adding special formatting instructions. You can select from: • Static is set text that is substituted directly for the contents of this field. • Formatter is only used for formatting numeric content in text or tables and does not work for objects. While you can select this option, it has no function on objects. • Provider associates the field with a classpath to an external source. • Disabled indicates that the alternate text is not used for this field. • CS_ALT_TEXT_MANAGER retrieves alternate text resources from a running Alternate Text Manager service.  Note: For more information, see “Page Markup Text Field Properties” on page 304 and “Tagging Elements as Artifacts” on page 300.
Static	Designates the static alternate text string to use. This parameter is only used if the Type is set to STATIC.

Formatter	Indicates the base format to use for this item. You can choose from the following options: • ReplaceCharacterParm. Replaces the specified character with another character. If no replacement character is specified, the input character is simply removed. • FormatDateParm. Formats all dates based on an input and output mask, which specifies how dates are read aloud by screen readers. For example, a due date on a statement may read, “11/08/09”, indicating the 11th day of August, 2009, but the screen reader not knowing the semantics of the string, would read it as “eleven slash zero eight slash zero nine”. By setting the formatter to FormatDate and setting an input mask and output mask, the screen reader can identify the string as a date field and read it in the appropriate manner. See InputMask and OutputMask for more details. • SpaceNumbersParm. Appends spaces in between all numbers. By adding spaces, text strings like account numbers are read more clearly by screen readers. Selecting this format causes all Locale settings to be ignored.
Locale	Contains locale settings to help structure the formatter parameter.
Language	Indicates the language code for this item as defined by ISO-639 standards.
Country	Indicates the country code for this item as defined by ISO-3166 standards.
IndexRange	Contains options for incorporating the index into the XFT results. If your results returns multiple lines, this index points to the lines containing the text to be formatted. Do not confuse this property with ApplyToCellRange, which is used for tables only.
Start	Designates the starting point of the index range. A negative number used as the value for the start results in a relative position from the end of the range while a zero (0) value represents the last item in the field.

End	Designates the ending point of the index range. A positive number represents an absolute position in the range while a negative number represents an offset from the last item. A zero (0) value represents the last item in the field.
ProviderReference	Contains settings for specifying a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection. This is useful when reusing large chunks of static text across all documents in a large job, as it allows you to insert the section as an object, saving time and disk space in compiling the output PDFs, such as those used for customer billing. An object is more efficient to insert into a PDF, but then it is not readable by screen reader technologies. By using a TextProviderReference to instruct screen readers where the text content is located, a screen reader can read the section from the source text without impacting the structure and efficiency of the PDF. The parameters in this section are only active if the AlternateText Type is set to PROVIDER.
InstanceName	References the InstanceName of an AlternateTextProvider as defined in the TextProviders collection. On the Page Markup tab, you can select the ellipsis button,  , beside the Alternate Text field to bring up the Alternate Text dialog where you can configure the provider settings. For more information, see “Managing Alternate Text Providers” on page 321 .
Properties	Controls custom properties which, if present, are passed into the providers as arguments. The properties defined here are specific to the MarkupItem associated with this ProviderInstance reference.

6.8.1.8 Page Markup Table Field Properties

If the XFT field type is Table, the following parameters can be configured. The basic parameters for table field types appear in the Synch Page Markup tool or at the bottom left side of the Page Markup tab, but the full set of parameters appear in the Properties tab.

TagOnlyOnce	Select this check box to tag a repeating element only once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. By default, this option is disabled.
TagType	Defines an appropriate tag type selection for this element.
StandardTagType	Designates the tag type for the element. Marking an item with any of the artifact tag types causes the item to be ignored by assistive technologies so that only meaningful content is detected for accessibility users. You can choose from the following options: • TABLE. Indicates the element is a table. • TOC. Indicates the element is a table of contents. • ARTIFACT. Indicates the element is a general type of artifact. • HEADER_ARTIFACT. Indicates the element is a header-specific artifact. • FOOTER_ARTIFACT. Indicates the element is a footer-specific artifact. • WMARK_ARTIFACT. Indicates the element is a watermark-type artifact. • LAYOUT_ARTIFACT. Indicates the element is a layout-specific artifact. • BACKGR_ARTIFACT. Indicates the element is a background-type artifact. • PAGE_ARTIFACT. Indicates the element is a page-type artifact.
Name	Indicates the name of the XFT Field that this Markup Item references. This is filled in automatically from the component. If you modify the name, the link between the field name and the field will be broken and the field will not be read by the screen reader. A broken link is identified in the list view like this:  Title x .

Language	Specifies the language to use for this item as defined by the RFC4646 Language tag registry. Do not enter a value if the language is the same as the main document language. If, however, a language needs to be set, any value specified here overrides the document for this item only.
AlternateCellTextList	Contains settings for arrays of alternate text definitions with each array applied to a specific range of rows and columns. If you are working on the Page Markup tab, you can click the ellipsis button beside this field to open the Alternate Text List window, which can be used to define the content of the table cells. To add  remove an item from the list, click the  buttons, respectively. Each item can have its own type to apply to one or more rows or columns. Click the ellipsis button in the Type column to set the type. See the Type parameter for more information on types.
Type	Specifies the appropriate alternate text type or formatting to use from the drop-down list when substituting existing text or adding special formatting instructions. You can select from: • Static is set text that is substituted directly for the contents of this field. • Formatter applies specific formatting to the numeric field contents. • Provider associates the field with a classpath to an external source. • Disabled indicates that the alternate text is not used for this field.  Note: For more information, see “Page Markup Text Field Properties” on page 304 and “Tagging Elements as Artifacts” on page 300.
Static	Designates the static alternate text string to use. This parameter is only used if the Type is set to STATIC.

Formatter	Indicates the base format to use for this item. You can choose from the following options: • ReplaceCharacterParm. Replaces the specified character with another character. If no replacement character is specified, the input character is simply removed. • FormatDateParm. Formats all dates based on an input and output mask, which specifies how dates are read aloud by screen readers. For example, a due date on a statement may read, “11/08/09”, indicating the 11th day of August, 2009, but the screen reader not knowing the semantics of the string, would read it as “eleven slash zero eight slash zero nine”. By setting the formatter to FormatDate and setting an input mask and output mask, the screen reader can identify the string as a date field and read it in the appropriate manner. See InputMask and OutputMask for more details. • SpaceNumbersParm. Appends spaces in between all numbers. By adding spaces, text strings like account numbers are read more clearly by screen readers. Selecting this format causes all Locale settings to be ignored.
Local	Contains locale settings to help structure the formatter parameter.
Language	Indicates the language code for this item as defined by ISO-639 standards.
Country	Indicates the country code for this item as defined by ISO-3166 standards.
IndexRange	Contains options for incorporating the index into the XFT results. If your results returns multiple lines, this index points to the lines containing the text to be formatted. Do not confuse this property with ApplyToCellRange, which is used for tables only.
Start	Designates the starting point of the index range. A negative number used as the value for the start results in a relative position from the end of the range while a zero (0) value represents the last item in the field.

End	Designates the ending point of the index range. A positive number represents an absolute position in the range while a negative number represents an offset from the last item. A zero (0) value represents the last item in the field.
ProviderReference	Contains settings for specifying a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection. This is useful when reusing large chunks of static text across all documents in a large job, as it allows you to insert the section as an object, saving time and disk space in compiling the output PDFs, such as those used for customer billing. An object is more efficient to insert into a PDF, but then it is not readable by screen reader technologies. By using a TextProviderReference to instruct screen readers where the text content is located, a screen reader can read the section from the source text without impacting the structure and efficiency of the PDF. The parameters in this section are only active if the AlternateText Type is set to PROVIDER.
InstanceName	References the InstanceName of an AlternateTextProvider as defined in the TextProviders collection. On the Page Markup  you can select the ellipsis button,  , beside the Alternate Text field to bring up the Alternate Text dialog where you can configure the provider settings. For more information, see “Managing Alternate Text Providers” on page 321 .
Properties	Controls custom properties which, if present, are passed into the providers as arguments. The properties defined here are specific to the MarkupItem associated with this ProviderInstance reference.
ApplyToCellRange	Controls alternate text that is applied only to cells defined within this range.
RowRange	Defines the Start and End points for the range of rows to apply the alternate text to.
ColumnRange	Defines the Start and End points for the range of columns to apply the alternate text to.

ColumnHeaderRow	Indicates the row in the table to use as a column header, where 1 is the first row in the table. If your table has no column headers, use 0 to disable this property. By default, the value is 0.
RowHeaderColumn	Indicates the column in the table to use as a row header, where 1 is the first column in the table. If your table has no row headers, use 0 to disable this property. By default, the value is 0.
TableSummary	Allows you to specify an optional brief description of the table.
TableOfContents	Contains settings to indicate whether the table should be tagged in an accessible PDF as a table of contents. For more information on setting a table as a table of contents, see “Tagging Table of Contents Structures Using Page Markup for use with Accessible PDFs” on page 318.
IsTableofContents	Designates whether the table represents a table of contents. By default, the value is false .

6.8.2 Tagging Table of Contents Structures Using Page Markup for use with Accessible PDFs

Accessible PDFs use XFT field definitions and tags set by the author to direct screen reader applications through a document in a logical manner. As per PDF/UA specifications, table of contents (TOC) structures are essentially defined as regular XFT tables but since TOCs carry more information to end users, any TOC structure must be explicitly marked with a Table of Contents tag so that screen readers can indicate this distinction in order to properly support navigation through these assistive technologies.

To tag a table of contents structure:

1. In Output Transformation Designer, set up a project that contains the parser for your input document's data type, the XFT Field Technology and Page Markup components, and the Accessible PDF Generator.

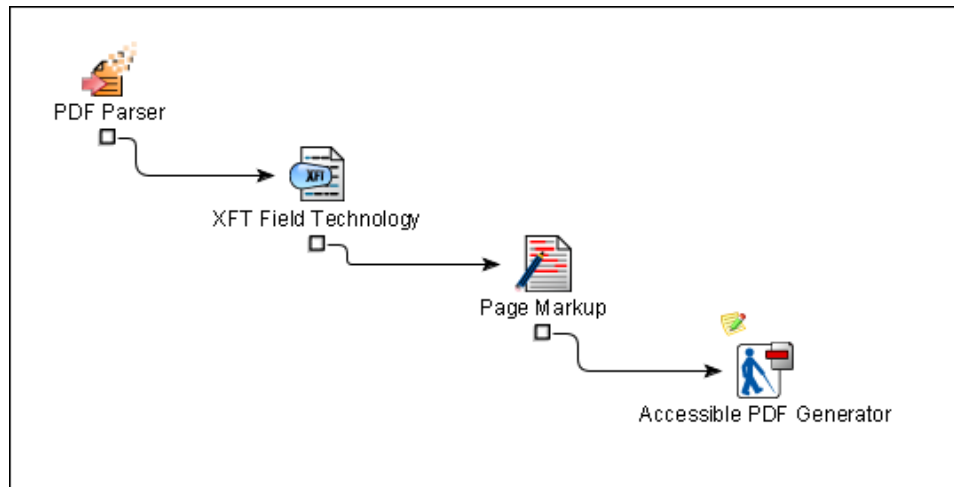


Figure 6-17: Sample PDF to Accessible PDF project

2. Open the component in the development window to view your input document in the XFT window and create the field definitions (i.e. XFT documents, text fields, object fields, etc.) for your document according to your document's requirements.



Note: You do not necessarily have to create all the field definitions for your document at this time, but at the bare minimum, you must define a Document as well as a Section that encompasses the table of contents before proceeding since these field elements are required for defining the table needed in the next step.

3. Locate your document's table of contents and click **Add Table** to define an XFT table with a single column around the table of contents. Once you are finished setting up the table, **save** your work and close the XFT window.

The development window reappears.

4. Double-click the **Page Markup** component in the development window to view the Page Markup window. Then, double-click Page Markup in the page view summary pane to bring up the **Synch Page Markup** dialog where you must import and set up the fields you defined using the XFT Field Technology component.

The left pane of the Synch Page Markup dialog displays the XFT fields currently defined by the component while the right pane presents the XFT fields that have been imported into the Page Markup component.

In the left pane of the Synch Page Markup dialog, locate and click the XFT document that contains your table of contents and import it to the Page Markup component by clicking **Add selected**, .

 **Tip:** You can add all defined XFT fields by selecting the **Add all**, , button.

Once added, the imported XFT items appear in the right pane.

5. In the right pane, expand the XFT item tree and select the XFT table element that contains your table of contents.

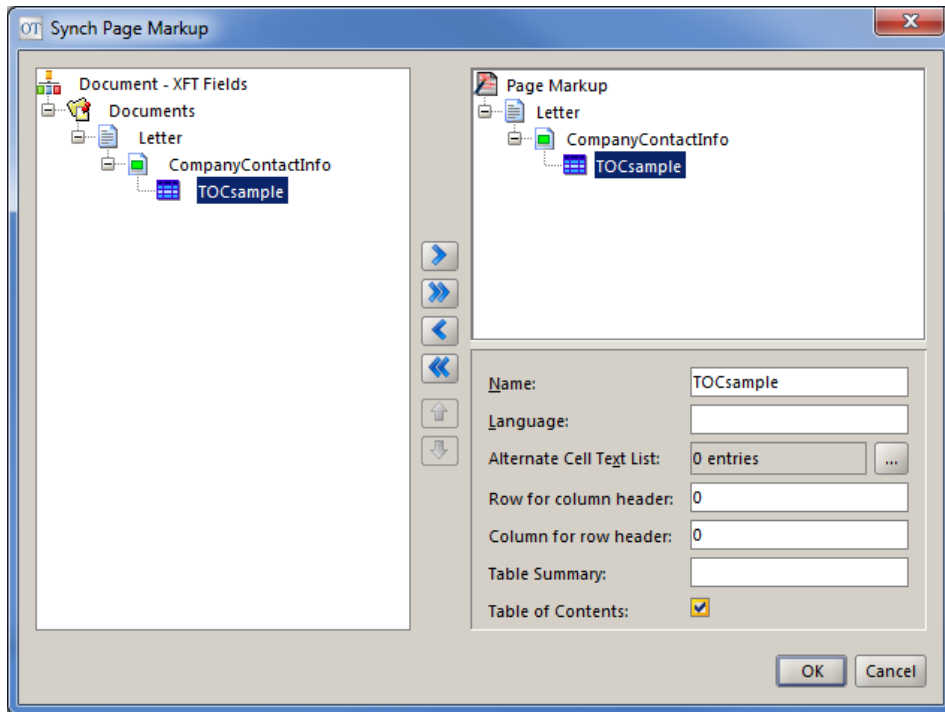


Figure 6-18: Synch Page Markup dialog

6. At the bottom of the right pane, select the **Table of Contents** check box to designate that the selected XFT table should be treated as a table of contents. Click **OK**.

The Page Markup window reappears.

7. **Save** your work in the Page Markup component.
The XFT table is tagged as a table of contents.

Related Links

- [“XFT Field Technology Component” on page 340](#)
- [“Page Markup Component” on page 295](#)
- [“Accessible PDF Generator” on page 174](#)

- “Getting Started with XFT” on page 356
- “Adding a Document” on page 370
- “Adding a Section” on page 375

6.8.3 Managing Alternate Text Providers

An AlternateTextProvider references an instance of text that can be reused across multiple documents. Alternate text providers are particularly useful when reusing large portions of static text that appear in all documents in a large job. The ProviderReference parameter is available for text, object, and table field property types; see the respective sections for more information.



Note: For more information, see “Page Markup Text Field Properties” on page 304, “Page Markup Object Field Properties” on page 309, and “Page Markup Table Field Properties” on page 313.

On the **Page Markup** tab, you can select the **ellipsis** button, , beside the **Alternate Text** field to bring up the **Alternate Text** dialog where you can configure the provider settings. You must have the **Type** set to **PROVIDER** in order to view these options on the Alternate Text dialog.

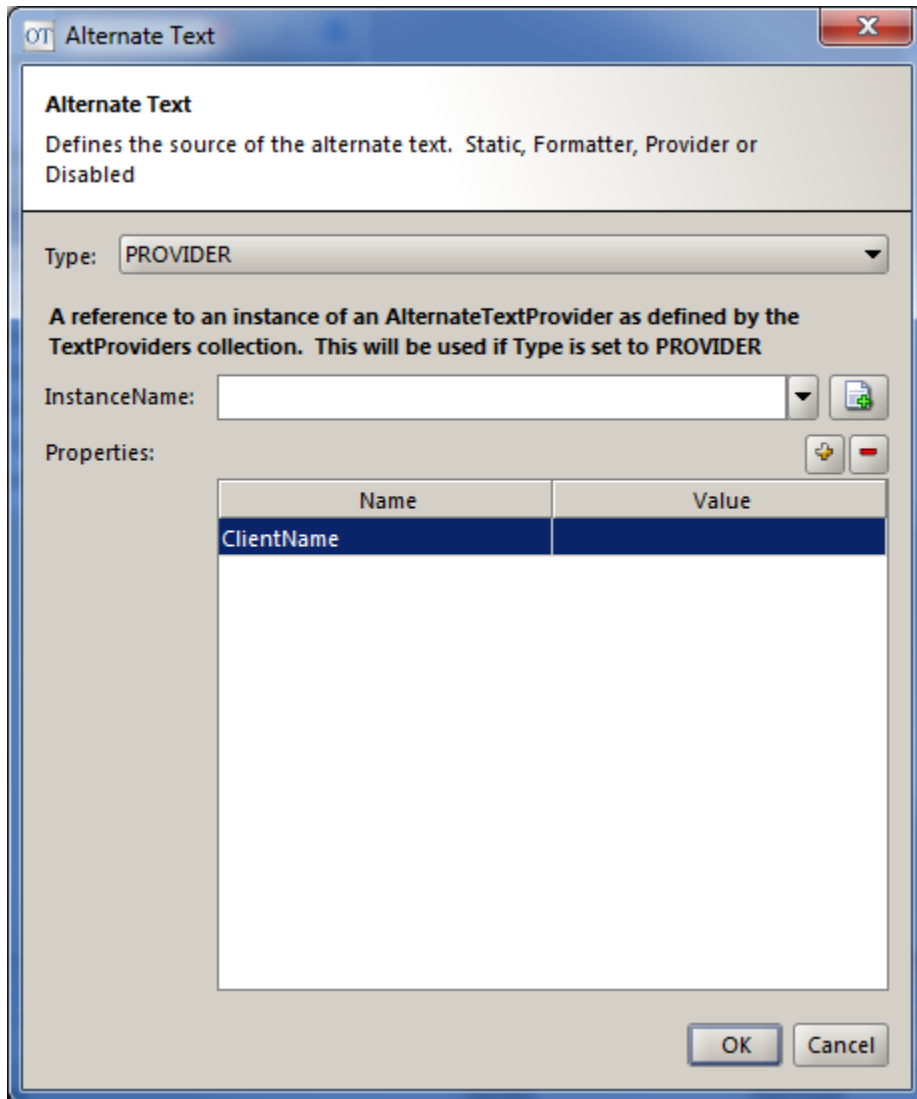


Figure 6-19: Alternate Text dialog

In the Alternate Text dialog, the following fields and options are available:

- **InstanceName.** Contains a list of all available provider instances in the dropdown menu.
- **Browse Providers,** . Opens the **Providers** dialog where all instances with their details are displayed. See the Providers dialog section for more information.
- **Properties.** Exposes the properties for the selected instance. Properties can be added or removed by clicking the **Add a new item,** , and **Remove selected item,** , buttons respectively

The **Providers** dialog allows you to see all instances with the associated properties and is also where you can create a new provider instance.

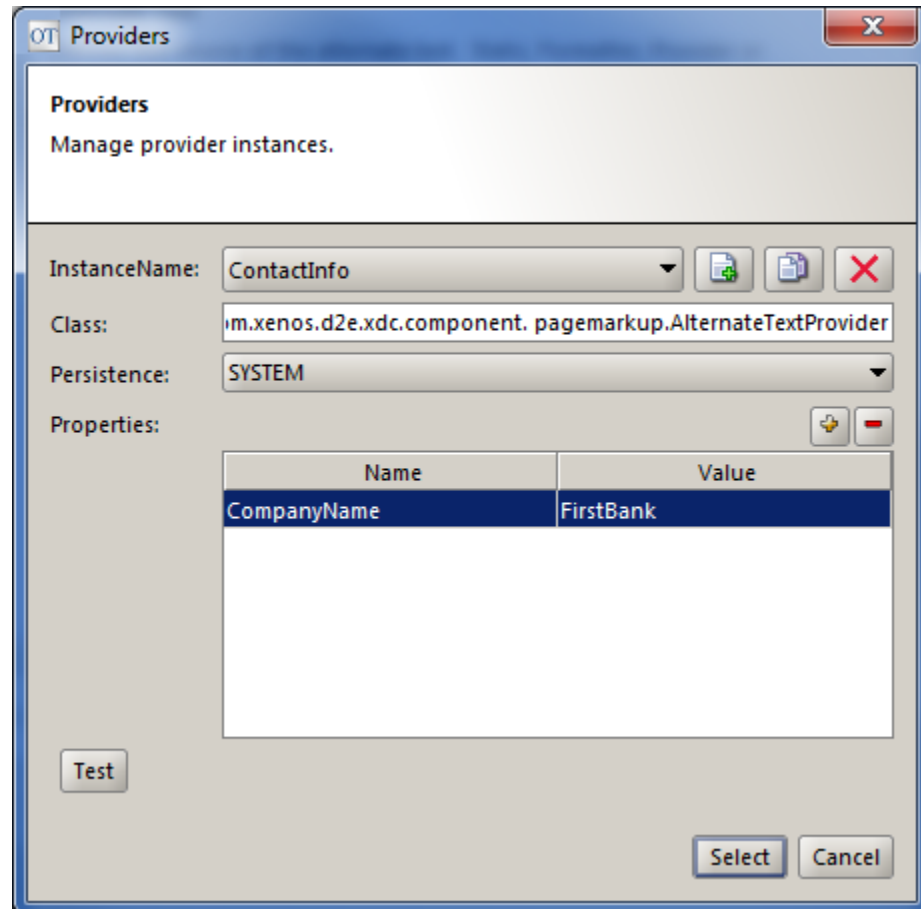


Figure 6-20: Providers dialog

In the Providers dialog, the following fields and options are available:

- **InstanceName.** Contains a list of all available provider instances in the dropdown menu.
- **Add new provider instance,** . Creates a new provider instance.
- **Copy selected provider instance,** . Clones the selected provider instance.
- **Delete selected Provider,** . Removes the selected provider instance.
- **Class.** Indicates the full name of a class which extends `com.xenos.d2e.xdc.component.pagemarkup.AlternateTextProvider` for use with the provider instance.
- **Persistence.** Defines the persistence of an instance of this class. **SYSTEM** indicates that there is only a single instance of this call for the lifetime of the

engine. **JOB** means each job gets its own instance and **PAGE** means that each XIF page gets its own instance. **NONE** indicates that each call gets its own instance.

- **Properties.** Controls custom properties which, if present, are passed into the providers as arguments. These properties are persistent for the lifetime of this instance.
- **Test.** Verifies whether the settings for your provider instance are valid and if it can locate the specified class.

6.9 Replace Text Component



The **Replace Text** component is used to replace or modify text that occurs in an XFT Field so that the replacement XFT Field text is shown in the output. This is particularly useful to mask confidential information such as Social Security numbers in a statement, or to correct addresses in a document.

The Replace Text Component parameters contain a list of those XFT Fields that have a field validator. When replacing an XFT Field value, you have the option of which field validator to use. One of the field validators built into Output Transformation Engine is the **ValidatorType** value of **RegExp**, which stands for Regular Expression. In addition to the ability to replace the XFT Field value with an alternative text value, the **RegExp** validator also provides the capability to use regular expressions, or plain text, to match XFT Field values to determine if an XFT Field should be replaced. You can also opt to create your own field validators by choosing the **ValidatorType** of **Custom**.



Note: The Replace Text Component does not support rasterization of text. Also, it is not supported for use with the TIFF or Image Generators.

6.9.1 Replace Text Wizard

The **Replace Text Wizard** simplifies the process of setting the component's parameters. You can also manually set the parameters on the Properties tab.

The Replace Text Wizard streamlines the following functions:

- Adding or Editing a Replace Text Field Condition
- Removing a Replace Text Field Condition

6.9.1.1 Adding or Editing a Replace Text Field Condition

To add or edit a replace text field condition:

1. In the Replace Text Wizard, do one of the following:
 - To add a new replace text field condition, click the **Add** button. The **Replace Text Field** dialog appears.
 - To edit an existing replace text field condition, select the row containing the settings you want to change and click **Edit**. The **Replace Text Field** dialog containing your current settings appears.
2. The options in the **Replace Text Field** dialog allow you to set your Replace Text options. There are several options available to be set:
 - **Field name.** Indicates the XFT text fields to modify.
 - **Validator.**
 - **Search text.** Specifies the text inside the XFT field to match the type of pattern or the text to be replaced. To match anything, you may enter the asterisk symbol (*) as a wildcard value for this parameter.
 - **Replace text.** Specifies the replacement for the text found by the SearchText parameter. Your ReplaceText value may contain regular expression meta characters such as \$1, \$2, and \$3.

 **Note:** If the replacement text is longer than the original text string, overlapping of text might occur.

 - **Validator type.** Indicates the type of validator to use. You can select from either **RegExp** or **Custom**. The RegExp validator matches and modifies XFT fields based on regular expressions, while the Custom validator allows you to write your own. When working with RegExp validators, you must also complete the other fields in the Validator group. When working with Custom validators, you must complete the fields in the Parameter group.
 - **Exact match.** Specifies, when **True**, whether the SearchText value must be an identical match for the value to be replaced by the ReplaceText. If **False**, every occurrence of the SearchText within the field value will be replaced by the ReplaceText parameter value. Non-exact matching is useful for replacing all numbers in a value with a constant character.
 - **Parameter.** Specifies the parameters when using the Custom validator.
 - **Class name.** Indicates the full package and class name of the Custom FieldValidator class. Your user-written FieldValidator class must implement the `com.xenos.d2e.xdc.component.replacetext.validators.IModifyFieldValidator` interface in the Java Reflection API.
 - **Name.** Specifies the name of the field.
 - **Value.** Specifies the value of the field to change.



Note: For more information on Custom validators, see “[Field Validators](#)” on page 326.

3. When you are finished, click **OK**.

The new replace text field parameters appear in the table on the Replace Text Wizard tab.

6.9.1.2 Removing a Replace Text Field Condition

To remove a replace text field condition:

1. In the table on the Replace Text Wizard tab, select the row you want to delete.
2. Click **Remove**.

The replace text field is removed.



Caution

There is no delete confirmation dialog or undo for this operation, so be extremely cautious when deleting replace text field conditions.

6.9.2 Field Validators

User-written Field Validators can be supplied to the Replace Text Component, which may be used to match, validate, and replace XFT field values. This validator is defined in the ReplaceText parameter class and is loaded through the Java Reflection API. A set of key value pairs will also be available through the ReplaceText parameter and will be passed to the filter on initialization. This enables the use of custom business logic for text replacement.



Note: For more information on how to write a field validator for your specific requirements, please see *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-PTE)*.

6.10 Script Component



The Script component provides users with a means of including scripts for interacting with key content in your projects such as document structure, page content, and extracted field data. API users can employ the component to add custom logic to their processes using JavaScript instead of writing, compiling, and managing custom Java-based components. At runtime, the component's script is executed once per page, following the same behavior as that of custom Java components.

The Script component is available in Output Transformation Engine's standard Palette and can be inserted into your project flows. It is a consumer type component so it must be positioned downstream from a parser. The Script component is also a producer type component and supports unlimited downstream consumers, so it is

suitable for branding and routing logic. Multiple instances of the Script component can be used within a project.

6.10.1 Script Component Wizard

The **Script Component Wizard** contains a JavaScript source editor where you can write code that interacts with the d2eJobState.

For more information on developing scripts with the Script component, see *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-PTE)*.

6.11 Simple Branch Component



The **Simple Branch** component distributes events to other components allowing you to create various types of output documents at the same time as it uses a single parser and multiple generators. The component's flexibility is useful for proof of concept or demonstration purposes.

When using the Simple Branch Component, connect the flow to the distributor then connect the distributor to the generators.

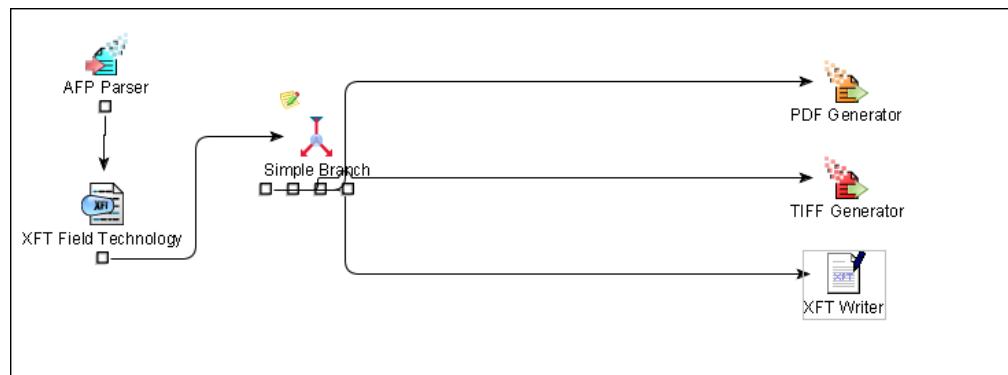


Figure 6-21: Example of a distributed flow

6.12 Skip Page Component



The **Skip Page** component removes pages from the input based on a variable number of conditions.

The Skip Page component is used to delete banner, blank, or redundant pages such as separators, headers, trailers, and other informational pages usually required when printing on paper.

These informational pages may occur:

- At the beginning of the stream (skip first page)

- At the end of the stream (skip last page)
- In between documents/statements (skip by field value)
- Every other page (skip even/odd pages)

The XFT Field Technology component is required if the **ByFieldValue** parameter is used. No other components are required for the Skip Page component to function properly.

6.12.1 Skip Page Wizard

The **Skip Page Wizard** assists in the customization of its parameters. You can also manually set the parameters on the Properties tab.

The following features are streamlined by the Skip Page Wizard:

- Adding or Editing a Skip Page Condition
- Removing a Skip Page Condition

6.12.1.1 Adding or Editing a Skip Page Condition

To add or edit a Skip Page condition:

1. In the **Skip Page Wizard**, do one of the following:
 - To add a new skip page condition, click the **Add** button. The **Skip Page Condition** dialog appears.
 - To edit an existing skip page condition, select the row containing the settings you want to change and click the **Edit** button. The **Skip Page Condition** dialog containing your current settings appears.
2. The options in the **Skip Page Condition** dialog allow you to set your Skip Page options. There are several options available to be set:
 - **When**. Specifies when to skip a page in the input. Valid values are:
 - **Blank**. Skips any blank pages, which are defined as pages with no elements on them.
 - **ByCommentValue**. Skips a page that contains the comment record with the specified value.
 - **ByFieldValue**. Skips any pages containing the given FieldValue definition.
 - **ByPageNumber**. Skips the page defined by the PageNumber property.
 - **Comment value**. Specifies that the page with the defined comment value will be skipped if the **When** parameter is set to ByCommentValue.
 - **Field name**. Specifies that pages with the defined field name in an XFT Field are skipped if the **When** parameter is set to ByFieldName. The XFT Field name is case sensitive and must match the one defined in the component.

- **Field value.** Specifies that pages with the defined field value are skipped if the **When** parameter is set to ByFieldName. This value is a case sensitive match.
 - **Page number.** Specifies the page to skip if the **When** parameter is set to ByPageNumber.
 - **Only constant form blank.** Indicates whether pages containing only ConstantForm objects should be considered to be blank. The parameter is set with a **True** or **False** value.
3. When you are finished, click **OK**.

The new skip page conditions appear in the table on the Skip Page Wizard tab.

6.12.1.2 Removing a Skip Page Condition

To remove a skip page condition:

1. In the table on the Skip Page Wizard tab, select the row you want to delete.
2. Click **Remove**.

The skip page condition is removed.



Caution

There is no delete confirmation dialog or undo for this operation, so be extremely cautious when deleting skip page conditions.

6.12.2 Skip Page Classes

The following lists the Skip Page Component classes:

com.xenos.psg.d2e.xdc.component.skippage	
SkipPage	The main component class. Skips one or more pages in the document using the given conditions.
FieldValuesMap	The utility class. Maps a unique field name "key" to multiple unique values: key -> (value1, value2, etc.)
com.xenos.psg.d2e.xdc.component.skippage.parm	
SkipPageParm	The main parameter class.

6.13 Table of Contents Component



The **Table of Contents** (TOC) component builds a Table of Contents tree for access by other components. The Table of Contents can be one that helps your readers easily locate sections of the document, or it can list names, account numbers, or other text to assist in pinpointing information within the document. This is useful in particularly large documents where you want to give readers the opportunity to use the bookmarks to jump to certain part of the text. In this way, they can minimize the switching forward or back that would otherwise be necessary.

If your output is PDF and you want to use the PDF Bookmarks feature, the TOC component is fully compatible. The TOC component of Output Transformation Engine helps you define and edit PDF bookmarks that correlate the text in your output to a specific location in a PDF output document to provide easier navigation through a document.



Note: Although the TOC component is used in the PDF Generator for bookmarking, it is not restricted to this generator, nor to only this type of usage. However, this component is not supported for use with the TIFF, AFP or Image Generators.

As an example of how the TOC component's functions could prove beneficial beyond making a straightforward listing of information from your input, consider an insurance company that might want to create three separate PDF files:

- PDF file with a TOC entry with the same information for each policyholder.
- PDF file with a TOC entry for each agent.
- PDF file for a branch manager with TOC entries showing all agents in the branch and each agent's policyholders.

By allowing multiple instances of table characteristic designs, the TOC Component allows for the creation of several personalized and catalogued outputs from a single set of input documents.

6.13.1 Table of Contents Wizard

The **Table of Contents** (TOC) **Wizard** contains buttons to assist in the creation of a TOC with the Table of Contents Wizard Controls. The wizard also has its own functions for placing PDF Bookmarks into your output PDF files; you can manually configure the component from the Properties tab, or there are two tabs in the TOC Wizard from which to configure the settings:

- Table Properties tab
- Entry Properties tab

6.13.1.1 Table of Contents Wizard Controls

The Table of Contents Wizard Controls appear throughout the Wizard tab. The menu and buttons manage actions within the TOC wizard.

The following controls are available:

- **Source XFT.** Indicates the XFT Field Technology component to be used to trigger TOC entries and from which to extract the TOC text. Select the source from the dropdown menu.
- **Add.** Adds a new TOC entry and enables the Entry Properties tab.
- **Remove.** Removes the selected TOC entry.
- **Apply.** Accepts the entries on the tab you are on.

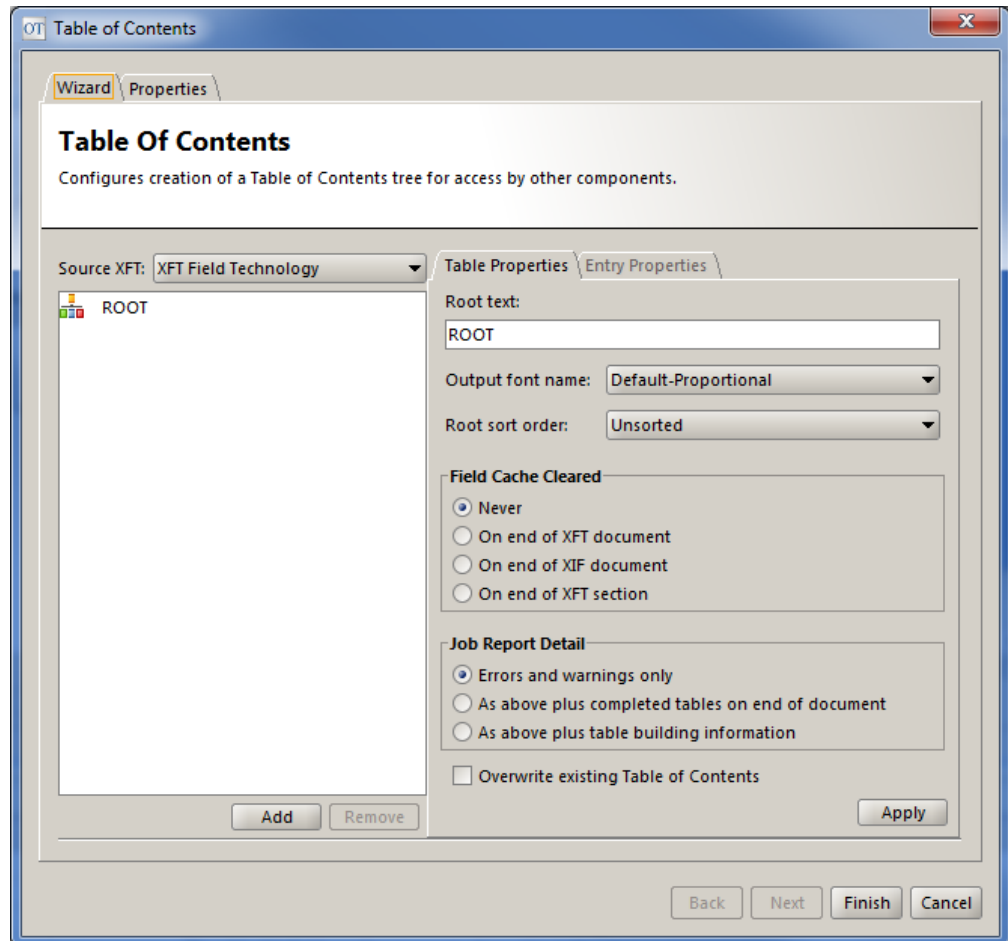


Figure 6-22: Table of Contents Wizard controls

6.13.1.2 Table Properties Tab

The Table Properties tab contains the specifications for the characteristics of the TOC table for your document. The following properties can be configured from the Table Properties tab:

- **Root text.** Indicates the text to appear at the top level, or root node, of the TOC list. For information on creating a TOC with multiple root nodes, please see [“Multiple Root Nodes” on page 334](#).
- **Output font name.** Specifies the font in which the text of the TOC appears. You can select the font name from the dropdown menu.
- **Root sort order.** Specifies the order in which nested bookmarks appear under their parent or root entry. You can select from the following options:
- **Unsorted.** Appears without being sorted.
- **Ascending.** Appears in ascending order (a-z).
- **Descending.** Appears in descending order (z-a).
- **Field Cache Cleared.** Specifies when the field cache is emptied. With the cache, you can allow for fields to be used which appear before the Table of Contents entry is added.

For example, you can use an account number retrieved from a previous section. The name remains pointing to the last value encountered, so the value can even be from a previous document, depending on your field cache settings.

You can choose from the following frequency options:

- **Never.** Designates that the field cache is not cleared under any condition.
- **On end of XFT document.** Designates that the field cache is cleared after the end of each XFT document.
- **On end of XIF document.** Designates that the field cache is cleared after the end of each XIF document.
- **On end of XFT section.** Designates that the field cache is cleared after the end of each XFT section.
- **Job Report Detail.** Determines the level of detail that the component adds to the job report. The following levels are available:
- **Errors and warnings only.**
- **As above plus completed tables on end of document.**
- **As above plus table building information.**
- **Overwrite existing Table of Contents.** Indicates that the existing Table of Contents is overwritten when checked.

Click the **Apply** button to accept your changes and the Table of Contents tree is added to the list on the left. If you want to clear all your changes, click the **Cancel** button.

6.13.1.3 Entry Properties Tab

The Entry Properties tab allows you to specify the properties of each TOC entry. The TOC entries are derived from fields defined in the **component**.

To activate this tab, select a TOC entry from the list or click the **Add** button.



Note: An component needs to be defined in the project before it can be selected from the Source XFT dropdown menu.

The following properties can be configured from the Entry Properties tab:

- **Text.** Specifies the actual text that appears in the Table of Contents for this particular entry. This text can contain constants and/or variables. Variables, contained inside { }, are defined by the XFT Field parameters. When the TOC is created, the variable is replaced with the text that is contained in the XFT Field on that page.

To select a field from the component, use the dropdown menu above the Text field. Select an XFT field and click **Insert**.

As an illustration of the Text field in use, suppose the TOC text is “Policy Number” followed by a variable field called “numbr” for the actual policy number. The entry in the TOC Text column of the TOC table is “Policy Number &numbr.” to indicate that the TOC is a text string followed by the variable, numbr. The ampersand, &, and the period, ., indicate that numbr is a variable.

- **Entry is unique.** Designates dynamic creation of TOC entries to ensure that entries at any level are not duplicated.

For instance, if pages 4 and 5 refer to the same report and family, but contain different account numbers, a new {Report Number} / {Family Name} TOC entry is not created on page 5. Structurally, the TOC entries would appear like:

```
{Report Number}
    {Family Name}
        {Account Number}
{Report Number}
    {Family Name}
        {Account Number}
```

However, with unique entries enabled, a new {Account Number} entry is created under the existing {Family Name} indent level:

```
{Report Number}
    {Family Name}
        {Account Number}
        {Account Number}
```

- **Child sort order.** Indicates the order in which nested bookmarks appear under their parent or root TOC entry. You can select from the following options:
 - **Unsorted.** Appears without being sorted.
 - **Ascending.** Appears in ascending order (a-z).
 - **Descending.** Appears in descending order (z-a).
- **Output font name.** Specifies the font in which the text of the TOC appears. You can select the font name from the dropdown menu.
- **Entry type.** Indicates the source of the TOC entry, either XFT or Index. Selecting one of these will change the remaining Trigger options:
- **XFT:**
 - **Trigger event.** Specifies an event to trigger the creation of a TOC. You must select one of the events from the dropdown menu.
 - **Event object list.** Contains a list of all the XFT documents, sections, and fields that the TOC is based on.
 - **Index:**
 - **Index entry name.** Specifies the index field name. For more information on the use of this field, see [“XFT Indexing tab” on page 285](#).
 - **Location.** Indicates settings to override the XFT field’s coordinate values. If the position, width, or height coordinates are set to -1, then the default XFT value will be used.
 - **X-position.** Specifies the X coordinates, in 72,000DPI.
 - **Y-position.** Specifies the Y coordinates, in 72,000DPI.
 - **Width.** Specifies the width, in 72,000DPI.
 - **Height.** Specifies the height, in 72,000DPI.
 - **Zoom.** Specifies the zoom factor.
 - **View.** Specifies the view for the bookmark. Select a view type from the dropdown menu.

6.13.2 Multiple Root Nodes

The Table of Contents component permits multiple instances of table characteristic designs, which allows for the creation of several personalized and catalogued outputs from a single set of input documents.

To create a TOC with multiple root nodes:

1. On the **Table Properties** tab of the Table of Contents wizard, create a root node entry, but leave the **Root Text** field empty.
2. Click **Add** to enable the Entry Properties tab.

3. Configure the settings for the TOC entries, and click **Apply** to create it. These will be child entries as they fall under the blank root node you created in step 1. You must create child entries for every node you wish to appear as root nodes in the TOC.
4. Run your project and in your output, the child nodes will be displayed in the root of the bookmarks when you open the document using Adobe Acrobat Reader.

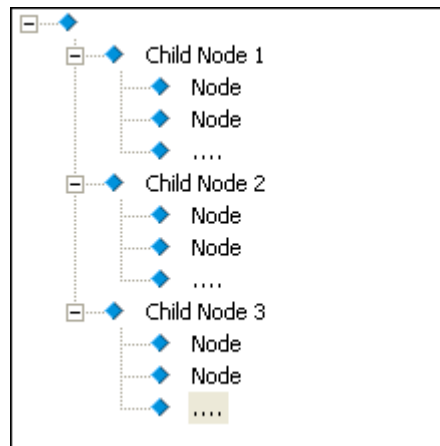


Figure 6-23: Example of multiple root nodes

6.14 URL Link Component



The **URL Link** component assists you in defining and placing web and internal links into your output documents.

With the URL Link component, you can define:

- Link trigger text.
- A Uniform Resource Locator (URL).
- A link's color and underline options.

The output file is created with the URL hyperlink inserted where the trigger text is found, and is attached to the text element. An option allows each individual trigger string to be linked each time it is found, or only the first time it is found. The link is built with an invisible PDF link box and the attributes of the text are set according to the link options selected.

6.14.1 URL Link Wizard

The **URL Link Wizard** simplifies the process of setting the component's parameters. You can also manually set the parameters on the **Properties** tab. The topics in this section describe how to use the various tabs of the **URL Link Wizard**.

6.14.1.1 URL XFT Field Options tab

If your project has more than one XFT Field Technology component, the options on the **XFT Field Options** tab allow you to select which XFT component (by name) the URL link uses.

The following options are available to be set on this tab:

- **Source XFT.** Indicates the XFT Field Technology component that the URL Link Tool is connected to and which XFT component to use when searching for trigger text and extracting link text. Select the component from the dropdown menu.
- **Enable XFT field value embedding.** Indicates that XFT field values can be embedded in URL links.

For instance, linking the phrase "OpenText" to <http://www.opentext.com> results in a plain link that does not have a field value embedded. However, if an account number field is needed to log users onto the website, like <http://www.opentext.com/logon={AcctNum}>, then this option needs to be enabled to allow for the field values.

- **Field cache flush event.** Specifies when the field cache is emptied, and therefore determines how long XFT field values remain stored. With the cache, you can allow for fields to be used which appear before the URL entry is added.

For example, you can use an account number retrieved from a previous section. The name remains pointing to the last value encountered, so the value can even be from a previous document, depending on your field cache flush settings.

You can choose from the following frequency options:

- **Never.** Designates that the field cache is not cleared under any condition.
- **On end of XFT section.** Designates that the field cache is cleared after the end of each XFT section.
- **On end of XFT document.** Designates that the field cache is cleared after the end of each XFT document.
- **On end of XIF document.** Designates that the field cache is cleared after the end of each XIF document.

6.14.1.2 URL Index Entry Options tab

The **Index Entry Options** tab contains only one option, which selects whether index entry name embedding is enabled:

- **Enable index entry name embedding.** Indicates that index entry names can be embedded in the URL links. These values will be taken from the index entry cache, which is a collection of all index values received by the URL Link component at runtime. If a valid match does not exist in the cache, a placeholder is inserted in its place.

6.14.1.3 URL Extraction Settings tab

The **Extraction Settings** tab contains options regarding the appearance of the extracted text in your document. The following options are available to be set on this tab:

- **Text print direction.** Specifies the direction in which the URL link text will be printed. From the dropdown menu, select the print direction from which you scan the text. If you want to scan in more than one print direction, multiple URL Link components are required.
- **Edit Threshold Setting.** Specifies threshold settings, which define how or if pieces of text are joined together to make words, phrases and paragraphs. For instance, if the characters in “Automobile” were put on a page one by one, it may appear to be an intact word to the naked eye, but you would need to make words from word fragments in order to find “Automobile” on the page.

Within the threshold settings, depending on your desired outcome, configure the settings on the **Word Join**, **Phrase Join**, or **Paragraph Join** tabs.

The **Word Join** tab controls how words are determined by the program. You can join two or more pieces of text that visually are the same word, but have different fonts for some of the characters, or are delimited by the input data stream controls (i.e. split across records). If two segments of text start or end within the threshold, the two segments are joined to make a single word.

The **Phrase Join** tab controls how phrases are established. You must specify the conditions for connecting two or more words into a phrase. If the two words start or end within the threshold, they are joined.

The **Paragraph Join** tab controls how paragraphs are detected. The paragraphs are created by spanning phrases across lines, which is needed to join text on separate base lines. If two lines start or end within the threshold, they are joined to make a paragraph.

The following options are available to be set on the **Word Join**, **Phrase Join**, or **Paragraph Join** tabs:

- **Name.** Indicates the name for the group of threshold options. These settings can be used throughout and shared by sections, fields, and tables.

- **Base line variance.** Determines the number of pixels of variance from the base line that text elements can have, but still be included on the same logical line. If there is a difference of 1 pixel or more, words on different base lines may not be joined together unless you set a value.

For example, the text “12345 1st Street” could be treated as 2 lines; “st” on the highest base line as one line, and “12345 1 Street” as the second line. By setting the base line variance, the text in this example would be rendered as text elements on a single line.

- **Enable.** Indicates that the settings on the particular tab should be used.
- **Options.** Determines how the inline absolute and percentage values will be used to determine if text elements are joined based on their distance apart.
- **Inline abs.** Specifies the maximum number of points that an element can be away from another element to still be considered joined.
- **Inline %.** Specifies the maximum percentage of space between two elements to still be considered joined. The value is calculated from the given percentage of the space character; therefore, 150% is calculated as 1.5 multiplied by the space character’s width in points.
- **Font growth.** Indicates whether text elements that grow in font size can be joined.
- **Font change.** Indicates whether text elements that have different fonts can be joined.
- **Size change.** Indicates whether text elements that have different font sizes can be joined.
- **One space.** Indicates whether all joined words will only have a single space between them. Otherwise, the joined words will have the number of spaces that was originally calculated between the two words.

6.14.1.4 URL Link General tab

The **General** tab contains options for creating Link tags in an active DOM.

- **Create DOM Link tags.** Designates the addition of Link tags in the active DOM tree.

When a link is applied to content that is tagged in the DOM, a child Link tag is added to wrap the content. New Link tags will inherit the alternate text description defined for the link entry. If there is no alternate text defined for the link, then the URL destination is set as the alternate text.

6.14.1.5 Adding or Editing URL links

The URL Links tab displays a table that summarizes the URL links you have defined for your project. The properties determine when a link is inserted on a page, its appearance, and the link destination. Keep in mind that field values need to have been processed on the same or previous page to have the correct value inserted in the link text.

To add or edit a URL link on the URL Link tab:

1. In the URL Link wizard on the **URL Links** tab, do one of the following:
 - To add a new URL link, click **Add**, .
 - To edit an existing URL link, select the row containing the properties you want to change from the table and click **Edit**, .
2. Configure the properties for your URL link in the **Link Properties** dialog. If you are editing properties, the dialog is pre-populated with your previous settings. The following options are available:
 - **Search text** and **Search fields**. Indicates whether text or XFT fields in your document are used to contain the link.
 - **Trigger text**. Specifies the text that contains the link. If your trigger text is written as a regular expression, you must indicate as such by selecting the **Regular expression** check box. Also, be aware that text can also be extracted using the settings found on the XFT Field Options and/or Extraction Settings tabs.
 - **XFT field name**. Specifies the XFT field that contains the link. You can select the XFT field from the drop-down list.
 - **URL link**. Indicates the address of the URL to link to. To enable this field, you must also select the **Provide custom URL** check box.
 - **Alternate description**. Specifies an alternate description to use for the link to meet accessibility requirements.
 - **Color (MFCT)**. Denotes the color of the link. You can select the output color from the Color (MFCT) list. If you do not define one, the default color is used.
 - **First occurrence only**. Specifies that only the first occurrence of the trigger text or XFT field on the page is turned into a URL link.
 - **Underline**. Designates if the URL link is underlined.
 - **Thickness**. Specifies the underline thickness in XIF points.
3. When you are finished making your selections, click **OK** and the URL Links tab reappears.

6.15 XFT Field Technology Component

With the **XFT Field Technology** component of Output Transformation Engine, you can define fields, then review, edit, and utilize the fields to make the most of the data in your printstream. XFT uses the XIF Intermediate Format, which contains all of the objects within a page parsed by a parser.

XFT is a flexible, document-oriented method of extracting any content from the document using rules through defined page sections and fields. Data can also be located and extracted from the input printstream for various purposes, such as document splitting.



Note: For more information, see *“Parsers” on page 131* and *“Document Breaking Component” on page 262*.

The text elements in XFT are normalized to Unicode, allowing support for double-byte character sets (DBCS) and Cyrillic or other non-ISO-Latin-1 character sets (also known as Acrobat’s Win ANSI encoding). Threshold rules control how these characters are joined into words, words into phrases, and phrases into paragraphs.

Defined fields are locations in your document containing specific pieces of information, like account numbers, names, or key phrases that are unique and can be used to differentiate one document or group of pages from another. For this reason, defined fields are commonly used as triggers for breaking input files into individual documents and for anchoring bookmarks and URL hyperlinks.

For example, you may define a field around a customer’s account number, and indicate that the PDF output should break when that field changes. This gives you a separate document for each account number, even if each document consists of different numbers of pages. Furthermore, by defining sections and locators, you can use fields that may appear at different locations in each document, such as account totals which move based on the number of entries. This type of field definition gives you options for true data extraction.

6.15.1 XFT Essentials

Before embarking on any work with the XFT Field Technology component, there are several concepts and terms that you should be familiar with in order to truly maximize this tool component’s potential within your projects.

6.15.1.1

**Field Technology**

XFT Field Technology (XFT) consists of two areas:

- **Field Definition.** Field definition is the process of defining particular fields in a document and determining how you want to extract the data from your printstream.
- **Field Extraction.** Field extraction is the actual process of mining and working with that data.

A file may contain one or more documents. Each document consists of one or more pages across which one or more sections are defined. A section has a start identifier and an end identifier. The start and end identifiers are used to determine whether a section is active in the current page of the document.

A section has one or more fields associated with it. A field can be either a text field, an object field, or a table field, all of which have thresholds that determine how text within the field is processed.

Data is mined from a field, which has a bounding box associated with it. The bounding box of a field can be defined in terms of fixed offsets within a page, or based on offsets from a locator. A locator is a search for a string pattern against all text on the page.

6.15.1.2 **XFT Documents**

When you include an XFT Field Technology component in your project flow, a document is created in the **XFT tree** by default. This document represents the entire printstream at this point.

If you have different types of documents in your printstream, such as correspondence letters, statements, or valuations, you can define additional, separate documents within XFT. Or, you can combine different types of information into one XFT document.

This defines the logical document, which may comprise a number of pages. If you intend to split the printstream into many documents, you only need to define the one logical document that is the model for the others. More complex splitting can be achieved by defining additional documents. For example, you may have a printstream that consists of a series of customers' statements of coverage, a cover letter, and a bill for premiums. The entire customer set of information can be considered as one XFT document.

Since each of these physical documents in the set has the same account number or policy number, unique to each customer, you could break the printstream when the Account Number field changes. This groups the pages in the documents together based on account number. It also accommodates different numbers of pages for each "document." Even if some customers only get the letter, or the letter and coverage statement but no bill, your output set for each customer contains the correct pages.

If, for example, you also want to create a summary statement for an agent or broker, containing consolidated information from a number of customer records in that same printstream, you could define a second “document” with different rules for processing. The sections, fields and locators you specify can then group the data from the output according to the trigger AgentName. This document set might contain only mined data, such as:

- Customer's name
- Customer's address
- Customer's account number
- Policy anniversary date
- Premium rate

As the engine processes the printstream, it breaks the data into individual customer document sets, and writes and appends to a file for each agent the data it extracts from specified fields, all in the same pass.

6.15.1.3 XFT Component Properties

The following parameters apply to the XFT Field Technology component and are shown in the Properties window when the XFT component is selected. To retrieve more information about a particular parameter, click on the parameter to display its description in the Parameter Definitions area. Most of these parameters can only be configured in the Properties window, however, the parameters related to field definitions like Thresholds and Locators, can also be configured within their respective areas of the main XFT window.

Furthermore, some of the parameters are discussed in detail in the subsequent sections.

Thresholds	A threshold defines how pieces of text are joined together to make words, phrases, and paragraphs. For more information, see “Adding a Threshold” on page 462 .
Locators	Locators are used for identifying the start and end points in a document. For more information, see “Adding a Text Locator” on page 486 .
AutoDetectOptions	Allows the configuration of an auto detect options set, which can be referenced by name.
Documents	For information about adding and configuring an XFT document, see “Adding a Document” on page 370 .
WholePageThreshold	Defines the threshold that will be used to create the XftPage result string, which the XFT locators can use to locate patterns.

CacheAllResults	By default, text, table, and section results are not added to the higher level result object in order to conserve memory. Select this check box to override this behavior and have all results cached in their higher level result object.
WholePageIncFormText	Allows the whole page field to include Form Text elements.
WholePageIncConstantFormText	Allows the whole page field to include Constant Form Text elements.
EventHandlers	Indicates the event handlers that will be called when events occur within the XFT processing cycle.
FontFilter	Contains the collection of font filters which are used within this application. For each font filter, specify the filter name and a font description.
ValidateTextFieldInsideSection	Determines whether the text fields defined in a section need to validate their bounding boxes inside the section. This setting is enabled by default.
ActiveObjectsMayUseTopLocator	When selected (default value), then when a section or table continues on a subsequent page, the top locator can be used for positioning. When disabled, the fixed bounding box is used for the top location.
EnableRepeatingSectionProcessing	When a section or table repeats on the page and its end cannot easily be determined, select this check box to use the start locator of the next section or table as the end locator and bottom positioner of the current section/table. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section/table and the start of the next section/table. This feature is disabled by default.

6.15.1.3.1 Running projects using XFT1 vs. XFT2

The XFT Field Technology component uses the proprietary XFT engine to establish the properties of your XFT document. There are two versions of the XFT engine and through the component's **Version** parameter, you can control which version to use. **XFT1** is the original engine and provides backwards compatibility for any existing projects that were created with it. **XFT2** provides improved support for multiple orientations and multiple field instances to handle complex DOM structures. XFT2 is the default setting and it is highly recommended you use the newer XFT version for any new projects you create.

6.15.1.3.2 Detecting and tagging special characters

Dot leaders and other special characters are commonly used in documents to help define the page layout or simply to improve readability. When using auto detect, these elements can erroneously be included as text, which denotes to a screen reader that they are elements to be read. In practice, these dot leaders should be tagged as Artifacts.

There are a few parameters that can automatically mark up dot leaders or other special characters in your documents, which can either be used alone or in conjunction with each other, depending on your requirements. Under the **AutoDetectionOption** parameters for the XFT component, the following parameters are available towards handling special characters:

- **TagUnderscores.** Indicates whether multiple underscores, such as underlines or dots, are tagged. By default, this setting is disabled and these elements are not tagged.
- **BreakOnSpecialText.** Designates whether paragraph joins are stopped when the user-provided string is detected between two lines of text. The defined text from the auto detected fields is then removed and the logical fields are merged. Be aware that this parameter may have an effect on how tables are recognized.
 - **Enabled.** Toggles this parameter on or off. You must enable this parameter.
 - **SeparatorList.** Provides the list of literal strings and/or regular expressions to treat as separators.

This parameter makes the content more straightforward by creating a tag structure that is more correct through the merging of logical fields.

When working with XFT Sections or Fields, some DOM Properties can help with tagging special characters:

- **TextFilter.** Defines the filter criteria for the DOM elements created for this item, resulting in all elements removed by the filter being marked as artifacts. Be aware that this parameter only applies after auto detect or XFT field processing is finished running. Accordingly, if any conflict exists with the auto detect parameters then the results are dependent upon this parameter.

- **Enabled.** Toggles this parameter on or off. You must enable this parameter.
- **FilterPatternList.** Provides the list of literal patterns and/or regular expressions that is used to identify the text to filter out.

6.15.1.4 Identifiers

Start identifiers and **End identifiers** denote for which pages in the file an XFT document is active. Identifiers are defined in terms of one or more locators, which search for a text pattern that you specify. You may identify only start identifiers in your file, as the program knows that a document has ended when it encounters a new start identifier. Also, in some instances it may be necessary to have more than one locator for a start or end identifier because a locator may not yield any matches on a particular page.

As a sample of a situation where multiple locators would be useful, consider the sample AFP data included with Output Transformation Designer. In the section of the document called 'Coverage on Contents', the end identifier could be a locator based on the text 'Coverage on Personal'. If this locator happened to be on the next page, then you might want to define another locator based on something fixed in the footer of the page. This secondary locator prevents the section 'Coverage on Contents' from spreading onto the next page. In this case, the end identifier has two locators to work with to help you control your data. Additionally, a Confidence Factor can be applied to the presence of the locators to further determine whether the section is active.

For more information on identifiers in your project, see [“Adding a Text Locator” on page 486](#) and [“Confidence Factor” on page 345](#).

6.15.1.5 Confidence Factor

Every identifier is assigned a confidence factor, which represents the degree of certainty that the occurrence of the word or phrase correctly identifies the start or end of the section.

As a sample of a use case, the start of a document might be identifiable by the occurrence of the words 'Policy Document', but this phrase might also occur on other pages within the document. The start of the document might also be identifiable by the phrase 'Your Policy Details' but, again, the phrase could occur on other pages. You could assign a confidence factor of '50' to 'Policy Document' and '50' to 'Your Policy Details'. Then, you would define at the document level that the required confidence level to correctly identify the start of the document as 100 (50 + 50). By assigning these values, if both the phrases are found then it is certain to be the start of the document. However, if only one phrase is found, then the total confidence factor of 50 is below the required level, so the start of the document has not been reached yet.

6.15.1.6 Field Definitions

In addition to data extraction, projects can include a Document Breaking component, PDF bookmarks (Table of Contents component), Indexes (Indexer component), or URLs (URL Link component). All of these features are implemented based on rules that specify the fields which exist within each section.

A text field can be used to extract a single word or number or an entire paragraph. The data contained in the text field is determined from a bounding box. The bounding box can be defined in terms of fixed offset from the page, or offsets from one or more locators. It is possible to define the bounding box of a field in terms of offsets from many locators, to allow for the event that a locator does not always exist on a page.

The data returned from a text field is structured into words and paragraphs. The method used to structure the text extracted from a field is defined by thresholds. Thresholds group single letters into words, words into lines, and lines into paragraphs. You can also define rules to validate the data in a field.

Types of Fields

Output Transformation Designer supports **text fields**, **object fields** and **table fields**. Since the location of the field may not be at exactly the same coordinates on each physical page, defining fields within sections ensures that the field moves with the data.

Text fields contain any text string. This can include alphanumeric characters, as well as special characters.

Object fields contain objects, such as images, vectors, shades, or barcodes.

To accommodate data in tables, **table fields** incorporate rules that determine how table rows and columns break across pages. Table rows are further validated to ensure that the relationships between the columns are valid, as well as the column data values. For example, in a bank statement there are debit and credit columns, but there are no entries in both columns in the same row.



Note: You will not be able to use any Output Transformation Designer features that require trigger text or variable data until you have defined your fields.

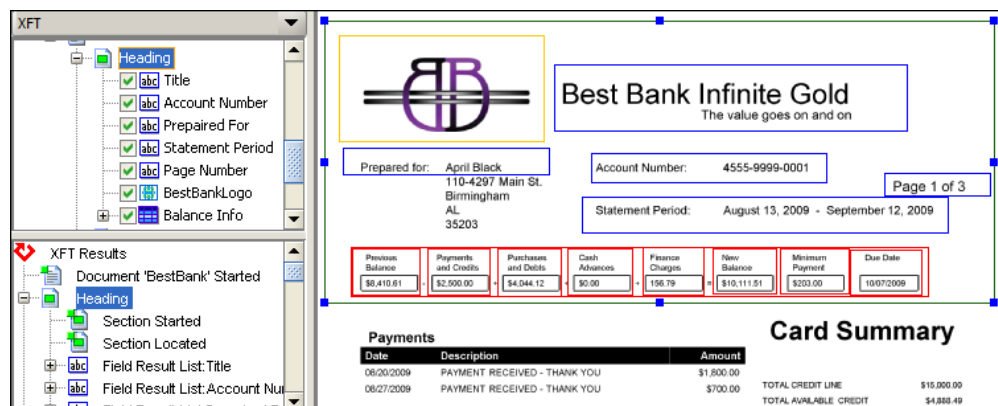


Figure 6-24: Field definition example

In this example, the text fields are defined by blue boxes, the object field is defined by a yellow box, and the table fields are defined by red boxes. The section box is green.

Data Extraction

Output Transformation Designer's comprehensive field definition capability gives you the ability to extract data from any area of the printstream, according to the rules you specify.

Documents are not always rigid; sections may be of variable size depending on how much data exists in the previous sections, and your printstream may consist of a number of types of documents. But because documents and sections are started and ended based on the presence of locators, data extraction is precise and accurate, no matter how many lines are contained within a section or how many pages exist within any logical document. This gives you a high degree of flexibility with regard to your data, enabling multiple extraction and output possibilities.

Related Links

- [“Data Extraction” on page 347](#)
- [“Document Breaking Component” on page 262](#)
- [“Table of Contents Component” on page 330](#)
- [“Indexer Component” on page 281](#)
- [“Index Writer Component” on page 272](#)
- [“Adding a Text Field” on page 393](#)
- [“Adding an Object Field” on page 401](#)
- [“Adding a Table” on page 432](#)

6.15.1.7 Field Definition Terminology

To understand field definition, it is important to understand the following terms:

Term	Definition
File	A unit of information. Each file may contain one or more documents.
Document	One or more pages, across which one or more sections are defined. For more information, see “Adding a Document” on page 370 .
Section	A defined piece of a page or document that has one or more fields associated with it. Sections can have start and finish identifiers to determine whether a section is active in the current page of the document. For more information, see “Adding a Section” on page 375 .
Field	A defined location in a section or page of a document, bounded by a box whose position and dimensions are specified in terms of fixed offsets within a page, or based on offsets from a locator. For more information, see: “Adding a Text Field” on page 393 “Adding an Object Field” on page 401 “Adding a Table” on page 432

Bounding Box	A rectangle that encloses data. The bounding box is established by its position and dimensions, expressed in X/Y coordinates or as measurements in relation to a locator. Using XFT in Output Transformation Designer, you can draw a bounding box on a graphical representation of your page or section to make defining the field easier. There are three types of fields: text fields, object fields, and table fields, all of which have thresholds that determine how data within the field is processed. To help you establish field coordinates, the XFT Field Technology component displays a graphical page view of your page, which you can then use to place field locations.  Note: Screen access to this graphical view is only through the XFT window or the Page Viewer; it is not an actual graphic file similar to what you can generate using the TIFF Generator. For more information, see: “Page View Tab” on page 90 or “TIFF Generator” on page 228.
Identifiers	Words or phrases which are sought within the file to determine when a document starts or ends. There can be many identifiers associated with the start or end of a document. For more information, see “Identifiers” on page 345
Locator	A search string pattern found within all text on the page. Locators are a means of identifying the presence of a word or phrase on a page. You can define zero, one or many locators for a document. Once a locator has been defined it can be linked to a document, section, or page in such a way that it identifies the start or end of the object. For more information, see “Adding a Text Locator” on page 486.

Thresholds	Defines how the data in a text field is structured into words and paragraphs. Thresholds group single letters into words, words into phrases, and phrases into paragraphs, depending on the threshold settings that indicate how close two text elements must be to one another to be considered part of the same word, phrase or paragraph. For more information, see “Adding a Threshold” on page 462.
Data	Information mined from within the bounding box of a field.
Field Validation	Confirms that data in a field are valid dates or numbers as required.
Confidence Factor	Each Identifier is assigned a Confidence Factor. This represents the degree of certainty that the occurrence of the word or phrase correctly identifies the start or end of the section. For more information, see “Confidence Factor” on page 345.

6.15.2 XFT Window

To display the XFT window, double click the XFT Field Technology component in the Development window. The XFT window consists of five main areas:

- **XFT Window toolbar.** Provides button shortcuts to menu items, as well as tools to help you customize the view of the page and define your documents, sections, and fields.
- **XFT Design pane.** Displays a tree view of the documents, sections, fields, locators, and thresholds that you define for your project.
- **XFT Results pane.** Displays a list of the documents and used locators.
- **Preview pane.** Presents a graphical image of the image file, on which you can draw the sections, fields, and locators. When any of these items are selected in the Preview pane, its associated properties are displayed in the Properties pane.
- **DOM Viewer pane.** Displays a tree view of all DOM nodes on the page. This pane is only available if DOM properties are enabled in the document currently shown in the Preview pane.
- **Status bar.** Displays the cursor coordinates (X, Y), representing the cursor location relative to the coordinates of the image rather than the screen. As such, each pixel on the screen may represent several pixels in the image.

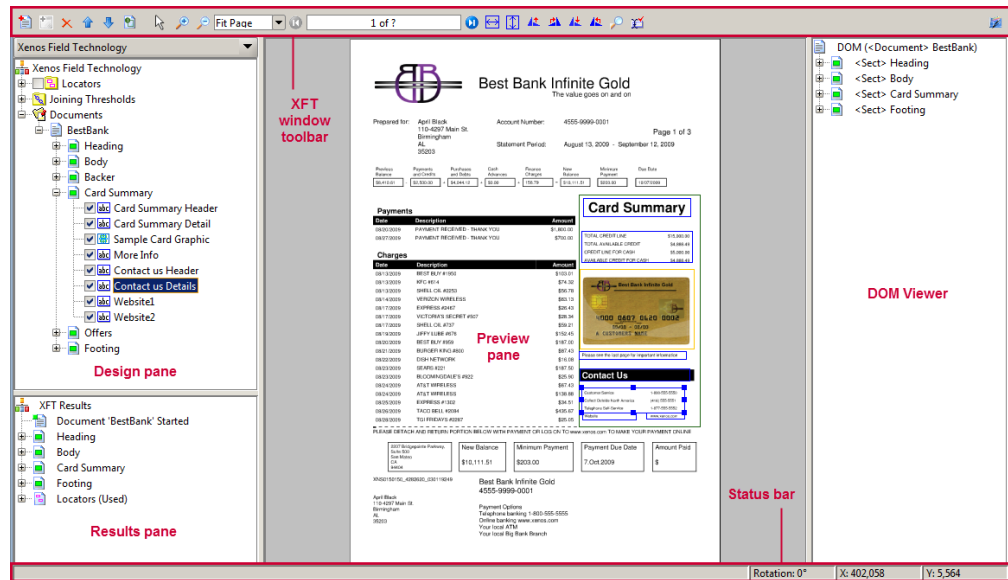


Figure 6-25: XFT window

6.15.2.1 XFT Window Toolbar

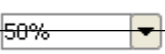
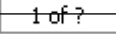
The XFT window toolbar is designed to help you be more efficient while defining the page elements within your documents where data is extracted for use in your printstreams. It also contains display options so that you can customize your view of the pages within your document.



Figure 6-26: XFT window toolbar

Main Toolbar

Icon	Description
	Add New. Opens the Add New menu where you can select the type of page element to add to your document. Some options are only enabled when you have selected the prerequisite page element type in the Design pane.
	Add Group. Allows you to group XFT fields of the same type together. For more information, see “Adding a Group” on page 479 .

	Delete Selected Item. Removes the item you have highlighted in the XFT tree.
	Move Up. Moves the selected field definition up one position in the XFT tree. Its position determines the order in which the field definition is processed.
	Move Down. Moves the selected field definition down one position in the XFT tree. Its position determines the order in which the field definition is processed.
	Show/Hide Order. Toggles the XFT element reading order in the Preview pane when DOM is enabled for the selected element in the Design pane.
	Select Tool. Changes your cursor to the pointer tool, and highlights the item you click on. Use this for selecting a bounding box in the page view.
	Zoom In. Changes your cursor to the Zoom In tool, which increases the document magnification by 5% with each click in the preview pane.
	Zoom Out. Changes your cursor to the Zoom Out tool, which decreases the document magnification by 5% with each click in the preview pane.
	Magnification. Specifies the magnification level at which to view the page. You can use the drop-down menu to select a magnification level, or enter the desired magnification in the field.
	Previous Page. Switches the view to the previous page.
	Page Number. Indicates the current page number and if available, the total number of pages in your document. You can also enter a page number to go directly to a specific page.
	Next Page. Switches the view to the next page.
	Fit Page Width. Adjusts the magnification in the preview pane so that the full width of the page can be seen.
	Fit Page Height. Adjusts the magnification in the preview pane so that the full height of the page can be seen.

	Rotate 0° Up. Reorients the preview image to the original 0°, upright view.
	Rotate 90° Right. Reorients the preview image from 0° to the right by 90°.
	Rotate 180° Down. Reorients the preview image down by 180°.
	Rotate 90° Left. Reorients the preview image from 0° to the left by 90°.
	Find Rotate 90° Left. Reorients the preview image from 0° to the left by 90°. Opens the Find dialog which allows you to search for specific text on the page.
	Filter. Opens the Page View Filter dialog where you can select which element types to display on the page. Elements that can be hidden include text, images, lines, barcodes, shading, forms, and vectors.

Add New Menu

Icon	Description
	Add Threshold. Opens the Threshold Properties window where you define how pieces of text are joined together to make words, phrases, and paragraphs. For more information, see “Adding a Threshold” on page 462.
	Add Text Locator. Allows you to draw a bounding box around any area in your document to create a text-based locator, which are identified by magenta borders. Once a locator is defined, the XFT Text Locator Properties window appears. For more information, see “Adding a Text Locator” on page 486.
	Add Image Locator. Allows you to draw a bounding box around any area in your document to create an image-based locator, which are identified by magenta borders. Once a locator is defined, the XFT Image Locator Properties dialog appears. For more information, see “Adding a Text Locator” on page 486.

	Add Document. Opens the XFT Document Properties dialog window where you input the specifications of your document. A document consists of a group of sections, fields, and any locators. You must define a document before you can define sections or fields. For more information, see “Adding a Document” on page 370 .
	Add Section. Allows you to draw a bounding box around any area on the document to create a section. Sections may overlap or even be created within the bounding box of another section. Once you have defined a section, the XFT Section Properties window appears. For more information, see “Adding a Section” on page 375 .
	Add Text Field. Allows you to draw a bounding box around any area on the document to create a text field. You must define at least one section before you can create a text field. Once you have defined a section, the XFT Text Field Properties window appears. For more information, see “Adding a Text Field” on page 393 .
	Add Object Field. Allows you to draw a bounding box around any area on the document to create an object field. You must define at least one section before you can create an object field. Once you have defined a section, the XFT Object Field Properties window appears. For more information, see “Adding an Object Field” on page 401 .
	Add Form Field. Allows you to draw a bounding box to define a form field, which must be within the boundaries of the selected section. Once a form field is defined, the XFT Form Field Properties dialog appears. For more information, see “Adding a Form Field” on page 409 .
	Add List Field. Allows you to draw a bounding box to define a list field, which must be within the boundaries of the selected section. Once a list field is defined, the XFT List Field Properties dialog appears. For more information, see “Adding a List Field” on page 417 .

	Add Table. Allows you to draw a bounding box around any area on the document to create a table. This is particularly useful for data that is aligned in rows and columns. Once you have drawn the table, the Table Properties window appears. For more information, see “Adding a Table” on page 432.
	Add Table Column. Allows you to draw a bounding box in an area of a table that you want to identify as a column. Defining columns determines how the data will be located, extracted, and validated. Once you have drawn the column, the Column Properties window appears. For more information, see “Adding a Column” on page 451.
	Add Cell. Allows you to define a group of fields within the boundaries of the selected table. Once a cell is defined, the XFT Cell Properties dialog appears. For more information, see “Adding a Cell” on page 452.
	Add Container. Allows you to define one or more grouped fields or tables, which must be within the boundaries of the selected section. Once a container is defined, the XFT Container Properties dialog appears. For more information, see “Adding a Container” on page 465.

6.15.2.2 DOM Viewer

The DOM Viewer pane displays a tree view of all DOM nodes contained on a page and is only available if DOM properties are enabled within your document. You can view the structure of the DOM nodes for section and other field definitions on the current page within the tree view and expand/collapse the branches to further examine the assigned properties. Right-clicking anywhere within the DOM Viewer displays a context menu where you can choose to **Collapse All** or **Expand All** the DOM nodes in the tree.

Each node in the DOM tree displays its specified tag type, enclosed in angled brackets, as well as the original XFT tag name for cross-referencing the DOM node to the XFT field definition. You can also select a node in the DOM tree and all related Xif elements are highlighted in light blue in the Preview pane. Furthermore, upon selection of a node, you can view the node's properties, such as the field definition's text or alternate text values, in the Properties window.

If any DOM nodes appear dimmed, they are present in the document currently displayed in the Preview pane, but not associated with any elements on the current page. If any DOM node appears in red, it indicates that the node has no graphical

elements associated within the document. For instance, this could happen when an XFT element failed to capture content on the page.

For more information on setting up DOM properties for field definitions, see [“Getting Started with XFT” on page 356](#) and referencing the sections on field definition types.

6.15.3 Getting Started with XFT

The topics in this section describe the actions you can perform while working with XFT.

6.15.3.1 Creating a Field Definition

After you have defined a parser for your project and specified an input file, you can begin working with (XFT).

To create a field definition for text or an object, you must first add the component to your project flow:

1. In Output Transformation Designer, select the **component** from the Components Sidebar Menu and drag it into the Development window. For more information, see [“Component Sidebar Menu” on page 110](#).
2. You can proceed to connect the component to a parser and generators, or you may opt to configure the XFT first.
3. Double-click the **component** icon.

The **XFT window** opens in a new tab.



Note: You must identify an input file before the component is able to render a preview page view for field definitions in the XFT window.

4. Add the required XFT documents, sections, fields, and tables. For more information on how to perform these tasks, please see the appropriate sections within [“Getting Started with XFT” on page 356](#).



Note: If you open an existing project with sections and fields already defined, it takes a little time for the XFT to load in the page view. This is due to the engine needing to perform data extraction on the file in order to provide information for the field list.

6.15.3.2 Handling Field Definitions

Creating a Field Definition

After you have defined a parser for your project and specified an input file, you can begin working with (XFT).

To create a field definition for text or an object, you must first add the component to your project flow:

1. In Output Transformation Designer, select the **component** from the Components Sidebar Menu and drag it into the Development window. For more information, see [“Component Sidebar Menu” on page 110](#).
2. You can proceed to connect the component to a parser and generators, or you may opt to configure the XFT first.
3. Double-click the **component** icon.

The **XFT window** opens in a new tab.



Note: You must identify an input file before the component is able to render a preview page view for field definitions in the XFT window.

4. Add the required XFT documents, sections, fields, and tables. For more information on how to perform these tasks, please see the appropriate sections within [“Getting Started with XFT” on page 356](#).



Note: If you open an existing project with sections and fields already defined, it takes a little time for the XFT to load in the page view. This is due to needing to perform data extraction on the file in order to provide information for the field list.



Note: For more information about the XFT window, see [“XFT Window” on page 350](#).

Organizing Field Definitions

Field definitions are processed in the same order that they appear in the XFT tree within the Design pane so as you develop your XFT documents, you must take into consideration how your field definitions are organized. For instance, you will most likely want to put a footer section following the processing of a header or body section since footers occur at the bottom of a page and should be among the last definitions processed.

Moving Field Definitions

Field definition types can be repositioned in multiple ways in the Design pane of the XFT window. You can only perform these functions on the following field definition types:

- Sections
- Text
- Object
- List
- Table
- Form Fields

Right-clicking on one of these XFT parm objects displays a context menu where you can shift the object one position using the Up, , or Down, , options. Alternatively, you can also drag objects to your desired position in the Design pane.

For increased efficiency while designing your XFT templates in the XFT window, these field definition types can also be copied and pasted, which is particularly handy if you wish to create a duplicate of the selected field definition. If you are copying a section, all objects included in the section are copied as well. To copy and paste a field definition:

1. In the Design pane of the XFT window, select the section, text field, list field, or object field that you want to copy.
2. You can select the copy command either by pressing **Ctrl + C** or right-clicking on the field definition and from the context menu that appears, selecting **Copy**.
The field definition is copied to the clipboard.
3. In the Design pane, navigate to and select the section where you wish to paste the copied field definition.
4. Paste your field definition either by pressing **Ctrl + V** or right-clicking on the field definition and from the context menu that appears, selecting **Paste**. (The Paste command only appears in the context menu if there is an appropriate item to paste.)

The field definition is pasted to the selected location. The name of the new field definition is the original name appended by **-copy**.

Copying and Pasting a Field Definition

For increased efficiency while designing your XFT templates in the XFT window, some field definition types can be copied and pasted. The copy command creates a duplicate of the selected field definition. Then, the paste command inserts the field definition at the selected position, which can help you quickly and accurately build your templates. You can only perform these functions on the following field definition types:

- Sections
- Text fields
- List fields

- Object fields

To copy and paste a field definition:

1. In the Design pane of the XFT window, select the section, text field, list field, or object field that you want to copy.
2. You can select the copy command either by pressing **Ctrl + C** or right-clicking on the field definition and from the context menu that appears, selecting **Copy**. The field definition is copied to the clipboard.
3. In the Design pane, navigate to and select the section where you wish to paste the copied field definition.
4. Paste your field definition either by pressing **Ctrl + V** or right-clicking on the field definition and from the context menu that appears, selecting **Paste**. (The Paste command only appears in the context menu if there is an appropriate item to paste.)

The field definition is pasted to the selected location. The name of the new field definition is the original name appended by **-copy**.

6.15.3.3 Auto-Detecting and Splitting Sections, Fields, or Table Columns

When creating field definitions for your input file, an auto detect tool is available for the automatic discovery of some field definition types. The auto detect tool can also split existing field definitions types to help you refine the ways data is extracted from your documents. Auto detection and splitting can be performed on the following types of field definitions:

- Sections
- Text, object, and list fields
- Columns

6.15.3.3.1 Detecting Sections

The **Detect Sections** tool is used for the auto-detection of sections within a document. After defining a new document and setting the properties, you are asked whether you wish to use the auto-detect feature to locate sections. If you click **Yes**, the **Detection Options** dialog displays. The Detect Sections tool can also be used on existing documents by right-clicking the document's entry in the tree on the Design pane and selecting Detect Sections.

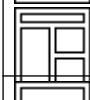
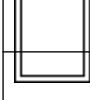
Detect Sections Detection Options

The Detection Options dialog is broken down into two areas: **Split To Try To Get** and **View Results**. The Split To Try To Get segment presents options on how your document should be divided, while the View Results segment displays the sections that were identified by the XFT component according to your settings. Be aware that the Preview pane of the XFT window changes dynamically to display the sections found as you modify the settings in the Detection Options dialog box. Therefore, if

the dialog obstructs your view of the document in the Preview pane, you should drag the dialog to another part of the screen.

In the **Split To Try To Get** segment, use the options to select the direction(s) that you want the document to be split and the number of sections to try to find in your document. The following options are available:

- The four page layouts designate the type of arrangement to use when trying to split your document into sections. You can select from the following:

Layout	Description
	Divides the page into horizontal sections, running from one side of the page to the other.
	Divides the page into vertical sections, running from the top of the page to the bottom.
	Divides the page into the specified number of sections using a mix of rows and columns.
	Specifies one section covering the whole page is to be created. Since no division of sections is occurring, the Number slider is disabled.

- Number.** Designates the number of sections that the auto detect tool will attempt to locate. Move the slider left and right to decrease or increase the number of sections. (Whether the tool will examine your document for horizontal, vertical, or mixed sections depends on your selection above.) As you move the slider, the amount of sections located is shown below in the Number Detected field. The tool cannot always find as many sections as you desire, but will try to come as close as possible. If the number of sections detected is not changing as you move the slider, it means that your particular number of sections cannot be detected; you can try to change the page layout type to see if you can achieve better results. If you are unsure of how many sections to divide the page into, you can select the **Automatic** check box.
- Automatic.** Allows the auto detect tool to determine the optimal number of sections to split your document into by itself. When this option is selected, the Number slider is disabled.

In the **View Results** segment, the regions detected are listed. The following options are available:

- Number Detected.** Indicates the number of sections detected in your document. This field is controlled by the Number slider above.
- Detected Regions.** Displays a breakdown of the sections found according to your settings in the Split To Try To Get segment. The names used in the dropdown menu for each section follow a Detected <Section_Name> pattern, with the <Section_Name> portion being some identifying text found near the beginning of

the detected section. After accepting the detected sections, the names for each section shown in the XFT tree in the Design pane follow a `<DocumentName>_Section_<SequentialNumber>` pattern, but these can be renamed by changing the **Name** on the **General** tab within the XFT Section Properties. You can select a section from the dropdown menu to highlight it in the Preview pane; this can be useful if you are having difficulty determining the boundaries for a particular section.

As you modify the properties on the Detection Options dialog, you can see the auto detection results in the Preview pane. Once you have reached a satisfactory result, click **OK** to create the document along with the detected sections, which will be shown in the Design pane. If you cannot obtain an acceptable result, it is recommended you choose the results that are closest to your desired arrangement and try to manually adjust the region boundaries or use the Split Section tool. If none of the suggestions are close, then click **Cancel** to discard all suggestions and then use **Add Section** to individually add each section you want.

Related Topics

- “Section Properties General Tab” on page 376
- “Splitting Sections” on page 361
- “Adding a Section” on page 375

6.15.3.3.2 Splitting Sections

Sections defined within your document can also be split into smaller sections using the Split Section option. This feature is particularly handy if the original attempt with Detect Sections did not provide adequate results or you simply wish to fine-tune the sections in an existing document.

To split a section:

1. In the XFT tree on the Design pane, open the top level Documents node and then expand the document node containing the section you wish to split.
The document’s field definitions appear.
2. Within your document’s expanded contents, locate the section you want to split and click on the section name.
The section is highlighted on the sample image of your document in the Preview pane.
3. Right-click on the section’s node in the Design pane and from the context menu that appears, select **Split Section**.
The **Detection Options** dialog appears.
4. Adjust the settings in the Detection Options window and when you are finished, click **OK**. The settings available in the Detection Options dialog are described below.

Split Sections Detection Options

When you are splitting sections, remember that all detection functions are only performed inside your currently selected section. The Detection Options dialog is broken down into two areas: **Split To Try To Get** and **View Results**. The Split To Try To Get segment presents options on how your section should be divided, while the View Results segment displays the subsequent sections that were identified by the XFT component according to your settings. Be aware that the Preview pane of the XFT window changes dynamically to display the sections found as you modify the settings in the Detection Options dialog box. Therefore, if the dialog obstructs your view of the document in the Preview pane, you should drag the dialog to another part of the screen.

In the **Split To Try To Get** segment, use the options to select the direction(s) that you want the section to be split in and the number of subsequent sections to try to find within your current section. The following options are available:

- The three layouts designate the type of arrangement to use when trying to split your current section into subsequent sections. You can select from the following:

Layout	Description
	Divides the section into smaller horizontal sections.
	Divides the section into smaller vertical sections.
	Divides the section into the specified number of sections using a mix of rows and columns.

- Number.** Designates the number of sections that the auto detect tool will attempt to locate. Move the slider left and right to decrease or increase the number of sections. (Whether the tool will examine your document for horizontal, vertical, or mixed sections depends on your selection above.) As you move the slider, the amount of sections located is shown below in the Number Detected field. The tool cannot always find as many sections as you desire, but will try to come as close as possible. If the number of sections detected is not changing as you move the slider, it means that your particular number of sections cannot be detected; you can try to change the page layout type to see if you can achieve better results. If you are unsure of how many sections to divide the page into, you can select the **Automatic** check box.
- Automatic.** Allows the auto detect tool to determine the optimal number of sections to split your document into by itself. When this option is selected, the Number slider is disabled.

In the **View Results** segment, the sections detected are listed. The following options are available:

- Number Detected.** Indicates the number of sections detected in your document. This field is controlled by the Number slider above.

- **Detected Regions.** Displays a breakdown of the sections found according to your settings in the Split To Try To Get segment. The names used in the dropdown menu for each section follow a Detected <Section_Name> pattern, with the <Section_Name> portion being some identifying text found near the beginning of the detected section. After accepting the detected sections, the names for each section shown in the XFT tree in the Design pane follow a <DocumentName>_Section_<SequentialNumber> pattern, but these can be renamed by changing the **Name** on the **General** tab within the XFT Section Properties. You can select a section from the dropdown menu to highlight it in the Preview pane; this can be useful if you are having difficulty determining the boundaries for a particular section. For more information, see “Section Properties General Tab” on page 376.

As you modify the properties on the Detection Options dialog, you can see the auto detection results in the Preview pane. Once you have reached a satisfactory result, click **OK** to create the new sections within the selected section, which will be shown in the Design pane.

6.15.3.3.3 Detecting Fields

The Detect Fields tool is used for the auto-detection of text, object, or list fields within a section. After defining a new section and setting the properties, you are asked whether you wish to use the auto-detect feature to locate fields. If you click **Yes**, the **Detection Options** dialog displays. The Detect Fields tool can also be used on existing sections by right-clicking the section’s entry in the tree on the Design pane and selecting **Detect Fields**.

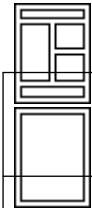
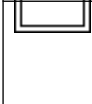
Detect Fields Detection Options

The Detection Options dialog is broken down into three areas: **Split To Try To Get**, **Return**, and **View Results**. The Split To Try To Get segment presents options on how your section should be divided. The Return segment specifies the types of fields to detect, while the View Results segment displays the fields that were identified by the XFT component according to your settings. Be aware that the Preview pane of the XFT window changes dynamically to display the fields found as you modify the settings in the Detection Options dialog box. Therefore, if the dialog obstructs your view of the document in the Preview pane, you should drag the dialog to another part of the screen.

In the **Split To Try To Get** segment, use the options to select the direction(s) for your detected fields and the number of fields to try to find in your section. The following options are available:

- The four layouts designate the type of arrangement to use when trying to detect fields in your sections. You can select from the following:

	Layout	Description
		Divides the section into horizontal fields.
		Divides the section into vertical fields.

	Divides the section into the specified number of fields using a mix of horizontal and vertical fields.
	Specifies one field covering the whole section. Since no division of the section is occurring, the Number slider is disabled.

- **Number.** Designates the number of fields that the auto detect tool will attempt to locate. Move the slider left and right to decrease or increase the number of fields. (Whether the tool will examine your section for horizontal, vertical, or a mix of fields depends on your selection above.) As you move the slider, the amount of fields located is shown below in the Number Detected field. The tool cannot always find as many fields as you desire, but will try to come as close as possible. If the number of fields detected is not changing as you move the slider, it means that your particular number of fields cannot be detected; you can try to change the layout type to see if you can achieve better results. If you are unsure of how many fields to divide the section into, you can select the **Automatic** check box.
- **Automatic.** Allows the auto detect tool to determine the optimal number of fields to split your section into by itself. When this option is selected, the Number slider is disabled.

In the **Return** segment, you must indicate the field types to locate. The following options are available:

- **Text Fields.** Specifies that the auto detect tool should return instances of text fields.
- **Images.** Specifies that the auto detect tool should return instances of image fields.
- **Bar codes.** Specifies that the auto detect tool should return instances of bar code or other object fields.

In the **View Results** segment, the regions detected are listed. The following options are available:

- **Number Detected.** Indicates the number of fields detected in your document. This field is controlled by the Number slider above.
- **Detected Regions.** Displays a breakdown of the fields found according to your settings in the Split To Try To Get segment. The names used in the dropdown menu for each field follow a Detected <Field_Name> pattern, with the <Field_Name> portion being some identifying text found near the beginning of the detected field. After accepting the detected fields, the names for each field shown in the XFT tree in the Design pane follow a <DocumentName>_<FieldType>_<SequentialNumber> pattern, but these can be renamed by changing the **Name** on the General tab found on the respective field type's properties screen. You can select a field from the dropdown menu to highlight it in the Preview pane; this can be useful if you are having difficulty determining the location of a particular field.

As you modify the properties on the Detection Options dialog, you can see the auto detection results in the Preview pane. Once you have reached a satisfactory result,

click **OK** to create the fields, which will be shown in the Design pane. If you cannot obtain an acceptable result, it is recommended you choose the results that are closest to your desired arrangement and try to manually adjust the region boundaries or use the Split Field tool. If none of the suggestions are close, then click **Cancel** to discard all suggestions and then use respective adding a field topic for your field type to individually add each field you want.



Note: For more information on the Split Field tool, see “[Splitting Fields](#)” on page 365.

6.15.3.3.4 Splitting Fields

Fields defined within your sections can also be split into smaller divisions using the Split Section option. This feature is particularly handy if the original attempt with Detect Fields did not provide adequate results or you simply wish to fine-tune the fields in an existing document.

To split a field:

1. In the XFT tree on the Design pane, open the top level Documents node and then expand both the document and section node containing the field you wish to split.

The section’s field definitions appear.

2. Within your section’s expanded contents, locate the field you want to split and click on the field name.

The field is highlighted on the sample image of your document in the Preview pane.

3. Right-click on the field’s node in the Design pane and from the context menu that appears, select **Split Field**.

The **Detection Options** dialog appears.

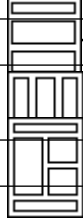
4. Adjust the settings in the Detection Options window and when you are finished, click OK.

Split Fields Detection Options

When you are splitting fields, remember that all detection functions are only performed inside your currently selected field. The Detection Options dialog is broken down into three areas: **Split To Try To Get**, **Return**, and **View Results**. The Split To Try To Get segment presents options on how your field should be divided. The Return segment specifies the types of fields to detect, while the View Results segment displays the fields that were identified by the XFT component according to your settings. Be aware that the Preview pane of the XFT window changes dynamically to display the sections found as you modify the settings in the Detection Options dialog box. Therefore, if the dialog obstructs your view of the document in the Preview pane, you should drag the dialog to another part of the screen.

In the **Split To Try To Get** segment, use the options to select the direction(s) that you want the field to be split in and the number of subsequent fields to try to find within your current field. The following options are available:

- The three sample layouts designate the type of arrangement to use when trying to split your current field into subsequent fields. You can select from the following:

	Layout	Description
		Divides the field into smaller horizontal fields.
		Divides the field into smaller vertical fields.
		Divides the field into the specified number of fields using a mix of horizontal and vertical fields.

- **Number.** Designates the number of fields that the auto detect tool will attempt to locate. Move the slider left and right to decrease or increase the number of sections. (Whether the tool will examine your field for horizontal, vertical, or mixed fields depends on your selection above.) As you move the slider, the amount of fields located is shown below in the Number Detected field. The tool cannot always find as many fields as you desire, but will try to come as close as possible. If the number of fields detected is not changing as you move the slider, it means that your particular number of fields cannot be detected; you can try to change the layout type to see if you can achieve better results. If you are unsure of how many fields to divide the current field into, you can select the **Automatic** check box.
- **Automatic.** Allows the auto detect tool to determine the optimal number of fields to split your current field into by itself. When this option is selected, the Number slider is disabled.

In the **Return** segment, you must indicate the field types to locate. The following options are available:

- **Text Fields.** Specifies that the auto detect tool should return instances of text fields.
- **Images.** Specifies that the auto detect tool should return instances of image fields.
- **Bar codes.** Specifies that the auto detect tool should return instances of bar code or other object fields.

In the **View Results** segment, the regions detected are listed. The following options are available:

- **Number Detected.** Indicates the number of sections detected in your document. This field is controlled by the Number slider above.
- **Detected Regions.** Displays a breakdown of the fields found according to your settings in the Split To Try To Get segment. The names used in the dropdown

menu for each field follow a Detected <Field_Name> pattern, with the <Field_Name> portion being some identifying text found near the beginning of the detected section. After accepting the detected sections, the names for each section shown in the XFT tree in the Design pane follow a <DocumentName>_<FieldType>_<SequentialNumber> pattern, but these can be renamed by changing the **Name** on the General tab found on the respective field type's properties screen. You can select a field from the dropdown menu to highlight it in the Preview pane; this can be useful if you are having difficulty determining the boundaries for a particular field.

As you modify the properties on the Detection Options dialog, you can see the auto detection results in the Preview pane. Once you have reached a satisfactory result, click **OK** to create the new fields within the selected field, which will be shown in the Design pane.

6.15.3.3.5 Detecting Table Columns

The Detect Table Columns tool is used for the auto-detection of columns within a table. After defining a new table and setting the properties, you are asked whether you wish to use the auto-detect feature to locate table columns. If you click **Yes**, the **Detection Options** dialog displays. The Detect Columns tool can also be used on existing tables by right-clicking the table's entry in the tree on the Design pane and selecting **Detect Table Columns**.

Detect Table Columns Detection Options

The Detection Columns dialog is broken down into two areas: **Split To Try To Get** and **View Results**. The Split To Try To Get segment presents options on how your table should be divided, while the View Results segment displays the columns that were identified by the XFT component according to your settings. Be aware that the Preview pane of the XFT window changes dynamically to display the columns found as you modify the settings in the Detection Options dialog box. Therefore, if the dialog obstructs your view of the document in the Preview pane, you should drag the dialog to another part of the screen.

In the **Split To Try To Get** segment, use the options to select the number of columns and the minimum distance between columns in your table. The following options are available:

- **Number of Columns.** Designates the number of columns that the auto detect tool will attempt to locate within your table. Move the slider left and right to decrease or increase the number of columns. As you move the slider, the amount of columns located is shown below in the Number Detected field. The tool cannot always find as many columns as you desire, but will try to come as close as possible. If the number of columns detected is not changing as you move the slider, it means that your particular number of columns cannot be detected. If you are unsure of how many columns to divide the current table into, you can select the **Automatic** check box.

- **Automatic.** Allows the auto detect tool to determine the optimal number of columns to split your current table into by itself. When this option is selected, the Number of Columns slider is disabled.
- **Min Gap Between Cols (pts).** Indicates the minimum amount of space, in pixels, that must exist between detected columns.

In the **View Results** segment, the columns detected are listed. The following options are available:

- **Number Detected.** Indicates the number of columns detected in your table. This field is controlled by the Number of Columns slider above.
- **Detected Columns.** Displays a breakdown of the columns found according to your settings in the Split To Try To Get segment. The names used in the dropdown menu for each column follow a Detected <Column_Name> pattern, with the <Column_Name> portion being some identifying text found near the beginning of the detected column. After accepting the detected columns, the names for each column shown in the XFT tree in the Design pane can be renamed by changing the **Name** on the **General** tab found on the properties screen. You can also select a column from the dropdown menu to highlight it in the Preview pane; this can be useful if you are having difficulty determining the location of a particular column.

As you modify the properties on the Detection Options dialog, you can see the auto detection results in the Preview pane. Once you have reached a satisfactory result, click **OK** to create the table columns, which will be shown in the Design pane. If you cannot obtain an acceptable result, it is recommended you choose the results that are closest to your desired arrangement and try to manually adjust the region boundaries or use the Split Columns tool. If none of the suggestions are close, then click Cancel to discard all suggestions and then use Add Column to individually add each column you want.

Related Links

- [“Column Properties General Tab” on page 451](#)
- [“Splitting Table Columns” on page 368](#)
- [“Adding a Column” on page 451](#)

6.15.3.3.6 Splitting Table Columns

Columns defined within your tables can also be split into smaller divisions using the Split Column option. This feature is particularly handy if the original attempt with Detect Columns did not provide adequate results or you simply wish to fine-tune the fields in an existing document.

To split a column:

1. In the XFT tree on the Design pane, open the top level Documents node and then expand the document, section, and table node containing the column you wish to split.

The table's field definitions appear.

2. Within your table's expanded contents, locate the column you want to split and click on the column name.

The column is highlighted on the sample image of your document in the Preview pane.

3. Right-click on the column's node in the Design pane and from the context menu that appears, select **Split Column**.

The **Detection Options** dialog appears.

4. Adjust the settings in the Detection Options window and when you are finished, click OK.

Split Columns Detection Options

When you are splitting columns, remember that all detection functions are only performed inside your currently selected column. The Detection Options dialog is broken down into two areas: **Split To Try To Get** and **View Results**. The Split To Try To Get segment presents options on how your column should be divided, while the View Results segment displays the columns that were identified by the XFT component according to your settings. Be aware that the Preview pane of the XFT window changes dynamically to display the sections found as you modify the settings in the Detection Options dialog box. Therefore, if the dialog obstructs your view of the document in the Preview pane, you should drag the dialog to another part of the screen.

In the **Split To Try To Get** segment, use the options to select the number of columns and the minimum distance between columns in your table. The following options are available:

- **Number of Columns.** Designates the number of columns that the auto detect tool will attempt to locate within your column. Move the slider left and right to decrease or increase the number of columns. As you move the slider, the amount of columns located is shown below in the Number Detected field. The tool cannot always find as many columns as you desire, but will try to come as close as possible. If the number of columns detected is not changing as you move the slider, it means that your particular number of columns cannot be detected. If you are unsure of how many columns to divide the current table into, you can select the **Automatic** check box.
- **Automatic.** Allows the auto detect tool to determine the optimal number of columns to split your current table into by itself. When this option is selected, the Number of Columns slider is disabled.
- **Min Gap Between Cols (pts).** Indicates the minimum amount of space, in pixels, that must exist between detected columns.

In the **View Results** segment, the columns detected are listed. The following options are available:

- **Number Detected.** Indicates the number of columns detected in your document. This field is controlled by the Number of Columns slider above.
- **Detected Columns.** Displays a breakdown of the columns found according to your settings in the Split To Try To Get segment. The names used in the dropdown menu for each column follow a Detected <Column_Name> pattern, with the <Column_Name> portion being some identifying text found near the beginning of the detected column. After accepting the detected columns, the names for each column shown in the XFT tree in the Design pane can be renamed by changing the **Name** on the **General** tab found on the properties screen. You can also select a column from the dropdown menu to highlight it in the Preview pane; this can be useful if you are having difficulty determining the location of a particular column.

As you modify the properties on the Detection Options dialog, you can see the auto detection results in the Preview pane. Once you have reached a satisfactory result, click **OK** to create the new columns within the selected column, which will be shown in the Design pane.



Note: For more information on the **General** tab, see “[Column Properties General Tab](#)” on page 451.

6.15.3.4 Adding a Document

To add an XFT document, select **Add Document** from the XFT window toolbar and the **XFT Document Properties** dialog appears.

Use the available tabs to establish some universal properties regarding your document. When you are finished making your selections, click **OK** and then the **Auto Detect Sections** dialog box appears. The Detect Sections tool can assist with auto-detecting the sections within your document and is a convenient way of completing a first attempt at adding sections to your new document. To use the tool, click **Yes** and the Detection Options dialog box for Detect Sections appears.

If you opt to not use the tool, the document is created and shown in the Design pane, but you will have to manually add and define the sections yourself. However, the Detect Sections tool can also be used on existing documents by right-clicking the document’s entry in the tree on the Design pane and from the context menu that appears, selecting Detect Sections.

For more information on working with this auto detect tool, see “[Detecting Sections](#)” on page 359.



Note: If you are using DOM properties, you must ensure that they are first enabled at the document level as the permissions propagate through to subsequent field definitions. For example, if you set up DOM properties at the section level, but not at the document level then the section’s DOM properties will not function.

6.15.3.4.1 Document Properties General Tab

The **General** tab specifies general properties for your document. It has the following fields:

- **Name.** Specifies a name for your document. The name must be unique within the project as it is used to reference this item in report logs.
- **Enable.** Enables document processing, which is required if you have more than one document defined. If it is disabled, the engine does not look for document identifiers or locators.
- **Options.** Determines which additional tabs are available; you can select from **Enable start identification**, **Enable end identification**, or **Enable DOM Properties**.



Notes

- If you are using identifiers for your document, you must first define the locators which act as identifiers.
- If you are working with Output Transformation for AppEnhancer, it is recommended you turn off DOM Properties to optimize performance. For more information about this, see *OpenText Output Transformation for AppEnhancer - Integration Guide (VDTOTS-H-AIG)*.
- **Sections.** Displays a list of the sections that have already been defined for your current document. If you are adding this document for the first time and have not yet identified any sections, this list is empty.

To review or edit additional properties, click on another tab in the XFT Document Properties window. When you have completed making your selections, click OK to accept your changes and then the Auto Detect Sections dialog appears. The Detect Sections tool can assist with auto-detecting the sections within your document and is a convenient way of completing a first attempt at adding sections to your new document. To use the tool, click Yes and the Detection Options dialog for Detect Sections appears. For more information on working with this tool, see [“Detecting Sections” on page 359](#).

6.15.3.4.2 Document Properties Start Identification Tab

The **Start Identification** tab specifies when a new document begins. It has the following fields:



Note: This tab is only active after you have selected the **Enable Start Identification** option on the General tab of the XFT Document Properties window.

- **Required confidence factor.** Indicates the confidence factor you have set for this identifier.

- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the **Locator Name** column of the Identifiers table.

To add a start identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu.
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator.
- **Form Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Document Properties window. When you have completed making your selections, click OK to accept your changes and then the Auto Detect Sections dialog appears. The Detect Sections tool can assist with auto-detecting the sections within your document and is a convenient way of completing a first attempt at adding sections to your new document. To use the tool, click Yes and the Detection Options dialog for Detect Sections appears.

Related Links

- [“Adding a Text Locator” on page 486](#)
- [“Confidence Factor” on page 345](#)
- [“Detecting Sections” on page 359](#)

6.15.3.4.3 Document Properties End Identification Tab

The **End Identification** tab specifies when a document concludes. The end identifier does not have to be on the same page as the start identifier. It has the following fields:



Note: This tab is only active after you have selected the **Enable end identification** option on the General tab of the XFT Document Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add an end identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu.
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator.
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Document Properties window. When you have completed making your selections, click OK to accept your changes and then the Auto Detect Sections dialog appears. The Detect Sections tool can assist with auto-detecting the sections within your document and is a convenient way of completing a first attempt at adding sections to your new document. To use the tool, click Yes and the Detection Options dialog for Detect Sections appears.

Related Links

- [“Adding a Text Locator” on page 486](#)
- [“Confidence Factor” on page 345](#)
- [“Detecting Sections” on page 359](#)

6.15.3.4.4 Document Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your document. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by ISO-639 language registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information is critical to the screen reader application.
- **Logical Order.** Defines how elements within the container are ordered as they are inserted. You can choose from **Static** or **Auto** ordering. With Static ordering, the order in which the elements are defined and processed in is used when inserting elements into the DOM. Use this option when the order within your document is known to be consistent. With Auto ordering, elements are ordered based on their graphic page position. This option is useful when the exact order of elements may vary depending on the position of the elements on the physical page.

To review or edit additional properties, click on another tab in the XFT Document Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Sections** dialog appears. The Detect Sections tool can assist with auto-detecting the sections within your document and is a convenient way of completing a first attempt at adding sections to your new document. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Sections appears. For more information on working with this tool, see [“Detecting Sections” on page 359](#).

6.15.3.5 Adding a Section

Sections are the assigned areas of your document that hold the fields, tables, locators, and threshold definitions.



Note: You must have first defined your XFT document settings before you can add an XFT section.

To add an XFT section:

1. From the XFT window toolbar, select **Add Section**, .
2. Move the cursor into the Preview pane and the cursor itself changes from an arrow to crosshairs. Using the crosshair cursor, click and drag to draw a bounding box around the area you want to establish as a section.
The **XFT Section Properties** window appears.
3. Complete the fields in the XFT Section Properties dialog and when you are done, click **OK**.

Use the available tabs to establish some universal properties regarding your section. When you are finished making your selections, click **OK** and then the **Auto Detect Fields** dialog box appears. The **Detect Fields** tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog box for Detect Fields appears.

If you opt to not use the tool, the section is created and shown in the Design pane, but you will have to manually add and define the fields yourself. However, the Detect Fields tool can also be used on existing sections by right-clicking the section's entry in the tree on the Design pane and from the context menu that appears, selecting **Detect Fields**.

For more information on working with this auto detect tool, see [“Detecting Fields” on page 363](#).



Note: DOM Properties must be enabled on your section in order for Auto Detect to work. If you have enabled Auto Detect but not DOM properties, then the auto detection features will not function properly.

6.15.3.5.1 Section Properties General Tab

The **General** tab specifies general properties for your document. It has the following fields:

- **Name.** Specifies a name for your section. The name must be unique within the project as it is used to reference this item in report logs.
- **Enable.** Indicates that section processing is enabled, otherwise, does not check for the presence of sections.
- **Options.** Determines which additional tabs are available/enabled. You can select from **Enable Start Identification**, **Enable End Identification**, **Enable Dynamic Positioning**, **Enable Multiple Sections per Page**, or **Enable Waterfall Fields Flow**. Enabling either of the identification types or dynamic positioning opens up further options on separate tabs, which are detailed in separate topics.

Dynamic positioning refers to sections that are located based on their position relative to locators you specify, even if they shift on the page. **Multiple sections per page** indicates to the engine that your document has more than one instance of the same section on a page. **Waterfall fields flow** regulates how multiple instances of the same field type (text, list, or tables) are detected within a section. When enabled, following the initial instance of a named field, all subsequent occurrences can only be located below or after the previous instance. The default waterfall fields flow setting is disabled, which means it is possible for multiple instances of a field type to partially overlap with each other.



Note: If you are using identifiers for your document, you must first define the locators which act as identifiers.

- **Location Behavior.** Determines locator behavior for the section.
 - **Reposition On Page Break.** When selected (which is the default value), the top locator can be used for positioning when the section continues on a subsequent page. When disabled, the fixed bounding box is used for the top location.
 - **Allow Locator Duplicates For Identification** and **Allow Locator Duplicates for Positioning.** When the section repeats on the page and its end cannot easily be determined, select these check boxes to use the start locator of the next section as the end locator and bottom positioner of the current section. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section and the start of the next section. These features are disabled by default.
- **Position and Size.** Specifies the default position and size of the section.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.

- **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Fields** dialog appears. The Detect Fields tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Fields appears. For more information on working with this tool, see [“Detecting Fields” on page 363](#).

6.15.3.5.2 Section Properties Start Identification Tab

The **Start Identification** tab specifies when a new section begins. It has the following fields:



Note: This tab is only active after you have selected the **Enable Start Identification** option on the General tab of the XFT Section Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the **Locator Name** column of the Identifiers table.

To add a start identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu.
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator.

Forms Filter. Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.

- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to

accept your changes and then the **Auto Detect Fields** dialog appears. The Detect Fields tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Fields appears.

Related Links

- [“Adding a Text Locator” on page 486](#)
- [“Confidence Factor” on page 345](#)
- [“Detecting Fields” on page 363](#)

6.15.3.5.3 Section Properties Positioning Tab

The **Positioning** tab allows you to set the properties for locating sections by basing each position of the section’s bounding box on a locator as specified in the Locator Priority list. This type of dynamic positioning is helpful since sections are located based on their position relative to locators you specify, even if they move on the page.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator’s search string is used as the starting point from which the section’s position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.
- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section’s original bounding box position.

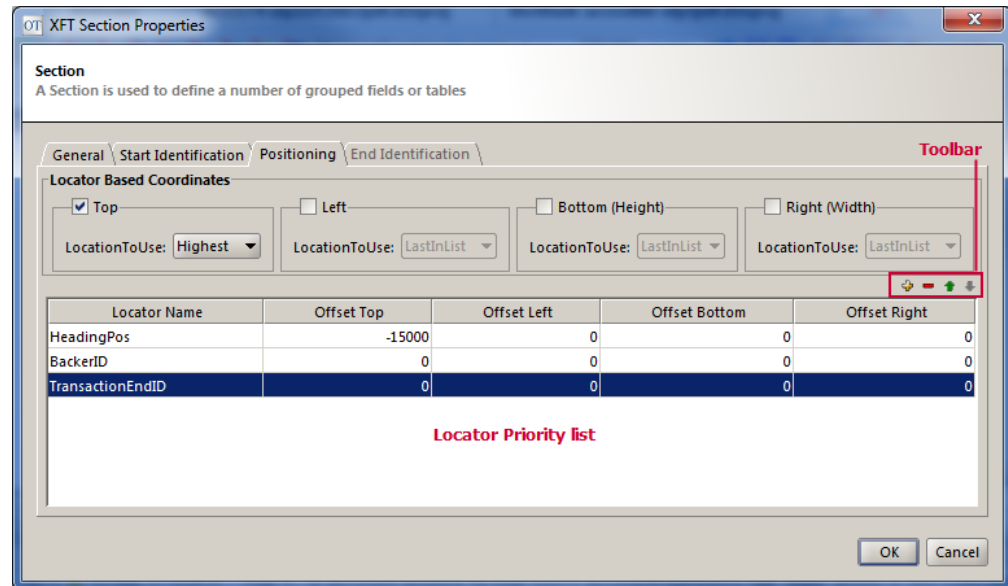


Figure 6-27: XFT Section Properties dialog showing Locator Priority list with toolbar

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.
 - **Top.** Indicates that the top side of the field's bounding box is offset by the given amount.
 - **Left.** Indicates that the left side of the field's bounding box is offset by the given amount.
 - **Right (Width).** Indicates that the right side of the field's bounding box is offset by the given amount.
 - **Bottom (Height).** Indicates that the bottom side of the field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the **LocationToUse** setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.

- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.
- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List Toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click the **Add** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the dropdown menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the field’s bounding box is offset from the found locator’s base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Fields** dialog appears. The Detect Fields tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Fields appears. For more information on working with this tool, see [“Detecting Fields” on page 363](#).

6.15.3.5.4 Section Properties End Identification Tab

The **End Identification** tab specifies when a section concludes. The end identifier does not have to be on the same page as the start identifier. It has the following fields:



Note: This tab is only active after you have selected the **Enable end identification** option on the General tab of the XFT Section Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add an end identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu.
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator.
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.

- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Fields** dialog appears. The Detect Fields tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Fields appears.

Related Links

- [“Adding a Text Locator” on page 486](#)
- [“Confidence Factor” on page 345](#)
- [“Detecting Fields” on page 363](#)

6.15.3.5.5 Section Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your section. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **SECT.** Represents a standalone section element.
 - **DIVISION.** Represents a division element, which can be either an individual or group of block-level elements.
 - **ARTICLE.** Represents any complete or self-contained content within your document. An article should still be coherent by itself if it were to be distributed independently from the rest of the document.

- **PART**. Represents a large division of a document and can also be used to group smaller segments of content together, like section, division, or article elements.
- **INDEX**. Represents an index element.
- **NONE**. Indicates that no tag should be applied to the content.
- **ARTIFACT**. Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.
- **HEADER_ARTIFACT**. Denotes a header element that should be ignored.
- **FOOTER_ARTIFACT**. Denotes a footer element that should be ignored.
- **WMARK_ARTIFACT**. Denotes a watermark element that should be ignored.
- **LAYOUT_ARTIFACT**. Denotes a layout element that should be ignored.
- **BACKGR_ARTIFACT**. Denotes a background element that should be ignored.
- **PAGE_ARTIFACT**. Denotes a page element that should be ignored.
- **Tag Name**. Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language**. Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text**. Defines the actual text for this item that is used by screen readers.
- **Tag Only Once**. Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags**. Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.
- **Logical Order**. Defines how elements within the container are ordered as they are inserted. You can choose from **Static** or **Auto** ordering. With Static ordering, the order in which the elements are defined and processed in is used when inserting elements into the DOM. Use this option when the order within your document is known to be consistent. With Auto ordering, elements are ordered based on their graphic page position. This option is useful when the exact order

of elements may vary depending on the position of the elements on the physical page.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Fields** dialog appears. The Detect Fields tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Fields appears. For more information on working with this tool, see [“Detecting Fields” on page 363](#).

6.15.3.5.6 Section Properties Auto Detect Tab

The **Auto Detect** tab specifies auto detection settings for your section. It has the following fields:



Note: This tab is only active after you have selected the **Enable Auto Detect** option on the General tab of the XFT Section Properties window.

- **Threshold Name.** Denotes the XFT threshold options to use for auto detection. You can select an existing threshold from the dropdown menu or select the ellipsis button, , to open the **Select Threshold** dialog to **Add** a new threshold. You can also **Edit** or **Remove** an existing threshold from the Select Threshold window. The **Default** threshold is selected by default and cannot be deleted. For more information, see [“Adding a Threshold” on page 462](#).
- **Auto Detect Options.** Denotes the auto detection tagging options to use on located elements. You can select an existing auto-detect options set from the dropdown menu or select the ellipsis button, , to open the **Auto Detect Options** dialog to **Add** a new options set. To configure the properties, select an existing option set and click the **Edit** button. You can also **Remove** an existing auto detect option from the Auto Detect Options window. The **Default** auto detect option is selected by default and cannot be deleted. For more information, see [“XFT Section Auto Detect Option Properties” on page 387](#).
- **Image Alternate Text.** Defines the alternate text for auto detected images in this section. Some of the options contain further fields that must be set so it is recommended you click the ellipses button, , to display the **Alternate Text** dialog where you can configure the options. If any additional fields must be configured after selecting an option from the **Type** list, the new fields are displayed below.
 - **STATIC.** Indicates that the alternate text is fixed and does not change. In the Static field, enter the alternate text you want to apply to images.
 - **FORMATTER.** Specifies the user of alternate text that is reformatted by a formatter class. The following fields must be completed:
 - **Locale.** Contains locale settings to help classify the alternate text.

- **Language.** Indicates the language code for the alternate text as defined by ISO-639 standards.
- **Country.** Indicates the country code for the alternate text as defined by ISO-3166 standards.
- **Index Range.** Contains options for incorporating the index into XFT results. If your results return multiple lines, this index points to the lines containing the text to be formatted.
 - **Start.** Designates the starting point of the index range. A negative number used as the value for the start results in a relative position from the end of the range, while a zero (0) value represents the last item in the field.
 - **End.** Designates the ending point of the index range. A positive number represents an absolute position in the range, while a negative number represents an offset from the last item. A zero (0) value represents the last item in the field.
 - **Input Mask.** Describes the format of the date as input from the original data to be used as a mask. This parameter only appears when the `FormatDate` formatter class is selected. The following characters can be used as accepted values:
 - yy = Two digit year (10)
 - yyyy = Four digit year (2010)
 - M = Single digit month (1)
 - MM = Two digit month (01)
 - MMM = Short form month name (Jan)
 - MMMM = Long form month name (January)
 - d = Single digit day of month (9)
 - dd = Two digit day of month (09)
 - E = Short form day of the week (Tue)
 - EEEE = Long form day of the week (Tuesday)For instance, a sample mask for 31/01/2010 would be represented by `dd/MM/yyyy`.
 - **Output Mask.** Describes the format of the date returned by the formatter to be used as a mask. This parameter only appears when the `FormatDate` formatter class is selected. The following characters can be used as accepted values:
 - yy = Two digit year (10)
 - yyyy = Four digit year (2010)
 - M = Single digit month (1)
 - MM = Two digit month (01)

MMM = Short form month name (Jan)

MMMM = Long form month name (January)

d = Single digit day of month (9)

dd = Two digit day of month (09)

E = Short form day of the week (Tue)

EEEE = Long form day of the week (Tuesday)

For instance, a sample mask for 31/01/2010 would be represented by dd/MM/yyyy.

- **Character to Replace.** Indicates the character(s) to replace. This parameter only appears when the ReplaceCharacter formatter class is selected.
- **Replacement Character.** Indicates the character(s) to use when replacing the character(s) specified in Character to Replace parameter. If no replacement character is specified, the input character is simply removed. This parameter only appears when the ReplaceCharacter formatter class is selected.
- **PROVIDER.** Indicates that the alternate text contains settings for specifying a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection. This is useful when reusing large chunks of static text across all documents in a large job, as it allows you to insert the section as an object, saving time and disk space in compiling the output PDFs, such as those used for customer billing. An object is more efficient to insert into a PDF, but then it is not readable by screen reader technologies. By using a TextProviderReference to instruct screen readers where the text content is located, a screen reader can read the section from the source text without impacting the structure and efficiency of the PDF.
 - **Instance Name.** References the Instance Name of an AlternateTextProvider as defined in the TextProviders collection. On the Alternate Text dialog, you can select **Browse Providers**, , to bring up the Providers dialog where you can configure your provider settings. For more information, see “[Managing Alternate Text Providers](#)” on page 321.
 - **Properties.** Controls custom properties which, if present, are passed into the providers as arguments. The properties defined here are specific to the MarkupItem associated with this ProviderInstance reference.
- **DISABLED.** Turns off alternate text for images in the output.
- **CS_ALT_TEXT_MANAGER.** Retrieves the approved alternate text used for images from the Alternate Text Manager console.
- **IMAGE_SOURCE.** Specifies that the Constant Print image’s alternate text setting in the DOM properties is used.

When auto detect is called, any tagged elements found in the section are pulled inline along with their associated DOM structure to the auto detection node in order

to maintain the flow's reading order. Any elements that are not already tagged will be added to the DOM structure with their detected types.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Fields** dialog appears. The Detect Fields tool can assist with auto-detecting the fields within your section and is a convenient way of completing a first attempt at adding fields to your new section. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Fields appears. For more information, see [“Detecting Fields” on page 363](#).

XFT Section Auto Detect Option Properties

The auto detect options for XFT Sections help you control how page elements are detected within your documents. By taking the time to fine tune these options about your sections, the auto detection tool can return better results from your document. Multiple option sets can be created to ensure an option set is readily available to fit the criteria for variations in page design of sections within the same document. The Default auto detect option set is used by default and cannot be deleted. The following parameters are available to be configured for an auto detect options set:

- **JoinUnderRatio**. Specifies the ratio of font height to the next line within the same paragraph.
- **JoinDiffFonts**. Denotes whether your section contains instances of consecutive lines with different fonts joining within the same paragraph. By default, this is disabled.
- **InTableColGap**. Defines the minimal amount of space, in 72000 DPI, between columns within a table.
- **AllTableRowsHaveNumbers**. Indicates whether all table rows possess at least one column with numeric data, which can be useful in keeping cells together until there is a cell with a numeric value. By default, this is disabled.
- **TableStartSameFont**. Indicates whether all instances of the first row in tables use the same font. By default, this is disabled.
- **TableMaxLinesToStart**. Denotes the maximum number of lines that can appear at the start of a table. If there is no maximum, enter 0.
- **BoldFontMeansH1**. Indicates whether an isolated bold line should be tagged as H1 (level one heading). By default, this is enabled.
- **TextToTag**. Contains a list of text constants and patterns to be marked with a predefined tag type. If multiple TextToTag sets are enabled concurrently, you must observe the order in which the sets appear in your list since the options from the first matching set is used.
- **IsActive**. Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.

- **TextType**. Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
- **IgnoreCase**. Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using **REGEX** setting for **TextType**, this setting is ignored.
- **MoreUpperNoPunctuation**. Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
- **Value**. Specifies the pattern or constant text to compare against.
- **TagType**. Indicates the type of tag to apply to data matching your specified value.
- **FontCompare**. Contains options for comparisons based on font names, styles, and point sizes.
 - **Fontname**. Denotes the regular expression name of the font typeface such as ***Bold** or **Arial-Bold**. If no value is provided, then the font name check is not performed.
 - **Styles**. Contains a comma separated list of acceptable font styles, represented by letters. You can enter **B** for Bold, **I** for Italic, **BI** for Bold-Italic, or **N** for Normal (neither bold or italic). For example, using the value **"B,BI"** would match fonts using the bold or bold-italic styles while ignoring normal and italic styles. If no value is provided, all styles are matched.
 - **PtSizes**. Contains a comma separated list of decimal numbers or number ranges to indicate font point sizes. For example, **"10.3,11-15"** would match fonts with a point size of 10.3 or anything between 11 and 15. If no value is provided, all point sizes are matched.
- **SingleLineOnly**. Designates whether only single lines (lines not joined into a paragraph) are processed. By default, this is disabled.
- **SegmentToCompare**. Denotes the portion of the paragraph or line to compare against. You can select from the following options:
 - **FIRST_LINE**. Indicates that only text from the first line of the paragraph is compared.
 - **FIRST_FONT**. Indicates that only text from the first font is compared when a paragraph containing multiple fonts is encountered.
 - **FULL_PARA**. Indicates that text from the whole paragraph is compared.
- **SegmentToTag**. Denotes the portion of the paragraph or line to tag. You can select from the following options:
 - **FIRST_LINE**. Indicates that the first line is tagged as indicated by the **TagType** parameter while the remainder is marked with the **P** (paragraph) tag.

- **FIRST_FONT**. Indicates that the text in the first font is tagged as indicated by the TagType parameter while the remainder is marked with the P (paragraph) tag.
 - **FULL_PARA**. Indicates that the whole paragraph is tagged as indicated by the TagType parameter.
- **FalseTableStart**. Contains a list of patterns and constant text to compare against as tables are detected. When a match is found, the table start is ignored in order to provide an override for incorrectly detected tables.
 - **IsActive**. Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType**. Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase**. Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.
 - **MoreUpperNoPunctuation**. Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value**. Specifies the pattern or constant text to compare against.
- **CombineSuperscript**. Denotes whether to join two base lines as one if the two base lines join in a manner that is likely to be superscript text. By default, this is enabled.
- **HandleVectorCelledTables**. Designates whether tables with cells that are contained within vector lines are processed with the vector lines intact. (Note that enabling this setting may affect the precision of the cell detection logic as not all vector lines are drawn in a manner that is completely compatible.)
- **ListLBLFonts**. Specifies the fonts used in list labels. Any text using these fonts following a bullet character is included with the list's LBL tag. If any of these values end with an asterisk (*), a BeginsWith test is performed.
- **BulletChars**. Contains a list of characters or strings that indicate a list is starting. There are two special values which may be used:
 - **#.** (pound period), which will match 1, 2, or 3 digit numbers followed by a period.
 - **@** (at-sign right-parentheses), which will match 1, 2, or 3 lower case letters followed by a right parentheses.
- **BreakOnObject**. Designates whether paragraph joins are stopped when a vector or image is between two lines of text. This setting may have an effect on how tables are recognized.
- **NotesReferences**. Contains options for identifying notes and references within your documents.

- **IncludeList.** Designates any patterns to include during note and reference processing.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using **REGEX** setting for **TextType**, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
 - **StyleType.** Indicates the style of text for comparison. You can choose from **SUPERSCRIPT**, **NON_SUPERSCRIPT**, and **ANY**.
- **ExcludeList.** Designates any patterns to exclude from note and reference processing.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using **REGEX** setting for **TextType**, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
 - **StyleType.** Indicates the style of text for comparison. You can choose from **SUPERSCRIPT**, **NON_SUPERSCRIPT**, and **ANY**.
- **Name.** Specifies the unique name assigned to this set of options.

6.15.3.5.7 Configuring Multiuse Locators for Sections

The component can be configured to reuse a locator multiple times to identify the start/end point of a section and the position of the top/bottom of the section. This feature is beneficial to users with documents where sections can start and stop on the same page or span multiple pages. Furthermore, the processing of documents with multiple instances of the same section on the same page is also greatly improved with the flexibility of multi-use locators.

An example of a document that multi-use locators are well-suited for is a transactions section on a customer's bank account or credit card statement since the size of the transactions section cannot be predetermined with an ending identifier because the dimensions of the section are likely to be different for each customer. Instead, this issue can be overcome by reusing a single locator to identify and/or position a section and then using the same locator to determine start/end identification or top/bottom positioning (these locators must be defined with the same name because the subsequent instance of the same locator ends the current section and also starts the next section). In situations where a successive end locator cannot be determined, such as at the end of a document, the position of the section will be determined by the bounding box that defines your section.

To enable multiple-use locators for multiple sections:

1. In the Preview pane of the XFT window, define a **section** as usual, but be sure to draw the bounding box defining your section as large as possible while excluding any content at the top or bottom of the page that does not apply to your section.

It is recommended that you make the bounding box around your sections as large as possible because the assumption by the component is to use the position of the fixed field bounding box to determine where the section ends in the absence of an end locator, hence, you want to include the maximum possible area for your section while excluding any unrelated content.

2. For your newly created section, open the XFT Section Properties dialog and set up a **top** Positioning locator, which is set as a fixed position from the bounding box. Since this locator can be used to both identify the start/end and top/bottom positioning, if you have multiple instances of the same section they should all use the same locator name.



Note: If you set a locator and top offset above the fixed position of the bounding box, then the locator and top offset settings are used instead of the bounding box's fixed position.

3. Configure any other settings you require for your section. When you are finished, click **OK** to close the dialog.
4. Repeat steps 1-3 until you have defined all the necessary sections with the XFT Field Technology component for your document.
5. **Save** your settings.

When you run your project, the component automatically recognizes the presence or absence of end locators and will use your locators and bounding boxes accordingly to determine start/end identifiers and top/bottom positions.

Advanced Configuration

The XFT Field Technology component's `ActiveObjectsMayUseTopLocator` and `EnableRepeatingSectionProcessing` parameters can also be set in order to facilitate more advanced configurations of multi-use section locators. For both parameters below, assume you have a multi-page document where sections span multiple pages and also contain some new sections starting on a page where a previous section ended.

ActiveObjectsMayUseTopLocator

When the `ActiveObjectsMayUseTopLocator` parameter is set to true (this is the default behavior), the component will use the subsequent section's top locator for positioning while assuming that it is a continuation of the previous section. However, if the following section is actually a new section and not a continuation of the previous section, then `ActiveObjectsMayUseTopLocator` should be set to false so that the component will recognize that it is a new section. Alternatively, instead of disabling this parameter you can draw a fixed section bounding box in order to define locator positioning with absolute values.

EnableRepeatingSectionProcessing

Some documents contain repeating sections on the same page, which may cause difficulty in determining the locators for where a particular section ends. The `EnableRepeatingSectionProcessing` parameter can be enabled to use the start locator for the next section as the end locator and bottom positioning coordinate for the current section. By default, this parameter is disabled.

Related Links

- [“XFT Window” on page 350](#)
- [“Section Properties Positioning Tab” on page 378](#)
- [“ActiveObjectsMayUseTopLocator”](#)
- [“EnableRepeatingSectionProcessing”](#)

6.15.3.6 Adding a Text Field

A text field is an area that contains specific items of data that you want to extract from the document.



Note: You must first have added a document and defined a section before you may add a text field.

To add a text field:

1. From the XFT window toolbar, select **Add Text Field**, .
2. When you move the cursor over the image preview area, the cursor changes from an arrow to a crosshair. With the crosshair cursor, click and drag to draw a bounding box around the area you want to define as a text field. The area that you want to mark as a text field must be within the boundaries of a previously defined section.
The **XFT Text Field Properties** dialog box appears.
3. Complete the fields in the XFT Text Field Properties dialog and when you are done, click **OK**.



Note: You can only edit one field at a time.

6.15.3.6.1 Text Field Properties General Tab

The **General** tab specifies general properties for the text field. It has the following fields:

- **Name.** Specifies a name for your text field. Text field names are allowed to contain spaces. The name must be unique within the project as it is used to reference this item in report logs.
 - **Options.** Determines which additional tabs are available; you can select from **Enable Identification Processing** or **Enable Dynamic Positioning**.
Identification processing refers to the start and end identifiers. **Dynamic positioning** refers to sections that are located based on their position relative to locators you specify, even if they shift on the page.
-
- Note:** If you are using identifiers for your document, you must first define the locators that act as identifiers.
- **Location Behavior.** Determines locator behavior for the field.
 - **Reposition on Page Break.** When selected (which is the default value), the top locator can be used for positioning when the section continues on a subsequent page. When disabled, the fixed bounding box is used for the top location.
 - **Allow Locator Duplicates For Identification** and **Allow Locator Duplicates for Positioning.** When the section repeats on the page and its end cannot

easily be determined, select these check boxes to use the start locator of the next section as the end locator and bottom positioner of the current section. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section and the start of the next section. These features are disabled by default.

- **Position and Size.** Specifies the default position and size of the section.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.
 - **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the XFT Text Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.6.2 Text Field Properties Identification Tab

The **Identification** tab contains settings that indicate when a text field begins or ends. It has the following fields:

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add an identification, click the **Add Identification**, , button and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.

- **Require Any.** Processes an item if any of the form filter parameters in your list are found.
- **Ignore All.** Ignores the page only if all form filter parameters are found.
- **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Text Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.6.3 Text Field Properties Positioning Tab

The **Positioning** tab allows you to set the properties for locating sections by basing each position of the section's bounding box on a locator as specified in the Locator Priority list. This type of dynamic positioning is helpful since sections are located based on their position relative to locators you specify, even if they move on the page.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator's search string is used as the starting point from which the section's position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.
- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section's original bounding box position.

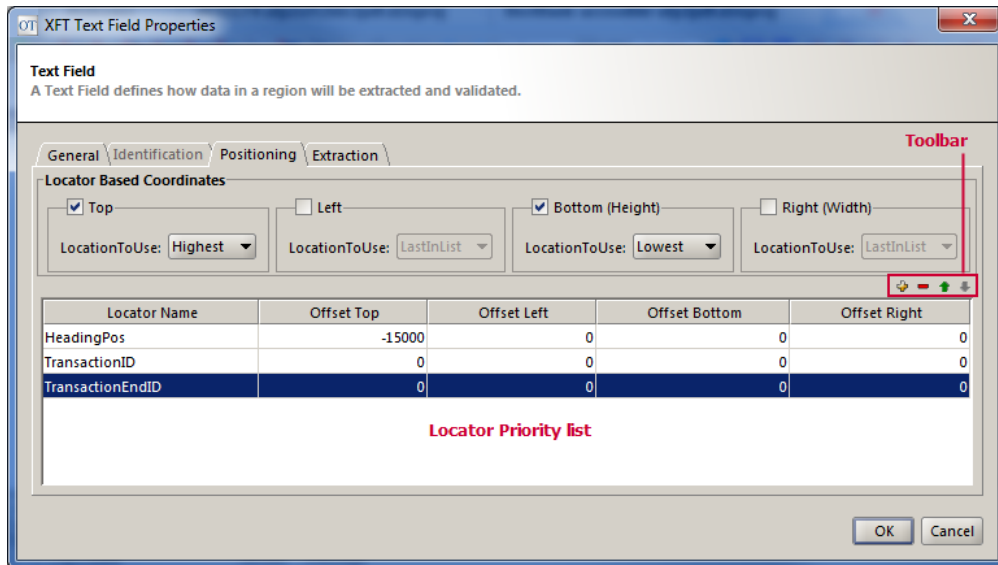


Figure 6-28: XFT Text Field Properties dialog showing Locator Priority list with toolbar

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.
 - **Left.** Indicates that the left side of the text field's bounding box is offset by the given amount.
 - **Top.** Indicates that the top side of the text field's bounding box is offset by the given amount.
 - **Right (Width).** Indicates that the right side of the text field's bounding box is offset by the given amount.
 - **Bottom (Height).** Indicates that the bottom side of the text field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the **LocationToUse** setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.

- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.
- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List Toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click the **Add** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the text field’s bounding box is offset from the found locator’s base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.6.4 Text Field Properties Extraction Tab

The **Extraction** tab specifies the font filter and extraction settings. It has the following text fields to set:

- **Text Print Direction.** Specifies the orientation in which the text prints or the degree of rotation on the page. From the drop-down menu, you can select from **Portrait** (0°), **Inverse Landscape** (90°), **Inverse Portrait** (180°), and **Landscape** (270°).
- **Thresholds Settings.** Indicates the threshold settings to use. You may opt to use the default settings or you can click the ellipsis button to display the **Select Threshold** dialog window to **Add**, **Edit**, or **Remove** a threshold you have defined. For more information, see [“Adding a PCL Soft Input Font” on page 526](#).
- **Include Form Text.** Indicates that text embedded in forms should be included in the search.
- **Font Filter.** Indicates which font filters should be used with your project.
- **Filter Type.** Specifies whether the list of the application’s font filters are included or excluded. From the list, select **Include List** or **Exclude List**.

To review or edit additional properties, click on another tab in the XFT Text Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.6.5 Text Field Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your text field. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

Basic

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **P.** Represents a paragraph or other generic block-level element.
 - **H.** Represents a heading element.
 - **H1 to H6.** Represents six levels of headings, with H1 being the highest or most important heading level and H6 indicating the lowest level.
 - **SPAN.** Represents an inline text element, but is more often used for delimiting any text coupled with some style attributes.
 - **CAPTION.** Represents a caption element, which is a segment of text describing a figure or table.
 - **NOTE.** Represents a note entry element, which is supplementary text that is referenced in the main body of content. Note elements include footnotes and endnotes, but can also consist of inline elements with special formatting.
 - **REFERENCE.** Represents a reference entry element, which is content that refers to some other content located in another part of the document. For instance, a cross reference or footnote.
 - **LBL.** Represents a label element, which can be a name, number, or bullet character used to identify and set apart an element from others in a list.
 - **LBODY.** Represents a list item body element, which is the text content of a list item.
 - **CODE.** Represents a code entry element, which is an embedded instance of software source code.
 - **QUOTE.** Represents a quoted content element, which is an inline segment of text within a paragraph or another block-level element. Similar to the BLOCKQUOTE tag, which consists of at least one paragraph as opposed to quoted content inside a paragraph for QUOTE.
 - **BIBENTRY.** Represents a bibliography entry element, which contains a summary of where citation information can be located .

- **ARTIFACT**. Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.
- **HEADER_ARTIFACT**. Denotes a header element that should be ignored.
- **FOOTER_ARTIFACT**. Denotes a footer element that should be ignored.
- **WMARK_ARTIFACT**. Denotes a watermark element that should be ignored.
- **LAYOUT_ARTIFACT**. Denotes a layout element that should be ignored.
- **BACKGR_ARTIFACT**. Denotes a background element that should be ignored.
- **PAGE_ARTIFACT**. Denotes a page element that should be ignored.
- **Tag Name**. Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language**. Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Alternate Text**. Defines the use of alternate text for this node/structural element. Alternate text is disabled by default, but you can enable it by selecting the source of the alternate text from the dropdown menu. If your selected source requires additional configuration, use the ellipsis button, , to access these properties in the Alternate Text dialog. The following choices are available:
 - **STATIC**. Indicates that the alternate text is static and does not change.
 - **FORMATTER**. Specifies the use of alternate text that is reformatted by a formatter class.
 - **PROVIDER**. Indicates a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection.
 - **DISABLED**. Indicates that alternate text is not applicable.
 - **CS_ALT_TEXT_MANAGER**. Specifies that approved alternate text from the Alternate Text Manager is to be used as the source of alternate text.
- **Actual Text**. Defines the actual text for this item that is used by screen readers.
- **Tag Only Once**. Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.

- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.
- **Detect Columns.** Indicates that columns are to be detected among text elements. By default, this is enabled.

Text Links

- **Text Expression.** Indicates the regular expression defining the text used for the hyperlink. If no value is provided, all text will be linked.
- **Type.** Indicates the type of text link to create.
- **Target.** Specifies the URL or XPATH expression, depending on your selected target type.
- **Alternate URL Description.** Specifies an alternate description to use for the link to meet accessibility requirements.
- **MFCT Color.** Denotes the color of the link. In the table on the DOM Properties tab, you can select a color from the MFCT Color list. If you do not define one, the default color is used.
- **Underline.** Designates if the link is underlined.
- **Underline Thickness.** Specifies the underline thickness in XIF points.

To review or edit additional properties, click on another tab in the XFT Text Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.7 Adding an Object Field

An object field is an area that contains specific items of data that you want to extract from or identify in the document. These objects may be images, lines, vectors, shades, or barcodes.



Note: You must first have added a document and defined a section before you may add an object field.

To add an object field:

1. From the XFT window toolbar, select **Add Object Field**, .
2. When you move the cursor over the image preview area, the cursor changes from an arrow to a crosshair. With the crosshair cursor, click and drag to draw a bounding box around the area you want to define as an object field. The area that you want to mark as an object field must be within the boundaries of a previously defined section.
The **XFT Object Field Properties** window appears.
3. Complete the fields in the XFT Object Field Properties dialog box and when you are done, click **OK**.



Note: You can only edit one field at a time.

6.15.3.7.1 Object Field Properties General Tab

The **General** tab specifies general properties for the object field. It has the following fields:

- **Name.** Specifies a name for your object field. Object field names are allowed to contain spaces. The name must be unique within the project as it is used to reference this item in report logs.
- **Options.** Determines which additional tabs are available; you can select from **Enable Identification Processing** or **Enable Dynamic Positioning**.

Identification processing refers to the start and end identifiers. **Dynamic positioning** refers to sections that are located based on their position relative to locators you specify, even if they shift on the page.



Note: If you are using identifiers for your document, you must first define the locators which act as identifiers.

- **Location Behavior.** Determines locator behavior for the field.
- **Reposition on Page Break.** When selected (which is the default value), the top locator can be used for positioning when the section continues on a subsequent page. When disabled, the fixed bounding box is used for the top location.
- **Allow Locator Duplicates For Identification** and **Allow Locator Duplicates for Positioning.** When the section repeats on the page and its end cannot easily be determined, select these check boxes to use the start locator of the next section as the end locator and bottom positioner of the current section. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section and the start of the next section. These features are disabled by default.
- **Position and Size.** Specifies the default position and size of the section.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.
 - **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the **XFT Object Field Properties** window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.7.2 Object Field Properties Identification Tab

The **Identification** tab contains settings that indicate when an object field begins or ends. It has the following fields:

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add an identification, click the **Add Identification**, , button and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Object Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.7.3 Object Field Properties Positioning Tab

The **Positioning** tab allows you to set the properties for locating sections by basing each position of the section's bounding box on a locator as specified in the Locator Priority list. This type of dynamic positioning is helpful since sections are located based on their position relative to locators you specify, even if they move on the page.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator's search string is used as the starting point from which the section's position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.
- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section's original bounding box position.

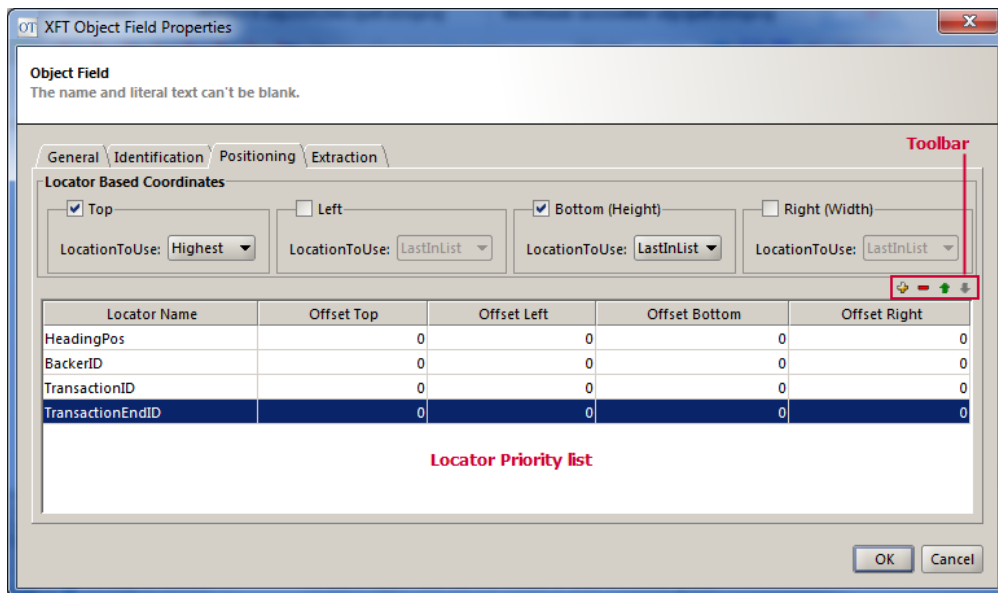


Figure 6-29: XFT Object Field Properties dialog showing Locator Priority list with toolbar

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.

- **Left.** Indicates that the left side of the object field's bounding box is offset by the given amount.
- **Top.** Indicates that the top side of the object field's bounding box is offset by the given amount.
- **Right (Width).** Indicates that the right side of the object field's bounding box is offset by the given amount.
- **Bottom (Height).** Indicates that the bottom side of the object field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the **LocationToUse** setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.
- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List Toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click the **Add** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the object field’s bounding box is offset from the found locator’s base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.7.4 Object Field Properties Extraction Tab

The **Extraction** tab specifies the font filter and extraction settings. It has the following object fields to set:

- **Include object types.** Specifies the object type(s) that will be extracted from the input PDF. You can select from one or more of these object types to extract: **Image, Line, Barcode, Shade, Vector, and Text**.
- **Include objects inside forms.** Specifies whether the application's specified object type(s) that are inside forms are included or excluded during extraction.

- **Select All.** Click this button to include all object types during extraction.

To review or edit additional properties, click on another tab in the XFT Object Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.7.5 Object Field Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your object field. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **FIGURE.** Represents a figure/image element within the document.
 - **FORMULA.** Represents content regarded as a mathematical or other scientific equation or formula.
 - **ARTIFACT.** Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.
 - **HEADER_ARTIFACT.** Denotes a header element that should be ignored.
 - **FOOTER_ARTIFACT.** Denotes a footer element that should be ignored.
 - **WMARK_ARTIFACT.** Denotes a watermark element that should be ignored.
 - **LAYOUT_ARTIFACT.** Denotes a layout element that should be ignored.
 - **BACKGR_ARTIFACT.** Denotes a background element that should be ignored.
 - **PAGE_ARTIFACT.** Denotes a page element that should be ignored.
- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the

language tag. This information overrides the document language for the scope of the node.

- **Alternate Text.** Defines the use of alternate text for this node/structural element. Alternate text is disabled by default, but you can enable it by selecting the source of the alternate text from the dropdown menu. If your selected source requires additional configuration, use the ellipsis button, , to access these properties in the Alternate Text dialog. The following choices are available:
 - **STATIC.** Indicates that the alternate text is static and does not change.
 - **FORMATTER.** Specifies the use of alternate text that is reformatted by a formatter class.
 - **PROVIDER.** Indicates a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection.
 - **DISABLED.** Indicates that alternate text is not applicable.
 - **CS_ALT_TEXT_MANAGER.** Specifies that approved alternate text from the Alternate Text Manager is to be used as the source of alternate text.
 - **IMAGE_SOURCE.** Specifies that the Constant Print image's alternate text setting in the DOM properties is used.
- **Actual Text.** Defines the actual text for this item that is used by screen readers.
- **Tag Only Once.** Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.
- **Create Multiple Tags.** Specifies that a <Figure> tag is created in the DOM tree and accessible PDF documents for each instance of an object instead of having one <Figure> tag containing multiple image objects. Each occurrence of the tag is treated separately, including the ability of assigning different alternate text to them.
- **URL.** Designates that the string provided is associated with the node as a URL.

To review or edit additional properties, click on another tab in the XFT Object Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.8 Adding a Form Field

Form fields are elements on the page that can accept some form of input from the user, such as a check box or radio button. As each type of form field has its own form of validation logic, you must indicate the field type during configuration. Within your document, a form field can be based off an image, vector, or character.

To add an XFT form field:



Note: You must first have added a document and defined a section before you may add a form field.

1. In the Design pane of the XFT window, select the section in which you want to add the form field.
2. From the XFT window toolbar, select **Add Form Field**, .
3. When you move the cursor over the image preview area, the cursor changes from an arrow to a crosshair. With the crosshair cursor, click and drag to draw a bounding box around the area you want to define as a form field. The area that you want to mark as a form field must be within the boundaries of a previously defined section.

The **XFT Form Field Properties** window appears.

4. Complete the fields in the XFT Form Field Properties dialog and when you are done, click **OK**.

6.15.3.8.1 Form Field Properties General Tab

The **General** tab specifies general properties for the form field. It has the following fields:

- **Name.** Specifies a name for your form field. Form field names are allowed to contain spaces. The name must be unique within the project as it is used to reference this item in report logs.
- **Search Interactive Objects Only.** Displays the options available on the **Interactive Extraction** tab for extracting different types of interactive objects. Be aware that enabling this tab disables the **Extraction** tab along with all of its associated options.
- **Options.** Determines which additional tabs are available; you can select from **Enable Identification Processing**, **Enable Dynamic Positioning**, **Enable Multiple Instances Per Section**, **Enable DOM Properties**.

Identification refers to the start and end identifier options. **Dynamic positioning** refers to sections that are located based on their position relative to locations you specify, even if they shift on the page.



Note: If you are using identifiers for your document, you must first define the locators that act as identifiers.

- **Field Type.** Denotes the type of form field you are configuring. You can select from **Check Box**, **Radio Button**, **Push Button**, **Input Text Field**, or **Choice Field**. When using the **Convert to Interactive** option, additional interactive object (Annotation) options appear depending on the field type selected.
- **Convert to Interactive.** Specifies whether flattened form fields are converted into interactive fields. Depending on the type of form selected for Field Type, additional configuration options appear. (Be aware that graphical appearance of interactive objects is controlled by XifOptions and XifAcroFormOptions style parameters while the Choice Field graphical style is controlled by InputTextStyle.)
 - If you selected Radio Button, for **Common Value** you must enter a string value to specify the appearance on name. Be aware that all radio buttons in a group with the same string value work in unison.
 - If you selected Push Button, for **Label** you must enter a string value that specifies a button label. If a valid button label is not used then the DOM name is used instead.
 - If you selected Input Text Field, for **Initial Text** you must enter a string value to specify the text to initialize the interactive input text field. If a valid text string is not used then the XFT text results string is used instead.
 - If you selected Choice Field, two additional panes appear:
 - In the **Available choices** pane, using  and  to add or remove items in the table, create a list of options to show users when completing a form field in a PDF viewer. The string values for the options are set in the **Choice** column while the **Pre-selected** column is used to denote an option as selected by default when opening the PDF (be aware that assigning a pre-selected default option is dependent upon whether it is supported by your chosen list style).
 - Furthermore, **Copy choices to clipboard**, , can be used to copy the contents of the table as multiline text for reuse in other fields. Conversely, the **Paste choices from Clipboard**, , is used to paste content from the clipboard to the table (note that the existing contents of the table are cleared when pasting).
 - In the **Choice style** pane, you must select how the options set in the Available choices pane are displayed in a PDF by selecting either **Combo** (similar to a list box) or **List**. When Combo is selected, you can also opt to enable **Allow Text Editing**, which lets users enter custom values in the PDF. When List is selected, you can **Enable Multiple Selections**, which allows the selection of multiple choices as pre-selected options when opening the document.
- **Density Range (%).** Denotes a check box's density as a percentage range to help detect whether a check box is selected or not. If the density value of the selected area is within the entered range, then the status of the check box is established as selected. Density is expressed as the amount of coverage black pixels have within a check box. For example, if a check box's selected indicator is a Y character and

the density of the Y is 40, then if the minimum density range is set to 35% and the maximum is set to 50%, then the check box with the Y character is detected as selected while all other check boxes with densities outside this range are marked as unselected.

- **Location Behavior.** Determines locator behavior for the field.
 - **Reposition On Page Break.** When selected (which is the default value), the top locator can be used for positioning when the section continues on a subsequent page. When disabled, the fixed bounding box is used for the top location.
 - **Allow Locator Duplicates For Identification and Allow Locator Duplicates For Positioning.** When the section repeats on the page and its end cannot easily be determined, select these check boxes to use the start locator of the next section as the end locator and bottom positioner of the current section. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section and the start of the next section. These features are disabled by default.
- **Position and Size.** Specifies the default position and size of the section.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.
 - **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the XFT Form Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.8.2 Form Field Properties Identification Tab

The **Identification** tab contains settings that indicate when a form field begins or ends. It has the following fields:

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add an identification, click the **Add Identification**, , button and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).

- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Form Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.8.3 Form Field Properties Positioning Tab

The **Positioning** tab allows you to set the properties for locating sections by basing each position of the section’s bounding box on a locator as specified in the Locator Priority list. This type of dynamic positioning is helpful since sections are located based on their position relative to locators you specify, even if they move on the page.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator’s search string is used as the starting point from which the section’s position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.
- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section’s original bounding box position.

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.
 - **Left.** Indicates that the left side of the form field’s bounding box is offset by the given amount.
 - **Top.** Indicates that the top side of the form field’s bounding box is offset by the given amount.

- **Right (Width).** Indicates that the right side of the form field's bounding box is offset by the given amount.
- **Bottom (Height).** Indicates that the bottom side of the form field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the **LocationToUse** setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.
- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List Toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above

or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click the **Add** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the text field’s bounding box is offset from the found locator’s base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT Section Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.8.4 Form Field Properties Extraction Tab

The **Extraction** tab specifies various options relating to how the data is obtained from your document. It has the following options to set:

- **Include Object Types.** Specifies the object type(s) that will be extracted within the form field area from the input PDF. You can select from one or more of these object types to capture: **Image**, **Line**, **Vector**, and **Character**.
- **Include Objects Inside Forms.** Specifies whether the specified object type(s) that are inside forms are included during extraction.
- **Include Constant Form Objects.** Specifies whether recurring form objects within your form field are included during extraction.
- **Include Form Text.** Specifies whether any text detected within the form field area is included during extraction.

- **Include Constant Form Text.** Denotes whether any recurring form text within your form field are included during extraction.

To review or edit additional properties, click on another tab in the XFT Form Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.8.5 Form Field Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your form field. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **FORM.** Represents a form entry element, which is an interactive component of PDF forms that can be filled out by a user.
 - **ARTIFACT.** Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.
 - **HEADER_ARTIFACT.** Denotes a header element that should be ignored.
 - **FOOTER_ARTIFACT.** Denotes a footer element that should be ignored.
 - **WMARK_ARTIFACT.** Denotes a watermark element that should be ignored.
 - **LAYOUT_ARTIFACT.** Denotes a layout element that should be ignored.
 - **BACKGR_ARTIFACT.** Denotes a background element that should be ignored.
 - **PAGE_ARTIFACT.** Denotes a page element that should be ignored.
- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.

- **Alternate Text.** Defines the use of alternate text for this node/structural element. Alternate text is disabled by default, but you can enable it by selecting the source of the alternate text from the dropdown menu. If your selected source requires additional configuration, use the ellipsis button, , to access these properties in the Alternate Text dialog. The following choices are available:
 - **STATIC.** Indicates that the alternate text is static and does not change.
 - **FORMATTER.** Specifies the use of alternate text that is reformatted by a formatter class.
 - **PROVIDER.** Indicates a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection.
 - **DISABLED.** Indicates that alternate text is not applicable.
 - **CS_ALT_TEXT_MANAGER.** Specifies that approved alternate text from the Alternate Text Manager is to be used as the source of alternate text.
- **Actual Text.** Defines the actual text for this item that is used by screen readers.
- **Tag Only Once.** Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags.
- **Alternate Name.** Defines an alternate name to be used to identify this field.

To review or edit additional properties, click on another tab in the XFT Form Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.8.6 Form Field Properties Interactive Extraction tab

The **Interactive Extraction** tab specifies the types of interactive form fields to include during the data extraction process from your document. You have the option of capturing content from the following object types during extraction: **Check Box**, **Radio Button**, **Push Button**, **Input Text Field**, **Choice Field** (like a drop-down list), or **Signature Field**.

To review or edit additional properties, click on another tab in the XFT Form Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9 Adding a List Field

To add an XFT list field:



Note: You must have first defined your XFT document settings before you can add a list field.

1. From the XFT Window Toolbar, select **Add List Field**, .
2. When you move the cursor over the image preview area, the cursor changes from an arrow to a crosshair. With the crosshair cursor, click and drag to draw a bounding box around the area you want to define as a list field.
The **XFT List Field Properties** window appears.
3. Complete the fields in the **XFT List Field Properties** and when you are done, click **OK**.

6.15.3.9.1 Tagging List Field Items and Accessible PDFs

Accessible PDFs are PDF format files that contain readable text or tags (markup data), which direct screen reader applications logically through a document. Most documents can be read by screen reader applications without issue, but tables and multi-column text present an obstacle as it is difficult for these applications to discern a logical reading order. By creating list field tags for these types of elements, you can arrange a reading order for the screen reader applications to easily present to users.

For more information on creating Accessible PDFs, see [“Accessible PDF Generator” on page 174](#) and [“Configuring the Page Markup Component” on page 295](#) as these components must be used in conjunction to create a final accessible PDF.

6.15.3.9.2 List Field Properties General Tab

The General tab specifies general properties for the list field. It has the following fields:

- **Name.** Specifies a name for your list field. List field names are allowed to contain spaces. The name must be unique within the project as it is used to reference this item in report logs.
- **Options.** Determines which additional tabs are available; you can select from **Enable Start Identification**, **Enable Dynamic Positioning**, and **Enable End Identification**.

Identification refers to the start and end identifiers options. **Dynamic positioning** refers to sections that are located based on their position relative to locations you specify, even if they shift on the page.



Note: If you are using identifiers for your document, you must first define the locators that act as identifiers.

- **Location Behavior.** Determines locator behavior for the field.

- **Reposition on Page Break.** When selected (which is the default value), the top locator can be used for positioning when the section continues on a subsequent page. When disabled, the fixed bounding box is used for the top location.
- **Allow Locator Duplicates For Identification** and **Allow Locator Duplicates for Positioning.** When the section repeats on the page and its end cannot easily be determined, select these check boxes to use the start locator of the next section as the end locator and bottom positioner of the current section. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section and the start of the next section. These features are disabled by default.
- **Position and Size.** Specifies the default position and size of the section.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.
 - **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the XFT List Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9.3 List Field Properties Start Identification Tab

The **Start Identification** tab specifies when a new list field begins. It has the following fields:



Note: This tab is only active after you have selected the **Enable Start Identification** option on the General tab of the XFT List Field Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add a start identification, click the Add Identification button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.

- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require all.** Processes items only if all form filter parameters are found.
 - **Require any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore all.** Ignores the page only if all form filter parameters are found.
 - **Ignore any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT List Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9.4 List Field Properties Positioning Tab

The Positioning tab allows you to set the properties for locating sections by basing each position of the section’s bounding box on a locator as specified in the Locator Priority list. This type of dynamic positioning is helpful since sections are located based on their position relative to locators you specify, even if they move on the page.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator’s search string is used as the starting point from which the section’s position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.
- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section’s original bounding box position.

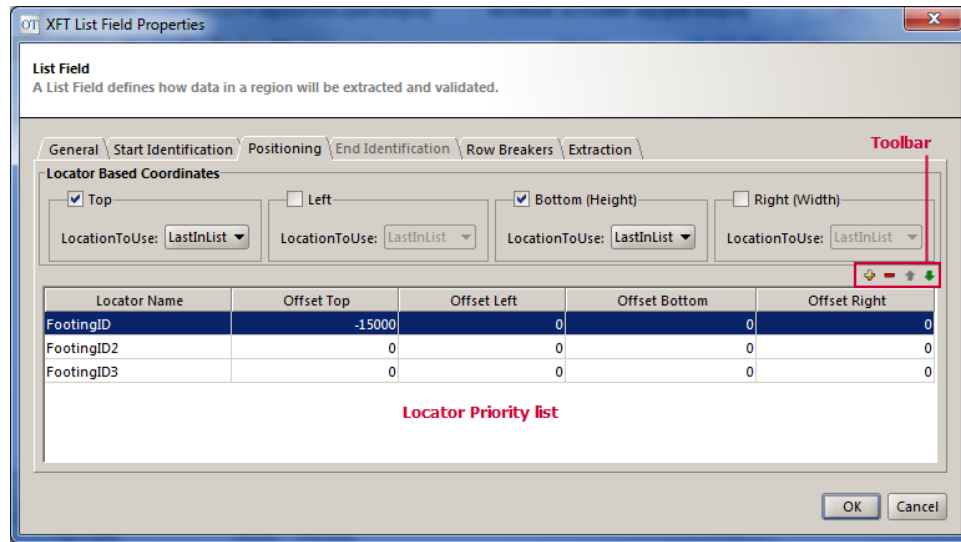


Figure 6-30: XFT List Field Properties dialog showing Locator Priority list with toolbar

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.
 - **Left.** Indicates that the left side of the field's bounding box is offset by the given amount.
 - **Top.** Indicates that the top side of the field's bounding box is offset by the given amount.
 - **Right (Width).** Indicates that the right side of the field's bounding box is offset by the given amount.
 - **Bottom (Height).** Indicates that the bottom side of the field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the **LocationToUse** setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.

- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click the Add button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the dropdown menu. For more information, see [“Adding a Text Locator” on page 486](#).

- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the field's bounding box is offset from the found locator's base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT List Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9.5 List Field Properties End Identification Tab

The **End Identification** tab specifies when a section concludes. The end identifier does not have to be on the same page as the start identifier. It has the following fields:



Note: This tab is only active after you have selected the **Enable end identification** option on the General tab of the XFT List Field Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the dropdown menu in the Locator Name column of the Identifiers table.

To add an end identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the dropdown menu.
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Form Filters.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.

- **Ignore All.** Ignores the page only if all form filter parameters are found.
- **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT List Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9.6 List Field Properties Row Breakers Tab

The **Row Breakers** tab defines how list fields are created when a list or table is processed.

List processing is performed by breaking up content blocks into rows. When multiple row breakers are provided, the results from the first row breaker is provided to the next row breaker to allow the result to be broken down into smaller rows. The final resulting rows will represent the individual items in the list. The List Style option provides support for complex nested lists. The Table Style option provides the same set of row breakers, which can be applied to XFT Table fields.

These row breakers are most useful in single level lists (no sub-lists) or when graphics help separate the list items.

To add a row breaker parameter:

1. On the Row Breakers tab, in the **Style** list, select the type of processing to perform. You can choose from **List Style** or **Table Style**.

2. Click **Add**, .

Depending on the row breaker style selected, different parameters are added.

If you selected List Style, then XftCompoundRowbreakerParm appears. Proceed to step 4.

If you selected Table Style, then XftSimpleRowbreakerParm appears. Proceed to the next step.

3. Double-click the XftSimpleRowbreakerParm row breaker to activate the dropdown menu and select a row breaker type parameter to use. Besides the default XftSimpleRowbreakerParm, you can also select one of the following row breakers:

- XftGraphicRowbreakerParm
- XftWhitespaceRowbreakerParm
- XftCustomRowbreakerParm
- XftCharacterRowbreakerParm
- XftIndentRowbreakerParm
- XftFontChangeRowbreakerParm

For more information on each row breaker type, see their respective sections in this topic.

4. Configure the parameters for the row breaker. The set of properties to configure varies depending on the type of row breaker selected.
5. When you are finished, click **OK** to save your changes.

List Style

XftCompoundRowBreakerParm parameter

Creates rows based on one or more conditions for each list level, where the level indicates the sublist hierarchy. The following properties can be set:

- **Line Break Character.** Specifies a string of one or more characters that is used to detect a new row.
 - **Regular Expression.** Indicates whether the Line Break Character value is a regular expression. If not selected, the value is interpreted as a literal string.
- **Indent.** Indicates that new rows are created when an indent is detected on the next line of text.
 - **Minimum Indent.** Specifies the minimum number of XIF units (72000 DPI) that must be detected to trigger the creation of a new row. This value is relative to the position of text found by the parent row breaker. If this is the first row breaker defined, the position is relative to the list's leftmost boundary. A positive indent value means the shift must be detected in the left direction, while a negative value indicates the shift must be to the right.
 - **Maximum Indent.** Specifies the maximum number of XIF units (72000 DPI) that must be detected to trigger the creation of a new row. This value is relative to the position of text found by the parent row breaker. If this is the first row breaker defined, the position is relative to the list's leftmost boundary. A positive indent value means the shift must be detected in the left direction, while a negative value indicates the shift must be to the right.
- **Font Changed.** Designates that new rows are created when a font change is detected at the beginning of the row.
- **Line Spacing.** Designates that new rows are created when the specified spacing amount is detected between lines.
 - **Minimum Line Spacing.** Denotes the minimum number of XIF units (72000 DPI) that is required to trigger the creation of a new row. The spacing area is measured as the distance between the bottom of the text area in the line above with the top of the text area in the line below.
- **List Level.** Designates that the list level is used to configure nested lists. When a new row is created for a higher list level, the new row becomes a nested sublist.
- **Enabled.** Indicates whether this row breaker is enabled. When building the configuration for multiple row breakers, it is sometimes useful to disable some row breakers temporarily for testing purposes.

Table Style

XftSimpleRowbreakerParm parameter

The **XftSimpleRowbreakerParm** creates a row based on the specified row height, measured in DPI from the top of the defined table's resultant rectangle. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Row Height.** Specifies the static height of one row. The default value is 10000, which is the height of a 10 point font. The minimum height is 1000, equivalent to the height of one point (or 1/72 inch).
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

XftGraphicRowbreakerParm parameter

The **XftGraphicRowbreakerParm** creates a row based on the presence of graphic elements in the data. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Graphic Type.** Use the dropdown list to select the type of graphic used to identify rows. Set to **LINES** if the input contains XifLineElements. Set to **VECTORS** if the input contains XifVectorElements.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.
- **TableRectIsFirstRow.** Select this check box to use the top horizontal line of the table as the start of the first row. This option should be set to **True** if the first text line of the table does not have a graphic above it.
- **TableRectIsLastRow.** Select this check box to use the bottom of the table rectangle as the end of the last row. This option should be set to **True** if the last text line of the table does not have a graphic below it.

XftWhitespaceRowbreakerParm parameter

The **XftWhitespaceRowbreakerParm** creates a row based on the size of the white space between each line of text. If this white space reaches a maximum size, a new row is created. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Max White Space.** Specifies the maximum white space permitted before a new row is created. The default value is 1000, equivalent to the height of one point (or 1/72 inch).
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.
- Select the **Break On Every Line** check box to break when there is a change in the vertical base line regardless of white space.

XftCustomRowbreakerParm parameter

The **XftCustomRowbreakerParm** is a custom-written row breaker which must extend the `com.xenos.d2e.transform.common.xft.IXftTableRowBreaker` class. See *Javadoc* for more information. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **ClassName.** Specifies the fully qualified path to the custom row breaker class which extends the `com.xenos.d2e.transform.common.xft.IXftTableRowBreaker` class.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

XftCharacterRowbreakerParm parameter

The **XftCharacterRowbreakerParm** creates a row based on the location of a predefined character. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **LineBreak Character.** Specifies the string or regular expression used to establish the location of a new row. If you are entering a regular expression, be sure to also select the **Regular Expression** check box otherwise the value is processed as a string.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.
- **Regular Expression.** Select this check box to enable the use of regular expressions in the **Line Break Character** field.

XftIndentRowbreakerParm parameter

XftIndentRowbreakerParm creates a row based on the position of indented text. When paired with **XftCharacterRowbreakerParm**, it can be used to detect the positioning of nested lists within tables. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Minimum Indent.** Denotes the indentation value in DPI, which is used to identify a nested list or beginning of a new row. This value is calculated from the left edge of the first line of text in a list.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

6.15.3.9.7 Detecting Nested Lists

To identify nested lists, you must use `XftIndentRowbreakerParm` in conjunction with `XftCharacterRowbreakerParm`. See the use case below for a sample.

Assume your document contains the following list:

- List A, item 1
 - Sub list item 1.1
 - Sub list item 1.2
- List A, item 2
 - Sub list item 2.1
 - Sub list item 2.2
- List B, item 3
 - Sub list item 3.1
 - Sub list item 3.2
- List C, item 4

In order to be able to detect the nested list items, you must add these separate row breakers:

1. A character row breaker matching the string value “•”.
2. An indent row breaker with the Minimum Indent value set to detect the nested list items.
3. Another character row breaker matching the string value “o” to identify the nested list item.

6.15.3.9.8 List Field Properties Extraction Tab

The **Extraction** tab contains options to specify the font filter and extraction settings for your list field. It has the following values:

- **Text Print Direction.** Specifies the orientation in which the text prints or the degree of rotation on the page. From the drop-down menu, you can select from **Portrait** (0°), **Inverse Landscape** (90°), **Inverse Portrait** (180°), and **Landscape** (270°).
- **Include Form Text.** Indicates that text embedded in forms should be included in the search.
- **Font Filter.** Indicates which font filters should be used with your project.
- **Filter Type.** Specifies whether the list of the application’s font filters are included or excluded. From the list, select **Include List** or **Exclude List**.

To review or edit additional properties, click on another tab in the XFT Table Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9.9 List Field Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your list field. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

Basic

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **L.** Represents a list element. This tag type is used to signify the start of a list structure; direct child elements should use the list item (LI) tag type to enclose individual content items within the list.
 - **ARTIFACT.** Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.
 - **HEADER_ARTIFACT.** Denotes a header element that should be ignored.
 - **FOOTER_ARTIFACT.** Denotes a footer element that should be ignored.
 - **WMARK_ARTIFACT.** Denotes a watermark element that should be ignored.
 - **LAYOUT_ARTIFACT.** Denotes a layout element that should be ignored.
 - **BACKGR_ARTIFACT.** Denotes a background element that should be ignored.
 - **PAGE_ARTIFACT.** Denotes a page element that should be ignored.
- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.

- **Actual Text.** Defines the actual text for this item that is used by screen readers.
- **Tag Only Once.** Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags.

Text Links

- **Text Expression.** Indicates the regular expression defining the text used for the hyperlink. If no value is provided, all text will be linked.
- **Type.** Indicates the type of text link to create.
- **Target.** Specifies the URL or XPATH expression, depending on your selected target type.
- **Alternate URL Description.** Specifies an alternate description to use for the link to meet accessibility requirements.
- **MFCT Color.** Denotes the color of the link. In the table on the DOM Properties tab, you can select a color from the MFCT Color list. If you do not define one, the default color is used.
- **Underline.** Designates if the link is underlined.
- **Underline Thickness.** Specifies the underline thickness in XIF points.

To review or edit additional properties, click on another tab in the XFT List Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.9.10 Configuring Multiuse Locators for List Fields

The component can be configured to reuse a locator multiple times to identify the start/end point of a list field and the position of the top/bottom of the list. This feature is beneficial to users with documents where lists can start and stop on the same page or span multiple pages. Furthermore, the processing of documents with multiple instances of the same list on the same page is also greatly improved with the flexibility of multi-use locators.

An example of a document that multi-use locators are well-suited for is a list of transactions on a customer's bank account or credit card statement since the size of the transactions list cannot be predetermined with an ending identifier because the dimensions of the list are likely to be different for each customer. Instead, this issue can be overcome by reusing a single locator to identify and/or position a list and then using the same locator to determine start/end identification or top/bottom positioning (these locators must be defined with the same name because the

subsequent instance of the same locator ends the current list and also starts the next list). In situations where a successive end locator cannot be determined, such as at the end of a document, the position of the list will be determined in relation to the bottom of the page.

To enable multiple-use locators for multiple lists:

1. In the Preview pane of the XFT window define a **list** as usual, but be sure to draw the bounding box defining your list as large as possible while excluding any content at the top or bottom of the page that does not apply to your list.

It is recommended that you make the bounding box around your lists as large as possible because the assumption by the component is to use the bottom of the page to determine where the list ends in the absence of an end locator, hence, you want to include the maximum possible area for your list while excluding any unrelated content.
2. For your newly created list, open the XFT List Field Properties dialog and set up a **top** Positioning locator, which is set as a fixed position from the bounding box. Since this locator can be used to both identify the start/end and top/bottom positioning, if you have multiple instances of the same list they should all use the same locator name.



Note: If you set a locator and top offset above the fixed position of the bounding box, then the locator and top offset settings are used instead of the bounding box's fixed position.

3. Configure any other settings you require for your list. When you are finished, click **OK** to close the dialog.
4. Repeat steps 1-3 until you have defined all the necessary lists with the XFT Field Technology component for your document.

When you are finished defining all your lists, continue to the next step.

5. **Save** your settings.

When you run your project, the XFT Field Technology component automatically recognizes the presence or absence of end locators and will use your locators and page dimensions accordingly to determine start/end identifiers and top/bottom positions.

Related Links

- [“XFT Window” on page 350](#)
- [“List Field Properties Positioning Tab” on page 419](#)

6.15.3.10 Adding a Table

Defining a table in your document specifies how the data in the table is extracted and validated. Tables establish relationships between the data in your document. Tables can continue across pages.



Note: Before adding a table, you must create a section as the table must exist within the section's boundaries.

To add a table:

1. In the **Design** pane, select the section where you want to create the table.
2. From the XFT window toolbar, navigate to **Add New> Add Table**, .
3. When you move the cursor over the Preview pane, the cursor changes from an arrow to a crosshair. Using the crosshair cursor, draw a bounding box around the area you want to define as a table within the boundaries of the selected section.
4. Complete the fields in the **XFT Table Field Properties** dialog and when you are done, click **OK**.

Use the available tabs to establish some universal properties regarding your table. When you are finished making your selections, click **OK** and then the **Auto Detect Table Columns** dialog box appears. The **Detect Columns** tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog box for Detect Columns appears.

If you opt to not use the tool, the table is created and shown in the Design pane, but you will have to manually add and define the columns yourself. However, the Detect Columns tool can also be used on existing tables by right-clicking the table's entry in the tree on the Design pane and from the context menu that appears, selecting **Detect Table Columns**.

For more information on working with this auto detect tool, see [“Detecting Table Columns” on page 367](#).

6.15.3.10.1 Table Properties General Tab

The **General** tab specifies general properties for the table. It has the following fields:

- **Name.** Specifies a name for your table. The name must be unique within the project as it is used to reference this item in report logs.
- **Options.** Determines which additional tabs are available; you can select from **Enable Identification** or **Enable Dynamic Positioning**.

Identification refers to the start and end identifiers. **Dynamic positioning** refers to sections that are located based on their position relative to locators you specify, even if they shift on the page.



Note: If you are using identifiers for your document, you must first define the locators which act as identifiers.

- **Location Behavior.** Determines locator behavior for the table.
- **Reposition on Page Break.** When selected (which is the default value), the top locator can be used for positioning when the table continues on a subsequent page. When disabled, the fixed bounding box is used for the top location.
- **Allow Locator Duplicates For Identification and Allow Locator Duplicates for Positioning.** When the table repeats on the page and its end cannot easily be determined, select these check boxes to use the start locator of the next table as the end locator and bottom positioner of the current table. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous table and the start of the next table. These features are disabled by default.
- **Position and Size.** Specifies the default position and size of the table.
 - **Left.** Specifies the distance of the table from the left margin.
 - **Top.** Specifies the distance of the table from the top margin.
 - **Width.** Specifies the width of the table.
 - **Height.** Specifies the height of the table.

To review or edit additional properties, click on another tab in the XFT Table Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Columns** dialog appears. The Detect Table Columns tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Columns appears. For more information on working with this tool, see [“Detecting Table Columns” on page 367](#).

6.15.3.10.2 Table Properties Start Identification Tab

The **Start Identification** tab specifies when a new table begins. It has the following fields:



Note: This tab is only active after you have selected the **Enable Start Identification** option on the General tab of the XFT Table Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the **Locator Name** column of the Identifiers table.

To add a start identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Table Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Columns** dialog appears. The Detect Table Columns tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Columns appears. For more information on working with this tool, see [“Detecting Table Columns” on page 367](#).

6.15.3.10.3 Table Properties Positioning Tab

The **Positioning** tab allows you to set the properties for locating sections by basing each position of the section’s bounding box on a locator as specified in the Locator Priority list. This type of dynamic positioning is helpful since sections are located based on their position relative to locators you specify, even if they move on the page.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator’s search string is used as the starting point from which the section’s position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.

- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section's original bounding box position.

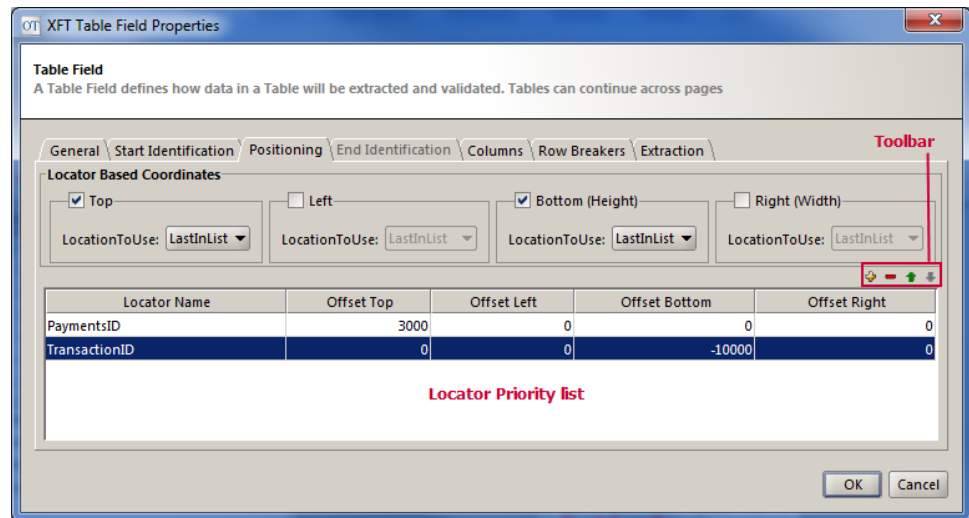


Figure 6-31: Locator Priority List with toolbar

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.
 - **Left.** Indicates that the left side of the field's bounding box is offset by the given amount.
 - **Top.** Indicates that the top side of the field's bounding box is offset by the given amount.
 - **Right (Width).** Indicates that the right side of the field's bounding box is offset by the given amount.
 - **Bottom (Height).** Indicates that the bottom side of the field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the **LocationToUse** setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following **LocationToUse** options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.

- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.
- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click the **Add** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the drop-down menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the field’s bounding box is offset from the found locator’s base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT Table Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Columns** dialog appears. The Detect Table Columns tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Columns appears. For more information, see [“Confidence Factor” on page 345](#).

6.15.3.10.4 Table Properties End Identification Tab

The **End Identification** tab specifies when a section concludes. The end identifier does not have to be on the same page as the start identifier. It has the following fields:



Note: This tab is only active after you have selected the **Enable End Identification** option on the General tab of the XFT Section Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator. The identifier values are selected from the drop-down menu in the Locator Name column of the Identifiers table.

To add an end identification, click the **Add Identification** button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list of locators you have specified in the dropdown menu. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.

- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Table Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Columns** dialog appears. The Detect Table Columns tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Columns appears. For more information on working with this tool, see “[Detecting Table Columns](#)” on page 367.

6.15.3.10.5 Table Properties Columns Tab

The **Columns** tab defines how each column’s data will be extracted and validated. It has the following fields to set:

- **Repeated Column Header.** Indicates that column headings are repeated across pages.
- **Repeated Column Positions.** Indicates that column positions remain static across pages.
- **Column Header Height.** Specifies the height of the header column, if applicable.
- **Column List:**
 - **Column Name.** Indicates the name for each column entered.
 - **Left.** Specifies the position of the column’s left edge.
 - **Width.** Indicates the column width.

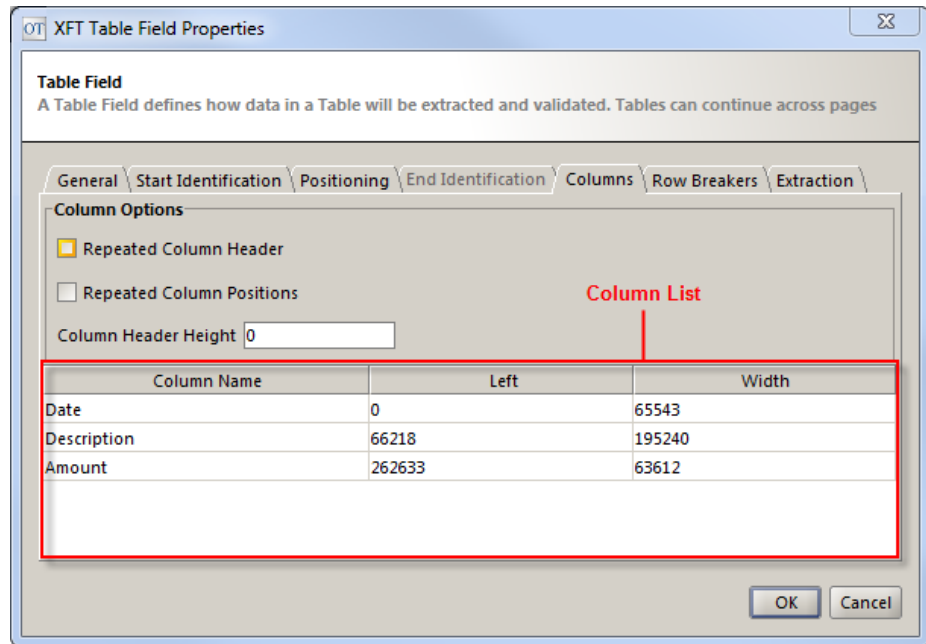


Figure 6-32: Column List area on Columns tab

To review or edit additional properties, click on another tab in the XFT Table Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Columns** dialog appears. The Detect Table Columns tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Columns appears. For more information on working with this tool, see [“Detecting Table Columns” on page 367](#).

6.15.3.10.6 Table Properties Rows Tab

The **Rows** tab defines how XFT table rows are created when a table is processed.

To add a rows parameter:

1. Click **Add**, .

The default parameter, **XftSimpleRowbreakerParm**, appears in the **Row Breaker** list.
2. Double-click on the row breaker to activate the dropdown menu and select which row breaker parameter to use. You can select one of the following row breakers:
 - [“XftSimpleRowbreakerParm” on page 440](#)
 - [“XftGraphicRowbreakerParm” on page 441](#)
 - [“XftWhitespaceRowbreakerParm” on page 442](#)

- “XftCustomRowbreakerParm” on page 442
 - “XftCharacterRowbreakerParm” on page 443
 - “XftIndentRowbreakerParm” on page 443
3. Enter the required information for the row breaker. The set of properties to configure will vary depending on the row breaker selected. See below for a complete description of each parameter.
 4. Click **OK** to save your changes.

Empty column headers and cells in tables

You can add invisible text strings to empty table data and header cells to provide improved screen reader support for tables that do not contain the ideal table cell structures for accessibility. The following options are available on the Rows tab:

- **Header Row.**
 - **Create Header Row.** Designates the creation of an empty table header row at the top of the table that can be populated with hidden text using the EmptyCellOptions parameters. The text string for the header is used in each column. This option is useful for tables that have DOM enabled, but do not contain a table header row.
- **Empty Cell.** Specifies options for adding an invisible text string to empty cells.
 - **Enable.** Indicates that adding text to empty cells is enabled.
 - **Text to Add.** Specifies the text string to add to the empty data cells, which can be announced by a screen reader.

If you want to add text to column headers and apply different text to each column, you can use the EmptyCellOptions parameters in the property settings for an XFT **Column**.

XftSimpleRowbreakerParm

The **XftSimpleRowbreakerParm** creates a row based on the specified row height, measured in DPI from the top of the defined table's resultant rectangle. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Row Height.** Specifies the static height of one row. The default value is 10000, which is the height of a 10 point font. The minimum height is 1000, equivalent to the height of one point (or 1/72 inch).
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

XftGraphicRowbreakerParm

The **XftGraphicRowbreakerParm** creates a row based on the presence of graphic elements in the data. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Graphic Type.** Use the dropdown list to select the type of graphic used to identify rows within a table You can choose from:
 - **LINES.** Indicates that the input contains XifLineElements.
 - **VECTORS.** Indicates that the input contains XifVectorElements, with each vector treated as an individual line.
 - **VECTOREDGEs.** Indicates that the input contains XifVectorElements, with the top and bottom of the vector area treated as lines.
 - **IMAGES.** Indicates that the input contains images or rasterized text.

When row breaking tables containing shaded rows, it is important to determine whether the VECTORS or VECTOREDGEs graphic type is better suited to your project requirements. The VECTORS type treats each vector as a single thick line, whereas VECTOREDGEs treats a vector as two individual lines; a top edge and a bottom edge (for rotated pages, this translates to the left and right edges of a vector). As an example, see how row breaking the following sample table with the two graphic types affects the results:

ABC10001
ABC20002
ABC30003
ABC40004
ABC50005

Figure 6-33: Sample table to help illustrate difference between VECTORS and VECTOREDGEs graphic row breaker types

If you use **VECTORS**, two rows (ABC20002 and ABC40004) are identified within the table, which are enclosed by three vector lines (ABC10001, ABC30003, and ABC50005). If you use **VECTOREDGE**s, five rows (ABC10001, ABC20002, ABC30003, ABC40004, and ABC50005) are recognized within the table since both the top and bottom edge of each vector is located.

- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.
- **TableRectIsFirstRow**. Select this check box to use the top horizontal line of the table as the start of the first row. This option should be set to **True** if the first text line of the table does not have a graphic above it.
- **TableRectIsLastRow**. Select this check box to use the bottom of the table rectangle as the end of the last row. This option should be set to **True** if the last text line of the table does not have a graphic below it.

XftWhitespaceRowbreakerParm

The **XftWhitespaceRowbreakerParm** creates a row based on the size of the white space between each line of text. If this white space reaches a maximum size, a new row is created. The following properties can be set:

- **Column Range**. Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From**. The initial column to include, where 1 is the first column in the table.
 - **To**. The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Max White Space**. Specifies the maximum white space permitted before a new row is created. The default value is 1000, equivalent to the height of one point (or 1/72 inch).
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.
- Select the **Break On Every Line** check box to break when there is a change in the vertical base line regardless of white space.

XftCustomRowbreakerParm

The **XftCustomRowbreakerParm** is a custom-written row breaker which must extend the `com.xenos.d2e.transform.common.xft.IXftTableRowBreaker` class. See *Javadoc* for more information. The following properties can be set:

- **Column Range**. Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:

- **From.** The initial column to include, where 1 is the first column in the table.
- **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **ClassName.** Specifies the fully qualified path to the custom row breaker class which extends the `com.xenos.d2e.transform.common.xft.IXftTableRowBreaker` class.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

XftCharacterRowbreakerParm

The **XftCharacterRowbreakerParm** creates a row based on the location of a predefined character. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Line Break Character.** Specifies the string or regular expression used to establish the location of a new row. If you are entering a regular expression, be sure to also select the **Regular Expression** check box otherwise the value is processed as a string.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.
- **Regular Expression.** Select this check box to enable the use of regular expressions in the **Line Break Character** field.

XftIndentRowbreakerParm

XftIndentRowbreakerParm creates a row based on the position of indented text. When paired with **XftCharacterRowbreakerParm**, it can be used to detect the positioning of nested lists within tables. The following properties can be set:

- **Column Range.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Specify the range by entering the following values:
 - **From.** The initial column to include, where 1 is the first column in the table.
 - **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- **Minimum Indent.** Denotes the indentation value in DPI, which is used to identify a nested list or beginning of a new row. This value is calculated from the left edge of the first line of text in a list.
- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

Detecting Nested Lists

To identify nested lists, you must use `XftIndentRowbreakerParm` in conjunction with `XftCharacterRowbreakerParm`. See the use case below for a sample.

Assume your document contains the following list:

- List A, item 1
 - Sub list item 1.1
 - Sub list item 1.2
- List A, item 2
 - Sub list item 2.1
 - Sub list item 2.2
- List B, item 3
 - Sub list item 3.1
 - Sub list item 3.2
- List C, item 4

In order to be able to detect the nested list items, you must add these separate row breakers:

1. A character row breaker matching the string value “•”.
2. An indent row breaker with the Minimum Indent value set to detect the nested list items.
3. Another character row breaker matching the string value “o” to identify the nested list item.

XftFontChangeRowbreakerParm

The `XftFontChangeRowbreakerParm` creates a row based on a font change, such as a change in typeface, point size, or other font attribute. The following properties can be set:

- **ColumnRange.** Indicates the range of columns whose combined width will constrain which areas can be used to break rows. Note that only the first character in each row across the specified range of columns will be inspected for font changes. Specify the column range by entering the following values:

- **From.** The initial column to include, where 1 is the first column in the table.
- **To.** The last column to include, where 0 is the last column in the table.

The default settings of From=1 and To=0 collectively specify all columns.

- Select the **Enabled** check box to enable this row breaker parameter. If not selected, this row breaker will be ignored.

6.15.3.10.7 Table Properties Cells tab

The **Cells** tab facilitates defining accessibility tags for table row headers, column headers, and table header cells in complex table layouts.

Before you can configure the cell values, you must specify the type of heading cells in the table. You can choose from either:

- **Static cells.** Denotes headers that contain persistent content and remains in a constant location in the table.
- **Dynamic cells.** Denotes headers that do not have a constant location in the table, contain content that can change, or have complex characteristics like headings within subtables, or cells that span multiple rows or columns.



Note: Table header cell parameters are processed in the order in which they are defined in the **Cell Option Name** list. You can change the processing order by moving an instance of cell options up or down in the list.

Static cells

For static cells you can set the **ColumnHeaderRow**, which defines the rows used as column headers, or the **RowHeaderColumn**, which defines the columns used as a row header. For both types of static cells, **1** represents the first row or column of a table. If the table has no column or row headers, enter **0** as the value.

Dynamic cells

For dynamic cells, you can define cell options to indicate locator areas and relative positions for the column or row headings or use text-based locators.

When adding a new set of dynamic cell options for a column or row header, you can configure the following options:

- **Name.** Identifies the instance of dynamic cell options.
- **Enabled.** Denotes the use of these options to locate a header cell using dynamic locators within a complex table structure.
- **Cell Type.** Indicates the type of header. You can choose from **COLUMN_HEADER**, **ROW_HEADER**, **BOTH** (a cell that is both a column and row header), **SUBTABLE_HEADER**, or **DATA**.
- **Dynamic Locator Search Area.** Defines a range where the header cells are located. You must specify:

- **Row from** and **to**. Specifies the start and end row numbers.
- **Column from** and **to**. Specifies the start and end column numbers.

If you want to include rows/columns from the starting row/column to the end of the table, you can use **0** in the respective **to** box to indicate the last row/column as the range's end point.

- **Starting Cell Position Relative to Locator**. Defines where the table row or column cell starts relative to the found locator's position. You must specify both a **Row offset** and **Column offset**.

Using a value of **0** denotes that the relative location is in the same row or column number as the found locator. Setting the value to **-1** denotes that the relative location starts at the found locator's position decreased by 1 while **+1** denotes that the relative location starts at the found locator's position increased by 1.

- **Cell Span**. Specifies the span width for the row and column header cells. You must specify a value for both **Row span** and **Column span**.
- **Locator**. Specifies options for identifying table headers using a text-based locator.

- **Enabled**. Denotes the use of these options to locate a header cell using a text string.
- **Indentation (72K DPI)**. Indicates an indentation width, in 72000 DPI, for locator search text, which is used to identify subheader rows and differentiate them from header rows.

If the subheader row text starts to the left of regular rows, then the value must be set to slightly less than the indentation width so that the locator search text is limited to the table row and within the search width. Typically, this is set to **0**, which denotes that no special indentation is needed and searches for locator text will include the entire table row.

- **Locator**. Designates a unique name used to identify the locator. You must also define the search pattern and/or extraction settings for the locator. Click  and configure your locator options in the **Table Header Locator** dialog box that opens.

For more information about text locator settings, see [“Text Locator Properties General Tab” on page 486](#) and [“Text Locator Properties Extraction Settings Tab” on page 487](#).

6.15.3.10.8 Table Properties Extraction Tab

The **Extraction** tab contains options to specify the font filter and extraction settings for your table. It has the following values:

- **Text Print Direction.** Specifies the orientation in which the text prints or the degree of rotation on the page. From the drop-down menu, you can select from **Portrait** (0°), **Inverse Landscape** (90°), **Inverse Portrait** (180°), and **Landscape** (270°).
- **Include Form Text.** Indicates that text embedded in forms should be included in the search.
- **Font Filter.** Indicates which font filters should be used with your project.
- **Filter Type.** Specifies whether the list of the application's font filters are included or excluded. From the list, select **Include List** or **Exclude List**.

To review or edit additional properties, click on another tab in the XFT Table Field Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.10.9 Table Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your table field. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

Basic

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **TABLE.** Represents a table element, which contains table row elements. (If your table spans multiple pages, you must still tag it as a single table.)
 - **TOC.** Represents a table of contents element, which is a structured list of items with labels to identify each item and retains its own hierarchy. Additionally, this element is also suitable for other lists like Tables of Figures.
 - **ARTIFACT.** Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.

- **HEADER_ARTIFACT**. Denotes a header element that should be ignored.
- **FOOTER_ARTIFACT**. Denotes a footer element that should be ignored.
- **WMARK_ARTIFACT**. Denotes a watermark element that should be ignored.
- **LAYOUT_ARTIFACT**. Denotes a layout element that should be ignored.
- **BACKGR_ARTIFACT**. Denotes a background element that should be ignored.
- **PAGE_ARTIFACT**. Denotes a page element that should be ignored.
- **Tag Name**. Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language**. Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text**. Defines the actual text for this item that is used by screen readers.
- **Tag Only Once**. Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags**. Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.

Advanced

- **Alternate Cell Text**. Details an array of Alternate Text definitions with each applied to a specific range of rows and columns. The number of entries is shown on the DOM Properties tab and new array entries can be added by clicking the ellipsis button, , beside the field.
- **Table Summary**. Specifies a brief description of the table.
- **Table Header**. Defines table headers, including ^{row} headers, column headers, and table header cells. Click the ellipsis button, , to enter the values.

Text Links

- **Text Expression**. Indicates the regular expression defining the text used for the hyperlink. If no value is provided, all text will be linked.
- **Type**. Indicates the type of text link to create.
- **Target**. Specifies the URL or XPATH expression, depending on your selected target type.

- **Alternate URL Description.** Specifies an alternate description to use for the link to meet accessibility requirements.
- **MFCT Color.** Denotes the color of the link. In the table on the DOM Properties tab, you can select a color from the MFCT Color list. If you do not define one, the default color is used.
- **Underline.** Designates if the link is underlined.
- **Underline Thickness.** Specifies the underline thickness in XIF points.

To review or edit additional properties, click on another tab in the XFT Table Properties window. When you have completed making your selections, click **OK** to accept your changes and then the **Auto Detect Columns** dialog appears. The Detect Table Columns tool can assist with auto-detecting the columns within your table and is a convenient way of completing a first attempt at adding columns to your new table. To use the tool, click **Yes** and the **Detection Options** dialog for Detect Columns appears. For more information on working with this tool, see [“Detecting Table Columns” on page 367](#).

6.15.3.10.10 Configuring Multiuse Locators for Table Fields

The XFT Field Technology component can be configured to reuse a locator multiple times to identify the start/end point of a table field and the position of the top/bottom of the table. This feature is beneficial to users with documents where tables can start and stop on the same page or span multiple pages. Furthermore, the processing of documents with multiple instances of the same table on the same page is also greatly improved with the flexibility of multi-use locators.

An example of a document that multi-use locators are well-suited for is a table of transactions on a customer’s bank account or credit card statement since the size of the transactions table cannot be predetermined with an ending identifier because the dimensions of the table are likely to be different for each customer. Instead, this issue can be overcome by reusing a single locator to identify and/or position a table and then using the same locator to determine start/end identification or top/bottom positioning (these locators must be defined with the same name because the subsequent instance of the same locator ends the current table and also starts the next table). In situations where a successive end locator cannot be determined, such as at the end of a document, the position of the table will be determined in relation to the bottom of the page.

To enable multiple-use locators for multiple tables:

1. In the Preview pane of the XFT window define a **table** as usual, but be sure to draw the bounding box defining your table as large as possible while excluding any content at the top or bottom of the page that does not apply to your table.

It is recommended that you make the bounding box around your tables as large as possible because the assumption by the XFT Field Technology component is to use the bottom of the page to determine where the table ends in the absence of an end locator, hence, you want to include the maximum possible area for your table while excluding any unrelated content.

2. For your newly created table, open the XFT Table Field Properties dialog and set up a **top** Positioning locator, which will be a fixed position from the bounding box. Since this locator can be used to both identify the start/end and top/bottom positioning, if you have multiple instances of the same table they should all use the same locator name.



Note: If you set a locator and top offset above the fixed position of the bounding box, then the locator and top offset settings are used instead of the bounding box's fixed position.

3. Configure any other settings you require for your table. When you are finished, click **OK** to close the dialog.
4. Repeat steps 1-3 until you have defined all the necessary tables with the XFT Field Technology component for your document.

When you are finished defining all your tables, continue to the next step.

5. **Save** your settings.

When you run your project, the XFT Field Technology component automatically recognizes the presence or absence of end locators and will use your locators and page dimensions accordingly to determine start/end identifiers and top/bottom positions.

Advanced Configuration

The XFT Field Technology component's `ActiveObjectsMayUseTopLocator` and `EnableRepeatingSectionProcessing` parameters can also be set in order to facilitate more advanced configurations of multi-use table locators. For both parameters below, assume you have a multi-page document where tables span multiple pages and also contain some new tables starting on a page where a previous table ended.

ActiveObjectsMayUseTopLocator

When the `ActiveObjectsMayUseTopLocator` parameter is set to true (this is the default behavior), the component will use the subsequent table's top locator for positioning while assuming that it is a continuation of the previous table. However, if the following table is actually a new table and not a continuation of the previous table, then `ActiveObjectsMayUseTopLocator` should be set to false so that the component will recognize that it is a new table. Alternatively, instead of disabling this parameter you can draw a fixed table bounding box in order to define locator positioning with absolute values.

EnableRepeatingSectionProcessing

Some documents contain repeating tables on the same page, which may cause difficulty in determining the locators for where a particular table ends. The `EnableRepeatingSectionProcessing` parameter can be enabled to use the start locator for the next table as the end locator and bottom positioning coordinate for the current table. By default, this parameter is disabled.

Related Links

- [“XFT Window” on page 350](#)
- [“Table Properties Positioning Tab” on page 434](#)
- [“ActiveObjectsMayUseTopLocator”](#)
- [“EnableRepeatingSectionProcessing”](#)

6.15.3.11 Adding a Column

Columns are used to determine the left and right boundaries of a set of data within a table. Each column is defined by name and may specify its own threshold settings to process the joining of text elements differently.



Note: You must first have defined a table prior to adding a column.

To add a column:

1. From the XFT window toolbar, select **Add Column**, .
2. When you move the cursor over the image preview area, the cursor changes from an arrow to a crosshair. With the crosshair cursor, click and drag to draw a bounding box around the area you want to define as a column. The area that you want to mark as a column must be within the boundaries of a previously defined table.
The **XFT Table Column Properties** window appears.
3. Complete the fields in the XFT Table Column Properties dialog box and when you are done, click **OK**.

6.15.3.11.1 Column Properties General Tab

The **General** tab specifies general properties for the columns. It has the following fields:

- **Name.** Specifies a name for your table. The name must be unique within the project as it is used to reference this item in report logs.
- **Thresholds Settings.** Indicates the threshold settings to use. You may opt to use the default settings or you can click the ellipsis button to display the Select Threshold dialog window to Add, Edit, or Remove a threshold you have defined. For more information, see [“Adding a Threshold” on page 462](#).
- **Include Form Text.** Indicates that text embedded in forms should be included in the search.
- **Position and Size.** Specifies the default position and size of the section.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.

- **Width.** Specifies the width of the bounding box.
- **Height.** Specifies the height of the bounding box.
- **Empty Cell.** Specifies options for adding an invisible text string to empty cells.
 - **Enable.** Indicates that adding text to empty cells is enabled.
 - **Text to Add.** Specifies the text string to add to the empty header cells, which can be announced by a screen reader.

When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.12 Adding a Cell

A cell is used to define a group of fields within a table.



Note: Before adding a cell, you must create a table as the cell must exist within the table's boundaries.

To add an XFT cell:

1. In the **Design** pane, select the table where you want to create the cell.
2. From the XFT window toolbar, navigate to **Add New> Add Cell**, .
3. When you move the cursor over the Preview pane, the cursor changes from an arrow to a crosshair. Using the crosshair cursor, draw a bounding box around the fields you want to define as a cell within the boundaries of the selected table.
4. Complete the fields in the **XFT Cell Properties** dialog and when you are done, click **OK**. (For more information about the options available on the tabs, see the subsequent topics in this chapter.)
5. Next, you can define the contents of the cell by adding additional XFT elements to the cell. (If you do not define the contents, the cell will be treated as a text field.) In the Design pane, select the newly created cell.
6. From the XFT window toolbar, click **Add New** and then select the **Add** option for the type of field you want to define within the cell. You can select from adding a **Text**, **Object**, **Form**, or **List** field type definition.
7. In the Preview pane and within the boundaries of the cell, draw a bounding box around the area to define.
8. In the properties dialog that appears for your XFT element type, complete the fields and when you are done, click **OK**. (See the respective chapter on the field definition type you are adding for more information on completing its properties.)
9. Repeat steps 5-8 as necessary until you have defined all the required XFT elements within the selected cell.

6.15.3.12.1 Cell Properties General Tab

The **General** tab specifies general properties for the cell. It has the following fields:

- **Name.** Specifies a name for your cell. The name must be unique within the project as it is used to reference this item in report logs.
- **Applies to rows.** Designates the rows the cell attributes apply to within the table.
 - **All Rows.** Denotes that the cell attributes apply to all rows within the table.
 - **Row Number.** Denotes that the attributes apply only to the specified rows. If you want the cell to apply to any row, you can use the value -1.
- **Applies to columns.** Designates the columns the cell attributes apply to within the table.
 - **All Columns.** Denotes that the cell attributes apply to all columns within the table.
 - **Named Columns.** Denotes that the attributes apply only to the specified columns.
- **Position and Size.** Specifies the default position and size of the cell.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.
 - **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the XFT Cell Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.12.2 Cell Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your cell. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:

- **SECT**. Represents a standalone section element.
- **DIVISION**. Represents a division element, which can be either an individual or group of block-level elements.
- **NONE**. Indicates that no tag should be applied to the content.
- **CAPTION**. Represents a caption element, which is a segment of text describing a figure or table.
- **NOTE**. Represents a note entry element, which is supplementary text that is referenced in the main body of content. Note elements include footnotes and endnotes, but can also consist of inline elements with special formatting.
- **BLOCKQUOTE**. Represents a block quote element, which is one or more paragraphs of text that are not inside a paragraph or other block-level element. Similar to the **QUOTE** tag, which consists of an inline segment of quoted text as opposed to at least one paragraph in length for **BLOCKQUOTE**.
- **INDEX**. Represents an index element.
- **TOC**. Represents a table of contents element, which is a structured list of items with labels to identify each item and retains its own hierarchy. Additionally, this element is also suitable for other lists like Tables of Figures.
- **TOCI**. Represents a table of contents item element, which is an item enclosed in a table of contents element list.
- **L**. Represents a list element. This tag type is used to signify the start of a list structure; direct child elements should use the list item (**LI**) tag type to enclose individual content items within the list.
- **LI**. Represents a list item element, which is any individual component in a list.
- **LBODY**. Represents a list item body element, which is the text content of a list item.
- **P**. Represents a paragraph or other generic block-level element.
- **SPAN**. Denotes a span element, which is used to group inline elements that cover two or more columns. The **SPAN** tag is typically used when you want the inline elements grouped, but merging the cells together would not be appropriate. Furthermore, for many screen readers this tag is a requirement when setting alternate text for it to be read aloud.
- **H**. Represents a heading element.
- **H1** to **H6**. Represents six levels of headings, with **H1** being the highest or most important heading level and **H6** indicating the lowest level.
- **ARTIFACT**. Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.

- **HEADER_ARTIFACT**. Denotes a header element that should be ignored.
- **FOOTER_ARTIFACT**. Denotes a footer element that should be ignored.
- **WMARK_ARTIFACT**. Denotes a watermark element that should be ignored.
- **LAYOUT_ARTIFACT**. Denotes a layout element that should be ignored.
- **BACKGR_ARTIFACT**. Denotes a background element that should be ignored.
- **PAGE_ARTIFACT**. Denotes a page element that should be ignored.
- **Tag Name**. Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language**. Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text**. Defines the actual text for this item that is used by screen readers.
- **Tag Only Once**. Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags**. Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.

To review or edit additional properties, click on another tab in the XFT Cell Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.12.3 Cell Properties Auto Detect Tab

The Auto Detect tab allows you to configure the auto detection rules performed on the associated cell. It has the following fields:



Note: This tab is only active after you have selected the **Enable Auto Detect** option on the General tab of the XFT Cell Properties window.

- **Threshold Name**. Denotes the XFT threshold options to use for auto detection. You can select an existing threshold from the dropdown menu or select the ellipsis button, , to open the **Select Threshold** dialog to **Add** a new threshold. You can also **Edit** or **Remove** an existing threshold from the Select Threshold window. The **Default** threshold is selected by default and cannot be deleted. For more information, see [“Adding a Threshold” on page 462](#).

- **Auto Detect Options.** Denotes the auto detection tagging options to use on located elements. You can select an existing auto detect options set from the dropdown menu or select the ellipsis button, , to open the **Auto Detect Options** dialog to **Add** a new options set. To configure the properties, select an existing option set and click the **Edit** button. You can also **Remove** an existing auto detect option from the Auto Detect Options window. The **Default** auto detect option is selected by default and cannot be deleted. For more information, see “XFT Section Auto Detect Option Properties” on page 387.
- **Image Alternate Text.** Defines the alternate text for auto detected images in this section. Some of the options contain further fields that must be set so it is recommended you click the ellipses button, , to display the **Alternate Text** dialog where you can configure the options. If any additional fields must be configured after selecting an option from the **Type** list, the new fields are displayed below.
 - **STATIC.** Indicates that the alternate text is fixed and does not change. In the Static field, enter the alternate text you want to apply to images.
 - **FORMATTER.** Specifies the user of alternate text that is reformatted by a formatter class. The following fields must be completed:
 - **Locale.** Contains locale settings to help classify the alternate text.
 - **Language.** Indicates the language code for the alternate text as defined by ISO-639 standards.
 - **Country.** Indicates the country code for the alternate text as defined by ISO-3166 standards.
 - **Index Range.** Contains options for incorporating the index into XFT results. If your results return multiple lines, this index points to the lines containing the text to be formatted.
 - **Start.** Designates the starting point of the index range. A negative number used as the value for the start results in a relative position from the end of the range, while a zero (0) value represents the last item in the field.
 - **End.** Designates the ending point of the index range. A positive number represents an absolute position in the range, while a negative number represents an offset from the last item. A zero (0) value represents the last item in the field.
 - **Input Mask.** Describes the format of the date as input from the original data to be used as a mask. This parameter only appears when the FormatDate formatter class is selected. The following characters can be used as accepted values:
yy = Two digit year (10)
yyyy = Four digit year (2010)
M = Single digit month (1)

MM = Two digit month (01)

MMM = Short form month name (Jan)

MMMM = Long form month name (January)

d = Single digit day of month (9)

dd = Two digit day of month (09)

E = Short form day of the week (Tue)

EEEE = Long form day of the week (Tuesday)

For instance, a sample mask for 31/01/2010 would be represented by dd/MM/yyyy.

- **Output Mask.** Describes the format of the date returned by the formatter to be used as a mask. This parameter only appears when the `FormatDate` formatter class is selected. The following characters can be used as accepted values:

yy = Two digit year (10)

yyyy = Four digit year (2010)

M = Single digit month (1)

MM = Two digit month (01)

MMM = Short form month name (Jan)

MMMM = Long form month name (January)

d = Single digit day of month (9)

dd = Two digit day of month (09)

E = Short form day of the week (Tue)

EEEE = Long form day of the week (Tuesday)

For instance, a sample mask for 31/01/2010 would be represented by dd/MM/yyyy.

- **Character to Replace.** Indicates the character(s) to replace. This parameter only appears when the `ReplaceCharacter` formatter class is selected.
- **Replacement Character.** Indicates the character(s) to use when replacing the character(s) specified in `Character to Replace` parameter. If no replacement character is specified, the input character is simply removed. This parameter only appears when the `ReplaceCharacter` formatter class is selected.
- **PROVIDER.** Indicates that the alternate text contains settings for specifying a reference to an instance of an `AlternateTextProvider` as defined by the `TextProviders` collection. This is useful when reusing large chunks of static text across all documents in a large job, as it allows you to insert the section as an object, saving time and disk space in compiling the output PDFs, such as those used for customer billing. An object is more efficient to insert into a PDF, but then it is not readable by screen reader technologies. By using a

TextProviderReference to instruct screen readers where the text content is located, a screen reader can read the section from the source text without impacting the structure and efficiency of the PDF.

- **Instance Name.** References the Instance Name of an AlternateTextProvider as defined in the TextProviders collection. On the Alternate Text dialog, you can select **Browse Providers**, , to bring up the Providers dialog where you can configure your provider settings. For more information, see [“Managing Alternate Text Providers” on page 321](#).
- **Properties.** Controls custom properties which, if present, are passed into the providers as arguments. The properties defined here are specific to the MarkupItem associated with this ProviderInstance reference.
- **DISABLED.** Turns off alternate text for images in the output.
- **CS_ALT_TEXT_MANAGER.** Retrieves the approved alternate text used for images from the Alternate Text Manager console.
- **IMAGE_SOURCE.** Specifies that the Constant Print image’s alternate text setting in the DOM properties is used.

When auto detect is called, any tagged elements found in the cell are pulled inline along with their associated DOM structure to the auto detection node in order to maintain the flow’s reading order. Any elements that are not already tagged will be added to the DOM structure with their detected types.

To review or edit additional properties, click on another tab in the XFT Cell Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

XFT Cell Auto Detect Option Properties

The auto detect options for XFT Cell help you control how page elements are detected within your documents. By taking the time to fine tune these options about your sections, the auto detection tool can return better results from your document. Multiple option sets can be created to ensure an option set is readily available to fit the criteria for variations in page design of sections within the same document. The Default auto detect option set is used by default and cannot be deleted. The following parameters are available to be configured for an auto detect options set:

- **JoinUnderRatio.** Specifies the ratio of font height to the next line within the same paragraph.
- **JoinDiffFonts.** Denotes whether your section contains instances of consecutive lines with different fonts joining within the same paragraph. By default, this is disabled.
- **InTableColGap.** Defines the minimal amount of space, in 72000 DPI, between columns within a table.
- **AllTableRowsHaveNumbers.** Indicates whether all table rows possess at least one column with numeric data, which can be useful in keeping cells together until there is a cell with a numeric value. By default, this is disabled.

- **TableStartSameFont.** Indicates whether all instances of the first row in tables use the same font. By default, this is disabled.
- **TableMaxLinesToStart.** Denotes the maximum number of lines that can appear at the start of a table. If there is no maximum, enter 0.
- **BoldFontMeansH1.** Indicates whether an isolated bold line should be tagged as H1 (level one heading). By default, this is enabled.
- **TextToTag.** Contains a list of text constants and patterns to be marked with a predefined tag type. If multiple TextToTag sets are enabled concurrently, you must observe the order in which the sets appear in your list since the options from the first matching set is used.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
 - **TagType.** Indicates the type of tag to apply to data matching your specified value.
- **FontCompare.** Contains options for comparisons based on font names, styles, and point sizes.
 - **Fontname.** Denotes the regular expression name of the font typeface such as **.*Bold** or **Arial-Bold**. If no value is provided, then the font name check is not performed.
 - **Styles.** Contains a comma separated list of acceptable font styles, represented by letters. You can enter **B** for Bold, **I** for Italic, **BI** for Bold-Italic, or **N** for Normal (neither bold or italic). For example, using the value “**B,BI**” would match fonts using the bold or bold-italic styles while ignoring normal and italic styles. If no value is provided, all styles are matched.
 - **PtSizes.** Contains a comma separated list of decimal numbers or number ranges to indicate font point sizes. For example, “**10.3,11-15**” would match fonts with a point size of 10.3 or anything between 11 and 15. If no value is provided, all point sizes are matched.
- **SingleLineOnly.** Designates whether only single lines (lines not joined into a paragraph) are processed. By default, this is disabled.
- **SegmentToCompare.** Denotes the portion of the paragraph or line to compare against. You can select from the following options:

- **FIRST_LINE**. Indicates that only text from the first line of the paragraph is compared.
- **FIRST_FONT**. Indicates that only text from the first font is compared when a paragraph containing multiple fonts is encountered.
- **FULL_PARA**. Indicates that text from the whole paragraph is compared.
- **SegmentToTag**. Denotes the portion of the paragraph or line to tag. You can select from the following options:
 - **FIRST_LINE**. Indicates that the first line is tagged as indicated by the TagType parameter while the remainder is marked with the P (paragraph) tag.
 - **FIRST_FONT**. Indicates that the text in the first font is tagged as indicated by the TagType parameter while the remainder is marked with the P (paragraph) tag.
 - **FULL_PARA**. Indicates that the whole paragraph is tagged as indicated by the TagType parameter.
- **FalseTableStart**. Contains a list of patterns and constant text to compare against as tables are detected. When a match is found, the table start is ignored in order to provide an override for incorrectly detected tables.
- **IsActive**. Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
- **TextType**. Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
- **IgnoreCase**. Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.
- **MoreUpperNoPunctuation**. Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
- **Value**. Specifies the pattern or constant text to compare against.
- **CombineSuperscript**. Denotes whether to join two base lines as one if the two base lines join in a manner that is likely to be superscript text. By default, this is enabled.
- **HandleVectorCelledTables**. Designates whether tables with cells that are contained within vector lines are processed with the vector lines intact. (Note that enabling this setting may affect the precision of the cell detection logic as not all vector lines are drawn in a manner that is completely compatible.)
- **ListLBLFonts**. Specifies the fonts used in list labels. Any text using these fonts following a bullet character is included with the list's LBL tag. If any of these values end with an asterisk (*), a BeginsWith test is performed.

- **BulletChars.** Contains a list of characters or strings that indicate a list is starting. There are two special values which may be used:
 - **#.** (pound period), which will match 1, 2, or 3 digit numbers followed by a period.
 - **@** (at-sign right-parentheses), which will match 1, 2, or 3 lower case letters followed by a right parentheses.
- **BreakOnObject.** Designates whether paragraph joins are stopped when a vector or image is between two lines of text. This setting may have an effect on how tables are recognized.
- **NotesReferences.** Contains options for identifying notes and references within your documents.
- **IncludeList.** Designates any patterns to include during note and reference processing.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using **REGEX** setting for **TextType**, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
 - **StyleType.** Indicates the style of text for comparison. You can choose from **SUPERSCRIPT**, **NON_SUPERSCRIPT**, and **ANY**.
- **ExcludeList.** Designates any patterns to exclude from note and reference processing.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using **REGEX** setting for **TextType**, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.

- **Value.** Specifies the pattern or constant text to compare against.
- **StyleType.** Indicates the style of text for comparison. You can choose from **SUPERSCRIPT**, **NON_SUPERSCRIPT**, and **ANY**.
- **Name.** Specifies the unique name assigned to this set of options.

6.15.3.12.4 XFT Cell Best Practices

Manually tagging a table with the proper XFT elements for cells can be a time-consuming endeavour, especially for sizeable tables with various types of content. To help make this procedure more efficient, it is recommended you review the following strategies and notes before marking up your documents:

- Assume you have a document containing a large table with mostly text-based cells and a few cells with images. Instead of manually tagging each text cell individually or as separate groups, you can tag the entire table as an XFT text element and in the XFT Cell Properties dialog, apply this setting to **All Rows**. Next, you should define the cells containing images. After defining the boundaries of the image's cell, use the **Row Number** and **Named Columns** parameters in the XFT Cell Properties dialog to establish the location of the cell. Continue defining XFT image elements for all other instances of image cells in the table while specifying its precise location each time. Consequently, while the image cell does match the text element's initial criteria of All Rows, the image cell's location is specifically named so the application's logic knows to apply your image element options to these cells instead of the text element's options.
- If you do not define a cell within a table, then the contents of the table will be treated as an XFT text field by default.
- The location of cells are relative to the table.
- When working with DOM properties, disabling DOM at the parent level also disables it for the child nodes. For example, switching off DOM properties at the section level will disable the creation of the DOM tree for the section and all field definitions below.

6.15.3.13 Adding a Threshold

A threshold defines how pieces of text are joined together to make words, phrases, and paragraphs.

To add a threshold, select **Add Threshold**, , from the XFT window toolbar and the XFT Threshold Properties dialog appears. Within the threshold settings, depending on your desired outcome, configure the settings on the **Word Join**, **Phrase Join**, or **Paragraph Join** tabs.

6.15.3.13.1 Threshold Properties General Tabs

The **General** tab specifies general properties to define the threshold. It has the following common fields for all join types:

- **Name.** Indicates the name for the group of threshold options. These settings can be used throughout Output Transformation Engine and shared by sections, fields, and tables.
- **Base Line Threshold Option.** Determines the type of base line threshold variance to use. You can choose from **Base Line Threshold** or **Base Line Percent Threshold**. For more information, see the respective parameter descriptions for each threshold variance type.
- **Base Line Variance.** Determines the number of pixels of variance from the base line that text elements can have, but still be included on the same logical line. If there is a difference of 1 pixel or more, words on different base lines may not be joined together unless you set a value.

For example, the text “12345 1st Street” could be treated as 2 lines; “st” on the highest base line as one line, and “12345 1 Street” as the second line. By setting the base line variance, the text in this example would be rendered as text elements on a single line.

- **Point Size Variance %.** Determines the base line threshold variance value as a scalable percentage of the text point size. Typically, using point size variance percentage in documents with varying font sizes eliminates the need to manage multiple threshold definitions.
- **Text Direction.** Determines the reading order for the text in your input document. The default setting, **Left to Right**, is suitable for most languages, but languages containing scripts with the reverse reading order, such as Arabic or Hebrew, must use the **Right to Left** setting. If you are using a type of XFT field definition that can handle Document Object Model (DOM) properties, the original reading order is retained in your DOM elements.

You must also configure the joining threshold type by setting options on the Word Join, Phrase Join, or Paragraph Join tabs.

The **Word Join** tab controls how words are determined by the program. You can join two or more pieces of text that visually are the same word, but have different fonts for some of the characters, or are delimited by the input data stream controls (i.e. split across records). If two segments of text start or end within the threshold, the two segments are joined to make a single word.

The **Phrase Join** tab controls how phrases are established. You must specify the conditions for connecting two or more words into a phrase. If the two words start or end within the threshold, they are joined.

The **Paragraph Join** tab controls how paragraphs are detected. The paragraphs are created by spanning phrases across lines, which is needed to join text on separate base lines. If two lines start or end within the threshold, they are joined to make a paragraph.

The following options are available to be set on the **Word Join**, **Phrase Join**, or **Paragraph Join** tabs:

- **Enable.** Indicates that the settings on the particular joining threshold type tab should be used.
- **Inline Options.**
 - **Options.** Determines how the inline absolute and percentage values will be used to determine if text elements are joined based on their distance apart.
 - **Inline Abs.** Specifies the maximum number of points that an element can be away from another element to still be considered joined.
 - **Inline %.** Specifies the maximum percentage of space between two elements to still be considered joined. The value is calculated from the given percentage of the space character; therefore, 150% is calculated as 1.5 multiplied by the space character's width in points.
- **Options.**
 - **Font growth.** Indicates whether text elements that grow in font size can be joined.
 - **Font change.** Indicates whether text elements that have different fonts can be joined.
 - **Size change.** Indicates whether text elements that have different font sizes can be joined.
 - **One space.** Indicates whether all joined words will only have a single space between them. Otherwise, the joined words will have the number of spaces that was originally calculated between the two words.

To review or edit additional properties, click on another tab in the XFT Threshold Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.13.2 Font Based Thresholds Tab

The **Font Based Thresholds** tab allows you to create font-based base line thresholds, which override the default thresholds when matched.

To set up a new font-based base line threshold, click **Add** and configure the new threshold entry in the table.

The following options are available to be set on the Font Based Thresholds tab:

- **Base Line Threshold Option.** Determines the type of base line threshold variance to use. You can choose from **Base Line Threshold** or **Base Line Percent Threshold**. For more information, see the respective parameter descriptions for each threshold variance type.
- **Base Line Variance.** Determines the number of pixels of variance from the base line that text elements can have, but still be included on the same logical line. If

there is a difference of 1 pixel or more, words on different base lines may not be joined together unless you set a value.

For example, the text “12345 1st Street” could be treated as 2 lines; “st” on the highest base line as one line, and “12345 1 Street” as the second line. By setting the base line variance, the text in this example would be rendered as text elements on a single line.

- **Point Size Variance %.** Determines the base line threshold variance value as a scalable percentage of the text point size. Typically, using point size variance percentage in documents with varying font sizes eliminates the need to manage multiple threshold definitions.
- **Font Names.** Specifies the font names the threshold settings apply to. All text using the listed fonts is affected. In the **Font Selection** box, you must move the fonts from the **Available fonts** list to the **Selected fonts** list.

To review or edit additional properties, click on another tab in the XFT Threshold Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.14 Adding a Container

A container is used to define one or more grouped fields or tables.



Note: Before adding a container, you must create a section as the container must exist within the section's boundaries.

To add an XFT container:

1. In the **Design** pane, select the section where you want to create the container.
2. From the XFT window toolbar, navigate to **Add New> Add Container**, .
3. When you move the cursor over the image preview area, the cursor changes from an arrow to a crosshair. Using the crosshair cursor, draw a bounding box around the fields or tables you want to define as a container within the boundaries of the selected section.
4. Complete the fields in the **XFT Container Properties** dialog and when you are done, click **OK**. (For more information about the options available on the tabs, see the subsequent topics in this chapter.)
5. Next, you must define the contents of the container by adding additional XFT elements to the container. In the Design pane, select the newly created container.
6. From the XFT window toolbar, click **Add New** and then select the **Add** option for the type of field you want to define within the container. You can select from adding a **Text**, **Object**, **Form**, **List**, **Table**, or a nested **Container** field type definition.
7. In the Preview pane and within the boundaries of the container, draw a bounding box around the area to define.

8. In the properties dialog that appears for your XFT element type, complete the fields and when you are done, click **OK**. (See the respective chapter on the field definition type you are adding for more information on completing its properties.)
9. Repeat steps 5-8 as necessary until you have defined the required XFT elements within the selected container.

6.15.3.14.1 Container Properties General Tab

The **General** tab specifies general properties for the container. It has the following fields:

- **Name.** Specifies a name for your container. The name must be unique within the project as it is used to reference this item in report logs.
- **Enable.** Indicates that container processing is enabled. If it is disabled, the engine does not look for container identifiers or locators.
- **Options.** Determines which additional tabs are available/enabled. You can select from **Enable Start Identification**, **Enable End Identification**, **Enable Dynamic Positioning**, **Enable Multiple Instances Per Section**, **Enable DOM Properties**, or **Enable Auto Detect**. Enabling either of the identification types or dynamic positioning opens up further options on separate tabs, which are detailed in separate topics.

Dynamic positioning refers to sections that are located based on their position relative to locators you specify, even if they shift on the page. Multiple instances per section indicates to the engine that your document has more than one instance of the same container within a section.

- **Location Behavior.** Determines locator behavior for the container.
- **Search Option.** Denotes the direction to search through the document. Typically, multiple instances of containers within a document appear in a top to bottom manner, but sometimes they may be ordered left to right so the correct search direction must be indicated.
- **Allow Locator Duplicates For Identification and Allow Locator Duplicates For Positioning.** When the container repeats on the page and its end cannot easily be determined, select these check boxes to use the start locator of the next section as the end locator and bottom positioned of the current container. This means that the start locator is used for start and end identification, as well as top and bottom positioning. The next occurrence of the locator indicates the end of the previous section and the start of the next section. These features are disabled by default.
- **Position and Size.** Specifies the default position and size of the container.
 - **Left.** Specifies the position of the left edge of the bounding box.
 - **Top.** Specifies the position of the top edge of the bounding box.
 - **Width.** Specifies the width of the bounding box.
 - **Height.** Specifies the height of the bounding box.

To review or edit additional properties, click on another tab in the XFT Container Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.14.2 Container Properties Start Identification Tab

The **Start Identification** tab specifies when a new container begins. It has the following fields:



Note: This tab is only active after you have selected the **Enable Start Identification** option on the General tab of the XFT Container Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator.

To add a start identification, click **Add Identification**, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list in this column. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence or non-existence of forms on a page. Click the **Add** or **Remove Filter** buttons ( and , respectively) to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.
 - **Require Any.** Processes an item if any of the form filter parameters in your list are found.
 - **Ignore All.** Ignores the page only if all form filter parameters are found.
 - **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Container Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.14.3 Container Properties Positioning Tab

The **Positioning** tab allows you to set the properties for locating containers by basing each position of the container's bounding box on the selected locator(s) as specified in the Locator Priority list. This type of dynamic positioning is helpful since containers are located based on their position relative to locators you specify, even if they move on the page.



Note: This tab is only active after you have selected the **Enable Dynamic Positioning** option on the General tab of the XFT Container Properties window.

The following rules govern the Locator Priority list:

- If no locator name is given, then the corresponding position is an absolute value.
- If a locator name is given, the base line point of the found locator's search string is used as the starting point from which the section's position is offset by the specified offset value. The offset value can be a positive or negative integer:
 - A positive number will offset the side either down or to the right.
 - A negative number will offset the side either up or to the left.
- If a defined locator cannot be found, the next locator in the Locator Priority list is searched for and its values will be applied to the section's original bounding box position.

The following locator offset options may be set or performed on this tab:

- **Locator Based Coordinates.** Designates which offset values are allowed to be set.
 - **Top.** Indicates that the top side of the field's bounding box is offset by the given amount.
 - **Left.** Indicates that the left side of the field's bounding box is offset by the given amount.
 - **Right (Width).** Indicates that the right side of the field's bounding box is offset by the given amount.
 - **Bottom (Height).** Indicates that the bottom side of the field's bounding box is offset by the given amount.

You may also define which locator based coordinate's setting (Top, Left, Bottom, and Right) takes precedence over the other locators when trying to establish the final position of a locator. This feature is advantageous for users who are utilizing multi-use locators in order to determine the top, left, bottom, or right dynamic positioning of a field since the LocationToUse setting provides the flexibility of selecting which positioning location settings a particular locator should use.

For top and bottom position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Highest.** Indicates that the highest of the locators plus any offset position settings is used to determine the position for the locator.
- **Lowest.** Indicates that the lowest of the locators plus any offset position settings is used to determine the position for the locator.

For left and right position locators, you can select from the following LocationToUse options:

- **LastInList.** Indicates that the last active locator in the Locator Priority list is used to determine the position for the locator.
- **Leftmost.** Indicates that the locator farthest to the left plus any offset position settings is used to determine the position for the locator.
- **Rightmost.** Indicates that the locator farthest to the right plus any offset position settings is used to determine the position for the locator.
- **Locator Priority List toolbar.** Performs various tasks related to the Locator Priority list.
 - **Add.** Adds a locator offset to the Locator Priority list. For more details on adding a locator offset to the Locator Priority list, please see below.
 - **Remove.** Deletes the selected locator offset settings from the Locator Priority list.
 - **Move Up.** Repositions the locator up in the Locator Priority list.
 - **Move Down.** Repositions the locator down in the Locator Priority list.



Note: Be aware that locators are processed in the sequence listed and subsequent locators found will override any offset values of previous locators found. (However, when using a value of zero (0) in a locator, the previous offsets are not altered.) Negative offset values indicate a position either above or to the left of a locator, while positive offset values indicate a position to the right or below a locator. If only top and bottom locators are set, then the left and width values are retrieved from the General tab.

Occasionally, a section may be followed by multiple different sections in separate documents. In these situations, one locator can be used to position the top of the section and then multiple locators used to define the bottom with special attention paid to the order of locators in the list since each subsequent value for a particular locator overrides any previous values. To mitigate any issues, it is recommended that the most likely locator be listed last. For instance, if section A can be followed by sections B or C, then the top value should be set relative to locator A and then negative bottom values that are set relative to locators B and C. By configuring the values like this, if section B and C both exist, then section A ends at B and will not continue to C.

To add a locator offset to the Locator Priority list, click **Add**, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list in this column. For more information, see [“Adding a Text Locator” on page 486](#).
- **Offsets from Locator (+/-).** Specifies the distance, in DPI, that the side of the field’s bounding box is offset from the found locator’s base line point. Refer to the rules managing base line point offset values above.



Tip: Place locators which may not be found closer to the bottom of the Locator Priority list. This will ensure that their values are not overwritten by locators below them in the list. An offset value of zero (0) will not change previous offsets.

To review or edit additional properties, click on another tab in the XFT Containers Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.14.4 Container Properties End Identification Tab

The **End Identification** tab specifies when a container concludes. The end identifier does not have to be on the same page as the start identifier. It has the following fields:



Note: This tab is only active after you have selected the **Enable End Identification** option on the General tab of the XFT Container Properties window.

- **Required Confidence Factor.** Indicates the confidence factor you have set for this identifier.
- **Add/Remove Identification.** Allows you to initialize or delete the properties of a locator.

To add a start identification, click the Add Identification button, , and complete the following columns:

- **Locator Name.** Indicates the locator to use. Select a locator from the list in this column. For more information, see [“Adding a Text Locator” on page 486](#).
- **Required.** Indicates whether the locator is required. If it is required, then the locator must be found for the start or end condition to be met.
- **Confidence Factor.** Specifies the confidence factor value for the locator. For more information, see [“Confidence Factor” on page 345](#).
- **Forms Filter.** Sets filter parameters to indicate whether processing should continue based on the existence/absence of forms on a page. Click the **Add** or **Remove Filter** buttons to add or remove form filter information.
- **Filter Type.** Identifies the type of filter that will be applied during processing. You can select from the following types:
 - **Require All.** Processes items only if all form filter parameters are found.

- **Require Any.** Processes an item if any of the form filter parameters in your list are found.
- **Ignore All.** Ignores the page only if all form filter parameters are found.
- **Ignore Any.** Ignores the page if any of the form filter parameters are found.

To review or edit additional properties, click on another tab in the XFT Container Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.14.5 Container Properties DOM Properties Tab

The **DOM Properties** tab specifies Document Object Model configuration for your container. It has the following fields:



Note: This tab is only active after you have selected the **Enable DOM Properties** check box on the General tab. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

- **Tag Type.** Defines the tag type for this node/structural element. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner. You can select from the following tag types:
 - **DIVISION.** Represents a division element, which can be either an individual or group of block-level elements.
 - **NONE.** Indicates that no tag should be applied to the content.
 - **CAPTION.** Represents a caption element, which is a segment of text describing a figure or table.
 - **NOTE.** Represents a note entry element, which is supplementary text that is referenced in the main body of content. Note elements include footnotes and endnotes, but can also consist of inline elements with special formatting.
 - **BLOCKQUOTE.** Represents a block quote element, which is one or more paragraphs of text that are not inside a paragraph or other block-level element. Similar to the QUOTE tag, which consists of an inline segment of quoted text as opposed to at least one paragraph in length for BLOCKQUOTE.
 - **INDEX.** Represents an index element.
 - **TOC.** Represents a table of contents element, which is a structured list of items with labels to identify each item and retains its own hierarchy. Additionally, this element is also suitable for other lists like Tables of Figures.
 - **TOCI.** Represents a table of contents item element, which is an item enclosed in a table of contents element list.

- **L**. Represents a list element. This tag type is used to signify the start of a list structure; direct child elements should use the list item (LI) tag type to enclose individual content items within the list.
- **LI**. Represents a list item element, which is any individual component in a list.
- **LBODY**. Represents a list item body element, which is the text content of a list item.
- **LBL**. Represents a list item label, which can contain distinctive information like number or bullet character.
- **P**. Represents a paragraph or other generic block-level element.
- **H**. Represents a heading element.
- **H1** to **H6**. Represents six levels of headings, with H1 being the highest or most important heading level and H6 indicating the lowest level.
- **ARTIFACT**. Designates that the tagged content should be ignored by assistive technologies when presenting the content to accessibility users. Typically, Artifact tags are used for content such as lines, borders, or other images in documents that may inhibit accessibility users from properly understanding the document when disclosed through the tag structure.
- **HEADER_ARTIFACT**. Denotes a header element that should be ignored.
- **FOOTER_ARTIFACT**. Denotes a footer element that should be ignored.
- **WMARK_ARTIFACT**. Denotes a watermark element that should be ignored.
- **LAYOUT_ARTIFACT**. Denotes a layout element that should be ignored.
- **BACKGR_ARTIFACT**. Denotes a background element that should be ignored.
- **PAGE_ARTIFACT**. Denotes a page element that should be ignored.
- **Tag Name**. Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language**. Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the dropdown menu or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text**. Defines the actual text for this item that is used by screen readers.
- **Tag Only Once**. Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.

- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags.
- **Logical Order.** Defines how elements within the container are ordered as they are inserted. You can choose from **Static** or **Auto** ordering. With Static ordering, the order in which the elements are defined and processed in is used when inserting elements into the DOM. Use this option when the order within your document is known to be consistent. With Auto ordering, elements are ordered based on their graphic page position. This option is useful when the exact order of elements may vary depending on the position of the elements on the physical page.

To review or edit additional properties, click on another tab in the XFT Container Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

6.15.3.14.6 Container Properties Auto Detect Tab

The **Auto Detect** tab allows you to configure the auto detection rules performed on the associated container. It contains the following fields:



Note: This tab is only active after you have selected the **Enable Auto Detect** option on the General tab of the XFT Container Properties window.

- **Threshold Name.** Denotes the XFT threshold options to use for auto detection. You can select an existing threshold from the dropdown menu or select the ellipsis button, , to open the Select Threshold dialog to Add a new threshold. You can also Edit or Remove an existing threshold from the Select Threshold window. The Default threshold is selected by default and cannot be deleted. For more information, see [“Adding a Threshold” on page 462](#).
- **Auto Detect Options.** Denotes the auto detection tagging options to use on located elements. You can select an existing auto detect options set from the dropdown menu or select the ellipsis button, , to open the Auto Detect Options dialog to Add a new options set. To configure the properties, select an existing option set and click the Edit button. For more information about these parameters, see [“XFT Containers Auto Detect Option Properties” on page 476](#). You can also Remove an existing auto detect option from the Auto Detect Options window. The Default auto detect option is selected by default and cannot be deleted.
- **Image Alternate Text.** Defines the alternate text for auto detected images in this section. Some of the options contain further fields that must be set so it is recommended you click the ellipses button, , to display the **Alternate Text** dialog where you can configure the options. If any additional fields must be configured after selecting an option from the **Type** list, the new fields are displayed below.
 - **STATIC.** Indicates that the alternate text is fixed and does not change. In the Static field, enter the alternate text you want to apply to images.
 - **FORMATTER.** Specifies the user of alternate text that is reformatted by a formatter class. The following fields must be completed:

- **Locale.** Contains locale settings to help classify the alternate text.
 - **Language.** Indicates the language code for the alternate text as defined by ISO-639 standards.
 - **Country.** Indicates the country code for the alternate text as defined by ISO-3166 standards.
- **Index Range.** Contains options for incorporating the index into XFT results. If your results return multiple lines, this index points to the lines containing the text to be formatted.
 - **Start.** Designates the starting point of the index range. A negative number used as the value for the start results in a relative position from the end of the range, while a zero (0) value represents the last item in the field.
 - **End.** Designates the ending point of the index range. A positive number represents an absolute position in the range, while a negative number represents an offset from the last item. A zero (0) value represents the last item in the field.
 - **Input Mask.** Describes the format of the date as input from the original data to be used as a mask. This parameter only appears when the `FormatDate` formatter class is selected. The following characters can be used as accepted values:
 - yy = Two digit year (10)
 - yyyy = Four digit year (2010)
 - M = Single digit month (1)
 - MM = Two digit month (01)
 - MMM = Short form month name (Jan)
 - MMMM = Long form month name (January)
 - d = Single digit day of month (9)
 - dd = Two digit day of month (09)
 - E = Short form day of the week (Tue)
 - EEEE = Long form day of the week (Tuesday)For instance, a sample mask for 31/01/2010 would be represented by `dd/MM/yyyy`.
 - **Output Mask.** Describes the format of the date returned by the formatter to be used as a mask. This parameter only appears when the `FormatDate` formatter class is selected. The following characters can be used as accepted values:
 - yy = Two digit year (10)
 - yyyy = Four digit year (2010)
 - M = Single digit month (1)

MM = Two digit month (01)

MMM = Short form month name (Jan)

MMMM = Long form month name (January)

d = Single digit day of month (9)

dd = Two digit day of month (09)

E = Short form day of the week (Tue)

EEEE = Long form day of the week (Tuesday)

For instance, a sample mask for 31/01/2010 would be represented by dd/MM/yyyy.

- **Character to Replace.** Indicates the character(s) to replace. This parameter only appears when the ReplaceCharacter formatter class is selected.
- **Replacement Character.** Indicates the character(s) to use when replacing the character(s) specified in Character to Replace parameter. If no replacement character is specified, the input character is simply removed. This parameter only appears when the ReplaceCharacter formatter class is selected.
- **PROVIDER.** Indicates that the alternate text contains settings for specifying a reference to an instance of an AlternateTextProvider as defined by the TextProviders collection. This is useful when reusing large chunks of static text across all documents in a large job, as it allows you to insert the section as an object, saving time and disk space in compiling the output PDFs, such as those used for customer billing. An object is more efficient to insert into a PDF, but then it is not readable by screen reader technologies. By using a TextProviderReference to instruct screen readers where the text content is located, a screen reader can read the section from the source text without impacting the structure and efficiency of the PDF.
 - **Instance Name.** References the Instance Name of an AlternateTextProvider as defined in the TextProviders collection. On the Alternate Text dialog, you can select **Browse Providers**, , to bring up the Providers dialog where you can configure your provider settings. For more information, see [“Managing Alternate Text Providers” on page 321](#).
 - **Properties.** Controls custom properties which, if present, are passed into the providers as arguments. The properties defined here are specific to the MarkupItem associated with this ProviderInstance reference.
- **DISABLED.** Turns off alternate text for images in the output.
- **CS_ALT_TEXT_MANAGER.** Retrieves the approved alternate text used for images from the Alternate Text Manager console.
- **IMAGE_SOURCE.** Specifies that the Constant Print image’s alternate text setting in the DOM properties is used.

When auto detect is called, any tagged elements found in the container are pulled inline along with their associated DOM structure to the auto detection node in order

to maintain the flow's reading order. Any elements that are not already tagged will be added to the DOM structure with their detected types.

To review or edit additional properties, click on another tab in the XFT Container Properties window. When you have completed making your selections, click **OK** to accept your changes or **Cancel** to discard your changes and exit the dialog window.

XFT Containers Auto Detect Option Properties

The auto detect options for XFT Container help you control how page elements are detected within your documents. By taking the time to fine tune these options about your sections, the auto detection tool can return better results from your document. Multiple option sets can be created to ensure an option set is readily available to fit the criteria for variations in page design of sections within the same document. The Default auto detect option set is used by default and cannot be deleted. The following parameters are available to be configured for an auto detect options set:

- **JoinUnderRatio**. Specifies the ratio of font height to the next line within the same paragraph.
- **JoinDiffFonts**. Denotes whether your section contains instances of consecutive lines with different fonts joining within the same paragraph. By default, this is disabled.
- **InTableColGap**. Defines the minimal amount of space, in 72000 DPI, between columns within a table.
- **AllTableRowsHaveNumbers**. Indicates whether all table rows possess at least one column with numeric data, which can be useful in keeping cells together until there is a cell with a numeric value. By default, this is disabled.
- **TableStartSameFont**. Indicates whether all instances of the first row in tables use the same font. By default, this is disabled.
- **TableMaxLinesToStart**. Denotes the maximum number of lines that can appear at the start of a table. If there is no maximum, enter 0.
- **BoldFontMeansH1**. Indicates whether an isolated bold line should be tagged as H1 (level one heading). By default, this is enabled.
- **TextToTag**. Contains a list of text constants and patterns to be marked with a predefined tag type. If multiple TextToTag sets are enabled concurrently, you must observe the order in which the sets appear in your list since the options from the first matching set is used.
 - **IsActive**. Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType**. Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase**. Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.

- **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
 - **TagType.** Indicates the type of tag to apply to data matching your specified value.
- **FontCompare.** Contains options for comparisons based on font names, styles, and point sizes.
 - **Fontname.** Denotes the regular expression name of the font typeface such as ***Bold** or **Arial-Bold**. If no value is provided, then the font name check is not performed.
 - **Styles.** Contains a comma separated list of acceptable font styles, represented by letters. You can enter **B** for Bold, **I** for Italic, **BI** for Bold-Italic, or **N** for Normal (neither bold or italic). For example, using the value “**B,BI**” would match fonts using the bold or bold-italic styles while ignoring normal and italic styles. If no value is provided, all styles are matched.
 - **PtSizes.** Contains a comma separated list of decimal numbers or number ranges to indicate font point sizes. For example, “**10.3,11-15**” would match fonts with a point size of 10.3 or anything between 11 and 15. If no value is provided, all point sizes are matched.
- **SingleLineOnly.** Designates whether only single lines (lines not joined into a paragraph) are processed. By default, this is disabled.
- **SegmentToCompare.** Denotes the portion of the paragraph or line to compare against. You can select from the following options:
 - **FIRST_LINE.** Indicates that only text from the first line of the paragraph is compared.
 - **FIRST_FONT.** Indicates that only text from the first font is compared when a paragraph containing multiple fonts is encountered.
 - **FULL_PARA.** Indicates that text from the whole paragraph is compared.
- **SegmentToTag.** Denotes the portion of the paragraph or line to tag. You can select from the following options:
 - **FIRST_LINE.** Indicates that the first line is tagged as indicated by the TagType parameter while the remainder is marked with the P (paragraph) tag.
 - **FIRST_FONT.** Indicates that the text in the first font is tagged as indicated by the TagType parameter while the remainder is marked with the P (paragraph) tag.
 - **FULL_PARA.** Indicates that the whole paragraph is tagged as indicated by the TagType parameter.

- **FalseTableStart.** Contains a list of patterns and constant text to compare against as tables are detected. When a match is found, the table start is ignored in order to provide an override for incorrectly detected tables.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
- **CombineSuperscript.** Denotes whether to join two base lines as one if the two base lines join in a manner that is likely to be superscript text. By default, this is enabled.
- **HandleVectorCelledTables.** Designates whether tables with cells that are contained within vector lines are processed with the vector lines intact. (Note that enabling this setting may affect the precision of the cell detection logic as not all vector lines are drawn in a manner that is completely compatible.)
- **ListLBLFonts.** Specifies the fonts used in list labels. Any text using these fonts following a bullet character is included with the list's LBL tag. If any of these values end with an asterisk (*), a BeginsWith test is performed.
- **BulletChars.** Contains a list of characters or strings that indicate a list is starting. There are two special values which may be used:
 - **#.** (pound period), which will match 1, 2, or 3 digit numbers followed by a period.
 - **@)** (at-sign right-parentheses), which will match 1, 2, or 3 lower case letters followed by a right parentheses.
- **BreakOnObject.** Designates whether paragraph joins are stopped when a vector or image is between two lines of text. This setting may have an effect on how tables are recognized.
- **NotesReferences.** Contains options for identifying notes and references within your documents.
 - **IncludeList.** Designates any patterns to include during note and reference processing.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.

- **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
- **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.
- **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
- **Value.** Specifies the pattern or constant text to compare against.
- **StyleType.** Indicates the style of text for comparison. You can choose from **SUPERSCRIPT**, **NON_SUPERSCRIPT**, and **ANY**.
- **ExcludeList.** Designates any patterns to exclude from note and reference processing.
 - **IsActive.** Denotes whether this particular option set is used. This allows users to set up multiple option sets at a time, enabling and disabling individual sets as they are required.
 - **TextType.** Indicates the type of text to examine. You can choose from **ENDSWITH**, **CONTAINS**, **STARTSWITH**, **REGEX**, and **CONSTANT**.
 - **IgnoreCase.** Designates whether to ignore the letter case for all text. By default, this is disabled. If you are using REGEX setting for TextType, this setting is ignored.
 - **MoreUpperNoPunctuation.** Denotes whether the line of text to locate must contain more words starting with upper case letters than lower case letters and does not end with a period or question mark character.
 - **Value.** Specifies the pattern or constant text to compare against.
 - **StyleType.** Indicates the style of text for comparison. You can choose from **SUPERSCRIPT**, **NON_SUPERSCRIPT**, and **ANY**.
- **Name.** Specifies the unique name assigned to this set of options.

6.15.3.15 Adding a Group

You can use XFT groups to group XFT fields of the same type together, allowing the output to logically flow from one field to another within the group. Grouping also provides better control over accessible output and XML output in situations where data is organized in columns.

To add an XFT group:



Note: You must have first defined one or more sections before you can add a group.

1. In the XFT Design Pane, select the section to which you want to add the XFT group.

2. From the XFT window toolbar, click **Add Group**, .
3. From the menu that appears, select one of the following to determine the group type:
 - Add Text Field Group
 - Add Table Field Group
 - Add List Field Group



Note: Once defined, the group type cannot be changed.

6.15.3.15.1 Adding a Text Field Group

In the **Add Text Field Group** dialog box, there are two separate tabs where you can configure properties for the group:

- “Text Field Group Properties General tab” on page 480
- “Text Field Group Properties DOM Properties tab” on page 480

Text Field Group Properties General tab

The **General** tab specifies general properties for the text field group. It has the following fields:

- **Group Name.** Specifies a name for your text field group. The name must be unique within the project.
- **Enable.** Indicates that text field group processing is enabled. If it is disabled, the engine does not look for the presence of text field groups.
- **Enable DOM Properties.** You must select this option to make the DOM Properties tab available. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

To review or edit additional properties, click on another tab in the Text Field Group properties dialog box. When you have completed making your selections, click **OK** to accept your changes.

Text Field Group Properties DOM Properties tab

The **DOM Properties** tab specifies Document Object Model configuration for your text field group. It has the following fields:

- **Tag Type.** Defines the tag type for this node element. Applicable tag types can be selected from the drop-down box. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner.

- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the drop-down box or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text.** Defines the actual text for this item that is used by screen readers.
- **Tag Only Once.** Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos, which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.
- **Detect Columns.** Indicates that columns are to be detected among text elements. By default, this is disabled.

To review or edit additional properties, click on another tab in the Text Field Group properties dialog box. When you have completed making your selections, click **OK** to accept your changes.

6.15.3.15.2 Adding a List Field Group

In the **Add List Field Group** dialog box, there are two separate tabs where you can configure properties for the group:

- “List Field Group Properties General tab” on page 481
- “List Field Group Properties DOM Properties tab” on page 482

List Field Group Properties General tab

The **General** tab specifies general properties for the list field group. It has the following fields:

- **Group Name.** Specifies a name for your list field group. The name must be unique within the project.
- **Enable.** Indicates that list field group processing is enabled. If it is disabled, the engine does not look for the presence of list field groups.
- **Enable DOM Properties.** You must select this option to make the DOM Properties tab available. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

To review or edit additional properties, click on another tab in the List Field Group properties dialog box. When you have completed making your selections, click **OK** to accept your changes.

List Field Group Properties DOM Properties tab

The **DOM Properties** tab specifies Document Object Model configuration for your list field group. It has the following fields:

- **Tag Type.** Defines the tag type for this node element. Applicable tag types can be selected from the drop-down box. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner.
- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the drop-down box or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text.** Defines the actual text for this item that is used by screen readers.
- **Tag Only Once.** Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos, which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.

To review or edit additional properties, click on another tab in the List Field Group properties dialog box. When you have completed making your selections, click **OK** to accept your changes.

6.15.3.15.3 Adding a Table Field Group

In the **Add Table Field Group** dialog box, there are two separate tabs where you can configure properties for the group.

- “Table Field Group Properties General tab” on page 482
- “Table Field Group Properties DOM Properties tab” on page 483

Table Field Group Properties General tab

The **General** tab specifies general properties for the list field group. It has the following fields:

- **Group Name.** Specifies a name for your table field group. The name must be unique within the project.
- **Enable.** Indicates that table field group processing is enabled. If it is disabled, the engine does not look for the presence of table field groups.
- **Enable DOM Properties.** You must select this option to make the DOM Properties tab available. Additionally, you must also enable DOM properties for any parent XFT elements; the status of DOM properties for child XFT elements is inherited from the parent element and consequently, is unavailable to the child element unless DOM properties are enabled on all related parent XFT elements.

To review or edit additional properties, click on another tab in the Table Field Group properties dialog box. When you have completed making your selections, click **OK** to accept your changes.

Table Field Group Properties DOM Properties tab

The **DOM Properties** tab specifies Document Object Model configuration for your table field group. It has the following fields:

Basic

- **Tag Type.** Defines the tag type for this node element. Applicable tag types can be selected from the drop-down box. It is important to select an appropriate tag type since the resulting tag structure helps direct screen reader applications through a document in a logical manner.
- **Tag Name.** Specifies the name used to identify a particular node/structural element found in the document. The tag name is optional, but if no value is provided then the name used on the General tab is also used for the tag name.
- **Language.** Designates the default language for the document as defined by RFC 4646 language subtag registry. You can either select the language from the drop-down box or select the autocomplete suggestion as you manually type the language tag. This information overrides the document language for the scope of the node.
- **Actual Text.** Defines the actual text for this item that is used by screen readers.
- **Tag Only Once.** Denotes that a repeating element should only be tagged once on its first appearance, and then artifact any subsequent occurrences of the element. As a result, only the first instance of the selected element will be read by the screen reader, and subsequent occurrences will be ignored. This is useful for elements such as company addresses or logos, which may repeat on multiple pages of billing statements. When a parent DOM element is an artifact, then the child element must also be an artifact. By default, this option is disabled.
- **Allow Empty Tags.** Designates that tags are created for elements even if no content exists within the tags. By default, this is enabled.

Advanced

- **Alternate Cell Text.** Details an array of Alternate Text definitions with each applied to a specific range of rows and columns. The number of entries is shown on the **DOM Properties** tab and new array entries can be added by clicking the ellipsis, , beside the field.
- **Row for Column Header.** Indicates the row to be used as the column header, where **1** represents the first row of a table. If the table has no column headers, enter **0** as the value.
- **Column for Row Header.** Indicates the column to be used as the row header, where **1** represents the first column in the table. If the table has no row headers, enter **0** as the value.
- **Table Summary.** Specifies a brief description of the table.
- **Table Header.** Defines table headers, including row headers, column headers, and table header cells. Click the ellipsis, , to enter the values.

To review or edit additional properties, click on another tab in the Table Field Group properties dialog box. When you have completed making your selections, click **OK** to accept your changes.

6.15.3.15.4 Managing XFT Groups

This section discusses management of XFT groups.

Adding XFT Fields to a Group

Once you have defined a group, you can add XFT fields of the same type to it:

In the XFT Design Pane, select the group to which you want to add a field.

- If it is a text field group, click the **Add Text Field** button in the XFT window toolbar, or right-click the group and select **Add Text Field** from the context menu. For the remaining steps on configuring your text field, see [“Adding a Text Field” on page 393](#).
- If it is a table field group, click the **Add Table** button in the XFT window toolbar, or right-click the group and select **Add Table** from the context menu. For the remaining steps on configuring your table field, see [“Adding a Table” on page 432](#).
- If it is a list field group, click the **Add List Field** button in the XFT window toolbar, or right-click the group and select **Add List Field** from the context menu. For the remaining steps on configuring your list field, see [“Adding a List Field” on page 417](#).



Note: You cannot add an XFT field to a group of a different XFT field type.

Editing a Group

You can apply the following edits to an XFT group:

- **Renaming the group.** Double-click the group in the XFT Design Pane to launch the **Edit Field Group** dialog, where you can edit the **Group Name** value.
- **Disabling the group.** Double-click the group in the XFT Design Pane to launch the **Edit Field Group** dialog, where you can clear the **Enable** check box.
- **Editing the items within a group.** Right-click on an item within a group to perform any of the following actions: edit, delete, hide/show, lock/unlock, and change its order within the group tree structure.



Note: Once defined, the group type cannot be changed.

Deleting a Group

To delete a group:

1. In the XFT Design Pane, right-click the group you want to delete and select **Delete** from the context menu.
2. In the **Remove Group** window, confirm the deletion by clicking **Yes**.
The group and its contents are removed.

6.15.3.15.5 Group Properties

The following properties apply to the **Group** parameter and can be configured in the **Properties** window:

- **GroupType.** Specifies whether the group type is text, table, or list. This property is defined when the group is created and cannot be changed.
- **Name.** Indicates the name of the group.
- **TextFields.** Specifies the properties for text fields. For descriptions of text field parameters, see [“Adding a Text Field” on page 393](#).
- **TableFields.** Specifies the properties for XFT tables. For descriptions of table field parameters, see [“Adding a Table” on page 432](#).
- **ListFields.** Specifies the properties for list fields. For descriptions of list field parameters, see [“Adding a List Field” on page 417](#).
- **Enabled.** Determines whether the group is enabled.

6.15.3.16 Adding a Text Locator

A text locator is a text-based search pattern that can be searched for in a specified area of a document. Locators are typically used for identifying the start and end points in a document and may not necessarily exist on an input page. When using locators, multiple instances of the same text or image object field on a single page can be found.

To add a text locator:

1. In the Design pane of the XFT tab, right-click the **Locators** node and from the context menu that appears, click **Add Text Locator**, .
2. Move the cursor into the Preview pane and the cursor itself changes from an arrow to crosshairs. Using the crosshair cursor, draw a bounding box around the area you want to establish as your locator.
3. Using the **General** and **Extraction Settings** tabs in the **XFT Text Locator Properties** dialog box, configure your text locator and when you are finished, click **OK**.

You can also duplicate an existing locator in your project using the **Clone** feature. Cloning a locator allows you to enter a new name and copies the existing locator's properties. To clone a locator:

1. In the Design pane, expand the **Locators** node and select the text locator you want to clone.
2. Right-click the locator and from the context menu that appears, click **Clone**.
3. In the **XFT Text Locator Properties** box, the properties from the existing locator are automatically populated. You must type a **Name** to identify the new locator. When you are finished, click **OK**.

6.15.3.16.1 Text Locator Properties General Tab

The General tab defines the locator and establishes parameters for the search operation. It has the following fields:

- **Name.** Indicates the name for your text locator.
- **Search Area.** Indicates the area to search, whether it is the **Whole Page** or only the area you **Defined** when drawing the bounding box. You may also customize the boundaries of the search by entering the coordinates in the fields for **Defined Search Area**.
- **Search Pattern.**
 - **Literal Text.** Indicates the specific string of text to search for. This value is case sensitive.
 - **Regular Expression.** Indicates the regular expression you want to search for.

6.15.3.16.2 Text Locator Properties Extraction Settings Tab

The Extraction Settings tab specifies the text direction and thresholds for extraction. It has the following fields:

- **Text Print Direction.** Specifies the orientation in which the text prints or the degree of rotation on the page. From the drop-down menu, you can select from **Portrait** (0°), **Inverse Landscape** (90°), **Inverse Portrait** (180°), and **Landscape** (270°).
- **Thresholds settings.** Indicates the threshold settings to use. You may opt to use the default settings or you can click the ellipsis button to display the **Select Threshold** dialog window to **Add**, **Edit**, or **Remove** a threshold you have defined. For more information, see [“Adding a Threshold” on page 462](#).
- **Include Form Text.** Indicates that text embedded in forms should be included in the search.
- **Font Filter.** Indicates which font filters should be used with your locator to find text in a particular font.
- **Filter Type.** Specifies whether the list of the application’s font filters are included or excluded. From the list, select **Include List** or **Exclude List**.

6.15.3.17 Adding an Image Locator

An image locator is a graphic-based search pattern that can be searched for in a specified area of a document. Locators are typically used for identifying the start and end points in a document and may not necessarily exist on an input page. When using locators, multiple instances of the same text or image object field on a single page can be found.

To add an image locator:

1. In the Design pane of the XFT tab, right-click the **Locators** node and from the context menu that appears, click **Add Image Locator**, .
2. Move the cursor into the Preview pane and the cursor itself changes from an arrow to crosshairs. Using the crosshair cursor, draw a bounding box around the area you want to establish as your locator.
3. Using the **General** and **Extraction Settings** tabs in the **XFT Image Locator Properties** dialog box, configure your text locator and when you are finished, click **OK**.

You can also duplicate an existing locator in your project using the **Clone** feature. Cloning a locator allows you to enter a new name and copies the existing locator’s properties. To clone a locator:

1. In the Design pane, expand the **Locators** node and select the image locator you want to clone.
2. Right-click the locator and from the context menu that appears, click **Clone**.

3. In the **XFT Image Locator Properties** box, the properties from the existing locator are automatically populated. You must type a **Name** to identify the new locator.

When you are finished, click **OK**.

6.15.3.17.1 Image Locator Properties General Tab

The General tab defines the locator and establishes parameters for the search operation. It has the following fields:

- **Name**. Indicates the name for your image locator.
- **Search Area**. Indicates the area to search, whether it is the **Whole Page** or only the area you **Defined** when drawing the bounding box. You may also customize the boundaries of the search by entering the coordinates in the fields for **Defined Search Area**.
- **Search Conditions**. Contains settings related to the conditions used to locate an image. More than one search condition can be enabled simultaneously by selecting the respective check boxes.
 - **Image Hash Key**. Identifies the checksum for the image content. If the bounding box you drew for the current image locator contains other image locators within it, their checksums will be available for selection in the dropdown menu when the Image Hash Key check box is enabled.
 - **Image Width**. Denotes the width of the image using XIF coordinates (72000 DPI). The default setting of **0** indicates any widths are accepted.
 - **Image Height**. Denotes the height of the image using XIF coordinates (72000 DPI). The default setting of **0** indicates any heights are accepted.
 - **Image Type**. Designates the type of d2e image used to wrap the locator image from the parser according to the color space. You can choose from **ALL**, **ONEBIT**, **RGB**, **CMYK**, **INDEXED**, **GRAY**, **DUOTONE**, **COMPOSITE_CMYK**, or **COMPOSITE_RGB**.

6.15.3.17.2 Image Locator Properties Extraction Settings

The Extraction Settings tab specifies whether embedded forms within the XifPage are open for extraction. It has the following fields:

- **Include Form Image**. Determines whether the locator searches the image elements that represent embedded forms within the XifPage.

6.15.3.18 Adding a Graphic Locator

A graphic locator is a graphic-based search pattern that can be searched for in a specified area of a document. Locators are typically used for identifying the start and end points in a document and may not necessarily exist on an input page. When using locators, multiple instances of the same text or image object field on a single page can be found.

To add a graphic locator:

1. In the Design pane of the XFT tab, right-click on the **Locators** node and from the context menu that appears, select **Add Graphic Locator**, .
2. Move the cursor into the Preview pane and the cursor itself changes from an arrow to crosshairs. Using the crosshair cursor, click and drag to draw a bounding box around the area you want to establish as your locator.
3. Using the **General** and **Graphic Object Models** tabs in the **XFT Graphic Locator Properties** dialog box, configure your graphic locator and when you are finished, click **OK**.

6.15.3.18.1 Graphic Locator Properties General Tab

The General tab defines the locator and establishes parameters for the search operation. It has the following fields:

- **Name.** Indicates the name for your graphic locator.
- **Search Area.** Indicates the area to search, whether it is the **Whole Page** or only the area you **Defined** when drawing the bounding box. You may also customize the boundaries of the search by entering the coordinates in the fields for **Defined Search Area**.

6.15.3.18.2 Graphic Locator Properties Graphic Object Models Tab

The Graphic Object Models tab designates the graphic object models for the locator to analyze. You must specify the graphic object model(s) to use by clicking  and in the **Add Graphic Object Model** dialog box that appears, selecting a model from the list of your existing graphic object models.

6.15.3.19 Adding a Graphic Object Model

Graphical elements like check boxes or option buttons are occasionally laid out on the page in very close proximity to text, which can cause difficulty in defining the required text locators to isolate and extract these objects. Graphic elements that use vectors, characters, or a combination of both, such as a check box containing a check mark, can also cause problems since none of these aspects are easily recognized with regular text locators. To minimize these issues, graphic object models can be established so that the graphic is clearly defined and can be located without the need to look for text near them to identify the graphic object. It is recommended graphic object models be used only after all other methods of capturing page elements have been attempted and prove unsuccessful.

Fully configuring a graphic object model for use in your projects in a multi-step process. Once a graphic object model is created, you must set up its supporting aspects so that it can be used in your project. For more information on the process, see [“Using Graphic Object Models to Identify Content” on page 491](#).

To integrate a graphic object model:

1. In the Design pane of the XFT window, right-click on the **Graphic Object Models** node and in the context menu that appears, select **Add Graphic Object Model**.
2. When you move the cursor over the Preview pane, the cursor changes from an arrow to a crosshair. Using the crosshair cursor, draw a bounding box around the content you want to define as a graphic object model.
3. Complete the fields in the **XFT Graphic Object Model Properties** dialog box and when you are done, click **OK**.

6.15.3.19.1 Graphic Object Model Properties General Tab

The **General** tab specifies general properties for the specific graphic object model. It has the following fields:

- **Name.** Specifies a name for the graphic object model. The name must be unique within the project as it is used as a key reference point by graphic locators.
- **Enabled.** Indicates that processing for this graphic object model is enabled.
- **Defined Search Area.** Indicates the defined area that covers the selected object element in its entirety.
 - **Left, Top, Width, Height.** Specifies the boundaries of the defined search area.
- **Model Raster.** Denotes the attributes of the selected model raster, which are used to identify the model.
 - **Version.** Denotes the internal code for maintenance purposes.
 - **Hash Key.** Indicates the unique checksum of the rasterized graphic object content, which is used to identify the model.
 - **Width.** Denotes the width of the image using XIF coordinates (72000 DPI).
 - **Height.** Denotes the height of the image using XIF coordinates (72000 DPI).
 - **Segments Info.** Contains a description of the sequence and types of graphic elements in the model raster.
 - **Reload.** Retrieves the fixed graphical attributes about the defined object. Initially, the model raster attributes are blank and must be manually loaded. If the locator or extraction settings are subsequently modified, then you must also reload the attributes to update the model.

6.15.3.19.2 Graphic Object Model Properties Extraction Settings Tab

The **Extraction Settings** tab specifies the types of page elements and associated properties to extract from the graphic object model. It has the following fields:

- **Object Inclusion Threshold.** Denotes the percentage of an object (vector, image, etc.) that must be inside the field's bounding box area for it to be captured. By default, 100% of the object must be inside this area, however, there are cases where the object is not completely within the boundaries, such as a check mark in a check box.
- **Include Forms Objects.** Indicates whether the form objects inside the selected area are included in the model.
- **Include Vectors.** Indicates whether the graphic vector objects inside the selected area are included in the model.
- **Include Images.** Indicates whether the image objects inside the selected area are included in the model.
- **Image Type.** Specifies the type of d2e image, according to the color space, that wraps the image from the parser output. You can choose from ALL, ONEBIT, INDEXED, RGB, YCBCR, CMYK, GRAY, DUOTONE, COMPOSITE_CMYK, COMPOSITE_RGB, COMPOSITE_ONEBIT, COMBINED, CALRGB, ICCBASED, STENCIL_MASK.
- **Include Characters.** Indicates whether the character objects inside the selected area are included in the model.
- **Character Orientation.** Identifies the element's orientation on the page using XIF coordinates. You can choose from 0, 90, 180, or 270 degree orientations.

6.15.3.20 Using Graphic Object Models to Identify Content

A graphic object model is used to explicitly define a graphic so that graphic objects can be located without the need to look for nearby text to identify them. When using graphic object models in an XFT document to identify content, some additional aspects must be configured and the following procedure provides an overview of the process.

To use a graphic object model to identify content from your document:

1. Create a graphic object model. A graphic object model must be created to identify the graphic content on the page. For more information, see [“Adding a Graphic Object Model” on page 489](#).

Any graphic object models you create appear under the **Graphic Object Models** node in the **Design** pane of the XFT window.

2. Create a graphic locator. A graphic locator defines the search conditions used to detect specific objects on the page to use as a reference point for XFT. While configuring the graphic locator, you must specify the graphic object model(s) to use. For more information, see [“Adding a Graphic Locator” on page 489](#).

Any graphic locators you create appear under the **Locators** node in the **Design** pane of the XFT window.

3. Create a field definition. Once the graphic locator has been created, it can be used in most standard field definitions to identify the start and end points on the page.

When creating the field definition, you must specify use of a graphic locator in the **Identifiers** section on the **Identification** tab for your respective field definition. Depending on your document, it may also be suitable to use the dynamic positioning properties on the field definition's respective **Positioning** tab to refine the locator's position or if there is more than once instance of the graphical object on the page that you want to locate, turn on the **Enable Multiple Instances Per Section** option.

Once these elements are set up, you can verify the identified content from the page according to your configured locators and field definitions in the **Results** pane.



Note: In situations where there are multiple instances of a graphic object on a page, you may discover that some instances are being found, whereas other instances of the object are not detected. This is most likely due to the objects appearing similar, but they are not actually identical. To circumvent this issue, you can create another graphic object model for the objects not being detected.

6.15.3.21 XFT Covering Letter Example

In the following example, a covering letter has been created with these rules:

- It always starts at the top of a new page, and it is at least a single page in length;
- It contains the policyholder's name and address, followed by the date of the letter;
- The next piece of information is the policyholder's policy number which immediately follows the words "Policy Number:";
- Then comes the salutation and the letter content;
- The end of the covering letter is indicated by the presence of the Agent details, starting with the phrase "Your Agent is".



Figure 6-34: Covering Letter Example

Steps to Implement

To define the sections, fields, locators, and identifiers needed to extract data from this document, and to correctly route output, the following steps are taken:

1. **Define the Section** to start below the logo and above the company information. This also excludes headers and footers. The bounding box is drawn around this area when the section is added.
2. **Define a Locator** called `XenosMutual` over the "Xenos Mutual Insurance" text at the top of the page.
3. **Define the Section** start identifier for the `XenosMutual` locator (created above) and give this the confidence factor to start the section on its own.
4. **Define a Locator** called `PolicyNumber` over the text "Policy Number:"; make the search area about three lines above and below the current position.
5. **Define a Text Field**, named `PolicyNumber`, whose location is determined by the `PolicyNumber` locator; specify that the field is offset to the right of the locator.
6. In the same fashion, **define Locators** for the 'Agent Name' field and 'Your Agent is' field.
7. **Define the Section End Identifier** as the 'Your Agent is' Locator.

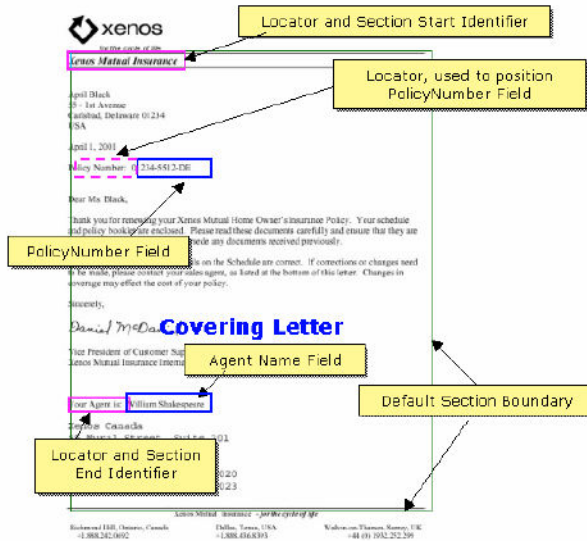


Figure 6-35: Covering Letter with Locators Defined

When you run the job, the information will be extracted as you have specified.

6.15.4 XIF Text Element Properties

The XIF Intermediate Format allows for font and character translation to be performed.

Open the **XIF Text Element** dialog window from the **Run** tab of the **XFT** window by doing the following:

1. Select an **XIF Element** from the **XFT tree** in the XFT window's **Run** tab.

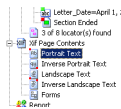


Figure 6-36: XIF Element in the XFT Tree

2. Double-click a **text element** in the table of extracted data below the XIF tree.

Text	Left
for	113790
the	120150
cycle	144950
of	170600
the	181650
Xenos	72000
Mutual	111190
Insurance	157400
April	72000
Black	100050
SS	72000
-	67100
1st	94000

Figure 6-37: Extracted Data

The **XIF Text Element** dialog has three tabs:

- Font Information Tab
- Character Information Tab
- Positional Information Tab

6.15.4.1 Font Information Tab

You can set the following font attributes for the selected XIF text element:

- **Font.** Click the **Map Font** button to launch the **Add AFP Charset** dialog. For more information, see [“Cell Properties General Tab” on page 453](#).
- **Font Attributes.** Click the **Map Family** button to launch the **Edit AFP Typeface** dialog. For more information, see [“Adding an AFP Typeface Input Font” on page 511](#).
- **Output Font.** The output font is listed in the text box. Click **Edit Output Font** to launch the **Edit Output Font** dialog. For more information, see [“Editing an AFP Output Font Family” on page 539](#).

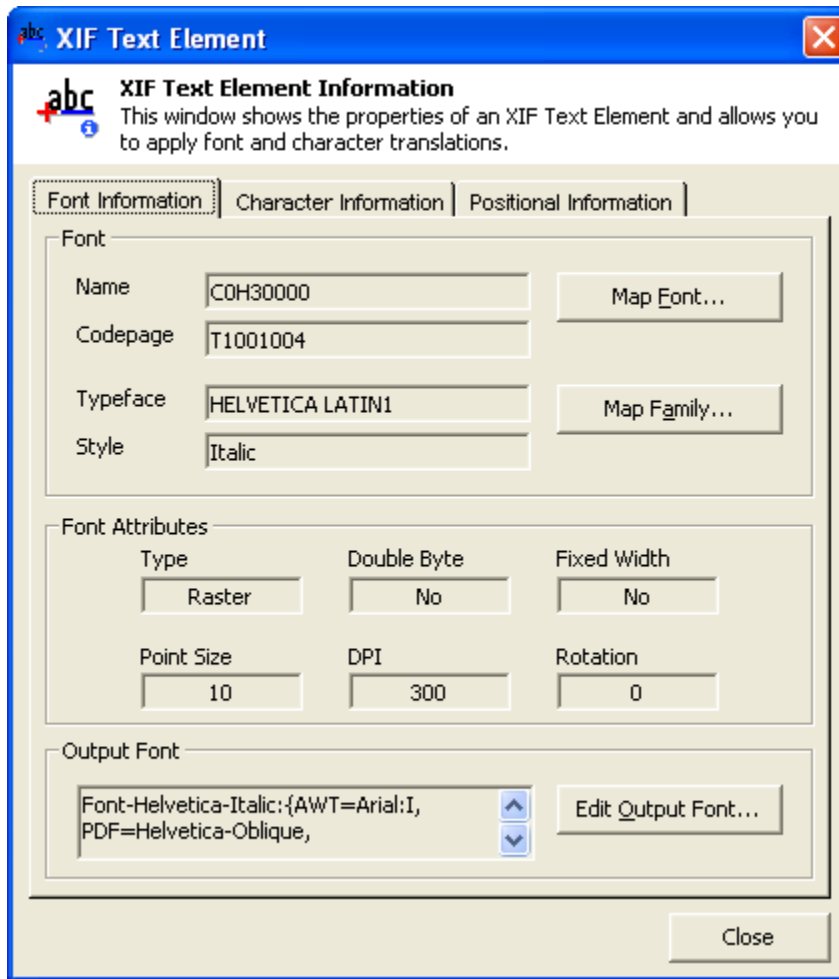


Figure 6-38: XIF Text Element – Font Information tab

6.15.4.2 Character Information Tab

You can set and view the following character attributes for the selected XIF text element:

- **Text.** Displays the XIF text found and extracted from the XFT field.
- **Characters.** Click the **Map AFP Glyph** button to launch the **Edit AFP Glyph Mapping** dialog. For more information, see *“Editing a Glyph Mapping” on page 560.*

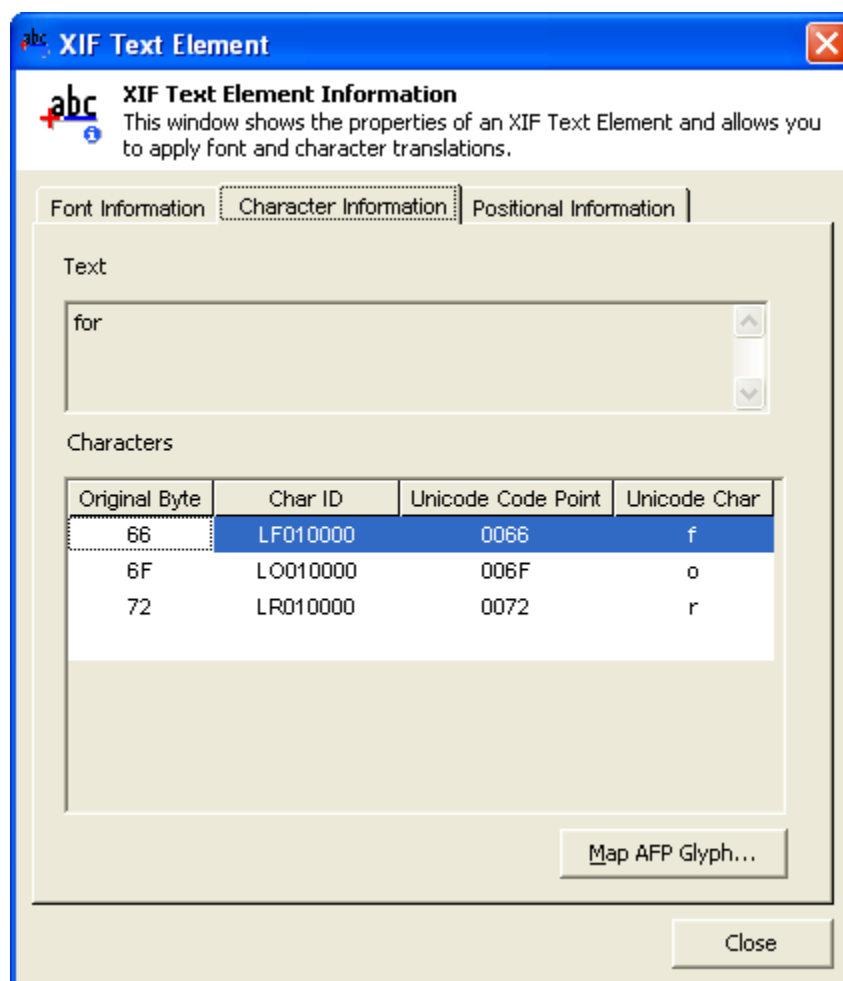


Figure 6-39: XIF Text Element - Character Information tab

6.15.4.3 Positional Information Tab

You can view the following positional information for the selected XIF text element:

- **Print Position.** Displays the value of various print related attributes of the XIF text.
- **Bounding Box.** Displays the position of the field's bounding box sides.

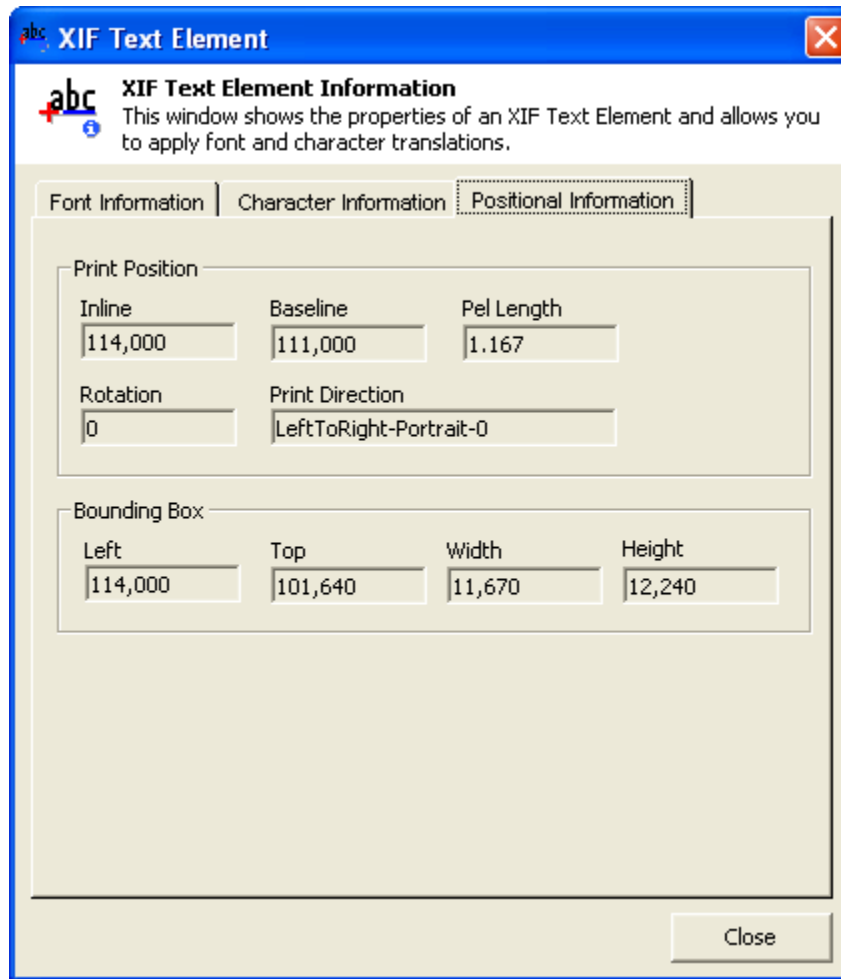


Figure 6-40: XIF Text Element - Positional Information tab

6.16 XFT Writer Component



The Output Transformation Engine **XFT Writer** component writes information collected by the XFT Field Technology component in a predefined XML format. The XML describes the start and end of XFT documents and sections, as well as data extracted from XFT fields or tables.

6.16.1 XFT Writer Wizard

The **XFT Writer Wizard** simplifies the process of setting the component's parameters. You can also manually set the parameters on the **Properties** tab.

The XFT Writer Wizard has the following commands:

- **Add.** Adds a new output file configuration.
- **Edit.** Edits an existing output file configuration.
- **Remove.** Removes an existing output file configuration.
- **Up.** Moves the selected output file configuration higher in the list to give it a higher priority when processing or writing.
- **Down.** Moves the selected output file configuration lower in the list to give it a lower priority when processing or writing.
- **Include locations.** Specifies whether location names are included in output index files.

The XFT Writer Wizard streamlines the following features:

- Adding or Editing an Output File
- Adding or Editing an Output Index File

The following parameter can be set on the **Properties** tab:

- **OutputStyle.** Controls the XML style of the output file. Select **XftStructure** to organize the data in the output file according to the XFT structure. Select **Descriptive** to organize the data in the output file in a format similar to the input file, using more descriptive tag names (for example, *productnumber*, *quantity* or *unitprice*) When using the Descriptive method, you must adhere to certain validation rules: XML element names cannot start with a number, punctuation character, or the letters XML.

6.16.1.1 Adding or Editing an Output File

You must set the path and file name for the output file that you want generated.

To configure the output file:

1. In the **XFT Writer wizard**, do one of the following:
 - To set up a new output file, click **OutputFile** and then the **Add** button. The **Select Resource Path/File** dialog appears.
 - To edit an existing output file configuration, select the output file settings you want to change and click the **Edit** button. The **Select Resource Path/File** dialog containing your current settings appears.

2. The options in the **Select Resource Path/File** dialog allow you to set your output file options. For more information, see [“Select Resource Path/File Window” on page 102](#).
3. When you are finished, click **OK**.

The new output file settings appear in the list on the XFT Writer wizard tab.

6.16.1.2 Adding or Editing an Output Index File

You must set the path and file name for the output index file that you want generated.

To configure the output index file:

1. In the **XFT Writer wizard**, do one of the following:
 - To set up a new output index file, click **FdODIndexFile** and then the **Add** button. The **Select Resource Path/File** dialog appears.
 - To edit an existing output index file configuration, select the output index file settings you want to change and click the **Edit** button. The **Select Resource Path/File** dialog containing your current settings appears.
2. The options in the **Select Resource Path/File** dialog allow you to set your output index file options. For more information, see [“Select Resource Path/File Window” on page 102](#).
3. When you are finished, click **OK**.

The new output index file settings appear in the list on the XFT Writer wizard tab.

Chapter 7

Font Correlation

The topics in this section discuss various aspects of font correlation.

7.1 Output Transformation Engine Font Mapping

Output Transformation Engine font mapping provides a powerful range of options for mapping or rendering incoming document fonts from printstreams into the desired output formats. These options include:

- Making a raster image of each word, which provides a high-quality rendition where no equivalent output font exists.
- Font mapping with control over each character's translation where each word (defined by blank delimited or non-printable characters) is placed precisely as it should be. The PDF Generator also supports mapping into embedded Type 1 fonts.
- A combination of raster images and font mapping where most characters of the font are mapped to an outgoing font but some special characters are turned into raster images.
- For PDF output only, handling custom fonts utilizing the .AFM and .PFB font files.

Output Transformation Engine works within Output Transformation Designer, a visual GUI, to more easily enable font mapping as well as mapping each character's glyph to Unicode.

7.1.1 Font Correlation Overview

In order to convert one data stream to another, there must be a common way to correlate the input format's fonts and colors to the output format's fonts and colors.

Font correlation is done using a **Font Correlation Table** (FCT). Understanding how the FCT works will help in building and maintaining the font correlation table.

When a parser processes a document, the text is represented through fonts. The parser determines which fonts are being used in the input file and finds the correlation in the FCT. The FCT determines how the conversion handles text within the input file font.

Output Transformation Designer helps you build, edit, and maintain the FCT to ensure your input file fonts are correctly mapped to output file fonts. FCT also stands for Fonts, Colors, and Translations.



Note: The terms **Master Font Table** (MFT), **Master Font Correlation Table** (MFCT), and **Font Correlation Table** (FCT) are used interchangeably.

When a parser runs, a font table is used to determine how each font is processed. Generally, the FCT relates each input font to the correct output font, but it may also control the color(s) used for each font, and special translations for mapping the code points in the input format to code points in the output format (or ASCII to EBCDIC).

For example, the FCT allows you to:

- Embed custom Type 1 fonts for PDF generation, using .AFM and .PFB files.
- Specify font, line, and image colors.
- Specify individual fonts or characters to be rasterized or imaged.
- Assign color on a font-by-font basis.



Note: You can use colors to debug a project in which text is misaligned. By assigning a font a unique color, it can be easily detected in the output. Problem fonts can be revealed in the output using this method.

- Correct character map differences.
- Separate different table sections used by different parsers and generators.

Output Transformation Designer comes with a default Master Font Correlation Table (`sample.d2emfct`) which can be changed to accommodate your input, or used as an example to aid you in viewing and proofreading your own table.

7.2 Master Font Correlation Table

Output Transformation Engine maps the incoming style of the font to an output font. For example, in converting AFP to PDF, should the glyphs be mapped to a Helvetica-Bold font, or to a Courier-Italic font?

Mapping the font style is not the only challenge. Each glyph must also be mapped to the appropriate outgoing glyph. This can be a difficult task, especially with a double-byte font containing 7000+ Asian language characters.

Output Transformation Engine utilizes a **Master Font Correlation Table** (MFCT) to give the user control over font style and glyph mappings.

OpenText Font Mapping

Font Mapping maps each incoming font typeface to the following:

- An output font or a family of fonts.
- Word images if no matching font is available.
- Controls ASCII/EBCDIC and special character translations.

Images can be made of individual characters when no matching character is available in the output font.

Font Style Mapping

The MFCT can map a font's style in two different ways:

- The **font's charset name** (for example, C0D0GT12).
- The **charset typeface name** (for example, GOTHIC).



Note: It is best to use the typeface name when possible, since this covers any and all fonts that share that typeface name. In the rare situation that the typeface name of the font is invalid or interferes with another set of fonts using the same typeface name, the font can be mapped individually by the charset name.

The MFCT supplied with Output Transformation Designer gives you the advantage of multiple font families being described within one table.

The MFCT may be easily shared between projects in Output Transformation Designer. Changes made to the master table will be reflected in all projects that use that MFCT, so be careful when you optimize.



Note: The terms **Master Font Table** (MFT), **Master Font Correlation Table** (MFCT), and **Font Correlation Table** (FCT) are used interchangeably in this document.

7.2.1 MFCT Tips

The following usage tips will help eliminate errors associated with a font correlation table:

- **First project.** An MFCT is included with the delivered Output Transformation Engine which should be used for the first project. OpenText recommends making a copy of this table and modifying the copy according to font mapping needs.
- **Table Size.** The more lines in the MFCT the more memory the job requires. Eventually, a table may become so large the job will not run efficiently. In this case, several font tables need to be built, each table containing the fonts used by a single project.

7.2.2 Managing an MFCT within a Project

The following options are available for administration of a Master Font Correlation Table with your project:

- In the Transformation Projects tab, right-click the MFCT element and from the shortcut menu that appears, you can:
 - **Add MFCT.** Adds a new MFCT to the project. The Select Resource Path/File dialog appears where you can select the MFCT you wish to add to your project.
 - **Save Active MFCT As.** Saves a copy of the current MFCT. Enter a file path and name in the dialog window that appears.
 - **Show Active MFCT.** Launches the MFCT Tab in the Development Window, showing the current MFCT.
 - **New Default MFCT.** Sets another MFCT as the default for your project. The Select Resource Path/File dialog appears where you can select the MFCT you wish to add to your project.
- In the Transformation Projects tab, right-click the MFCT file path entry and from the shortcut menu that appears, you can:
 - **Remove MFCT.** Removes the selected MFCT from your project.
 - **Edit MFCT Location.** Choose a different MFCT for your project. If you have made changes to the current MFCT, a dialog will be presented asking if you want to save or discard the changes to the current MFCT.

To open and edit the current MFCT, double-click the MFCT file path entry. If you make any changes to the MFCT, when closing it you will be presented with a dialog asking if you want to save or discard the changes. Choosing Save overwrites the current MFCT.

If you want to preserve the original unedited MFCT, but want to save your changes to a new MFCT location, right-click the MFCT Element and choose Save Active MFCT As. Change the filename in the Save dialog.

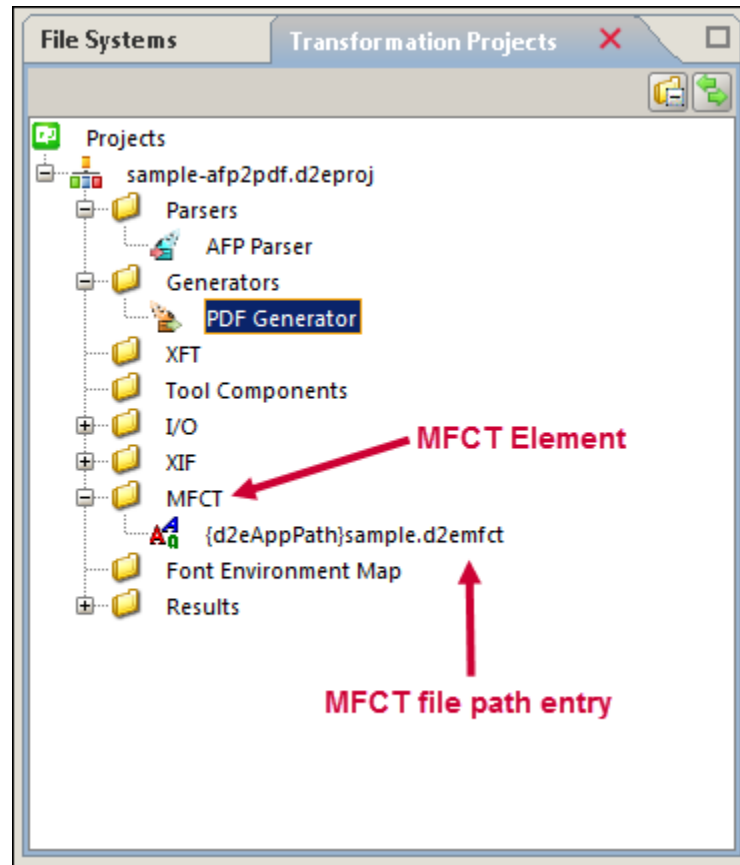


Figure 7-1: MFCT essentials in Transformation Projects tab

7.2.3 ICU Tables

International Components for Unicode (ICU) tables map AFP code points to the proper Unicode code point. Many AFP Codepages reference the proper ICU Table number to map the font's code points into Unicode. In these cases the use of the ICU table is automatic and no custom glyph mapping needs to be done. Some AFP codepages do not define the ICU table even though they use a standard defined encoding. In these cases it may be possible to determine the proper ICU table by choosing different ones in the glyph mapper and viewing the results.

Related Links

- [“Codepage” on page 582](#)
- [“Map Glyph Fonts” on page 563](#)

7.2.3.1 Why you need ICU Capability in Output Transformation Engine

A double-byte Character Set (DBCS) may contain tens of thousands of entries. In order for the glyph names to line up correctly with the output font code points, each individual glyph must be mapped if it falls outside a standard EBCDIC to ASCII translation. ICU tables greatly simplify this procedure by providing the correct glyph mappings. For more information, see [“Character Sets” on page 580](#).

For single-byte fonts whose encodings fall outside the standard EBCDIC to ASCII translation, either an ICU table is used or the glyphs are individually mapped.



Note: See the [“Map Glyph Fonts” on page 563](#) tab of the MFCT Window for information on using the [“Glyph Mappings Controls” on page 564](#) function to choose the ICU Table which best fits your data.

7.3 MFCT Window

The MFCT window is used to complete the following tasks:

- Map and manage fonts
- Map glyphs
- Control font properties
- Map translations

The MFCT window can be accessed by either:

- Double-clicking the **Master Font Correlation Table** file path entry in the **Transformation Project** tab. This branch of the Projects tree is easily identified by the  icon.
- Right-clicking the MFCT element/node and selecting **Show Active MFCT** from the shortcut menu.

The MFCT window has the following three regions:

- **MFCT Toolbar.** The toolbar enables quick editing of the MFCT entries.
- **MFCT Window Sidebar Menu.** The sidebar menus appear down the left side of the window, and enable rapid transitions between MFCT views to:
 - MFCT Fonts
 - MFCT Colors
 - MFCT Translations
- **The MFCT.** The Master Font Correlation Table itself.

Across the top, you also have access to the Mange Fonts and Map Glyphs tabs.

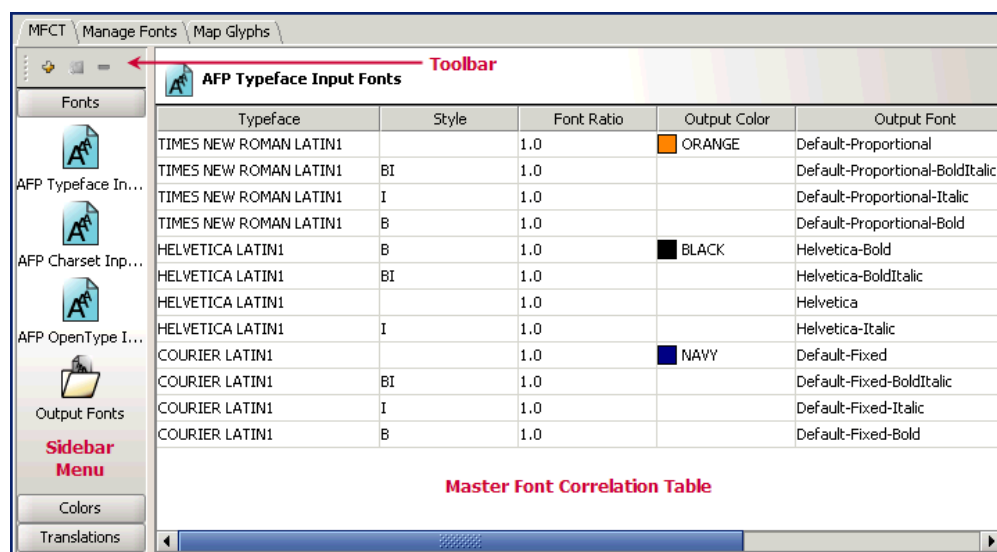


Figure 7-2: MFCT window

7.3.1 Resizing a Master Font Correlation Table

To resize a column in the MFCT:

1. Hover your cursor over the right edge of the column's title bar.
2. Click and drag the column edge until the desired width is achieved.

7.3.2 Sorting a Master Font Correlation Table

The data within the cells of an MFCT's columns can be sorted in either an ascending or descending ordered sequence. The capability to rearrange the table information in this manner makes working with MFCTs, particularly lengthy tables that contain hundreds of font sets, much easier to handle.

To sort an MFCT, click on a column header to arrange the table in order based on the column's information. Clicking the column heading again arranges the table order in the opposite direction (ascending or descending).

7.3.3 MFCT Toolbar

The following table describes actions that can be performed on the MFCT toolbar in the Fonts window.

Table 7-1: MFCT toolbar functions

Icon	Menu Command	Description
	Add	Adds an entry to the selected table or list.
	Edit	Opens the properties dialog box for the selected item.
	Remove	Removes the selected entry from the table or list.

7.3.4 MFCT Window Sidebar Menu

The MFCT window has three sidebar menus for setting up and editing your Font Table:

- **MFCT Fonts Sidebar Menu.** The fonts sidebar contains icons related to all of the font mapping properties for the font families in your input and output files.
- **MFCT Colors Sidebar Menu.** The color sidebar contains icons which provide a way for you to assign and edit the colors used to produce your output.
- **MFCT Translations Sidebar Menu.** The translations sidebar menu gives you character-level control over your font correlation. It contains icons which provide ways for you to control your font correlation.

Switching Between Sidebar Menus

To switch between sidebars, click the title of the sidebar you wish to display. The options available in the sidebar menu change according to which sidebar title is selected.

Related Links

- [Figure 7-2](#)
- [“MFCT Fonts Sidebar Menu” on page 509](#)
- [“MFCT Colors Sidebar Menu” on page 548](#)
- [“MFCT Translations Sidebar Menu” on page 554](#)

7.3.5 MFCT Fonts Sidebar Menu

The **Fonts** sidebar menu contains several options which allow you to control the fonts and the font attributes you use in your project. The Fonts Sidebar only displays the input fonts that match your input type:

- “AFP Input” on page 509 
- “Metacode Input” on page 509 
- “PCL Input” on page 509 
- “PDF Input” on page 510 

AFP Input

If your input file is AFP, then you see the following font table options:

- “AFP Typeface Input Fonts Table” on page 510. Displays a table showing the mapping of the AFP font families to the various output font typeface names used in your project.
- “AFP Charset Input Fonts Table” on page 512. Displays a table showing the mapping of individual AFP character set names or character codepage pairs to the matching output fonts used in the project.
- “AFP OpenType Input Fonts Table” on page 513. Displays a table showing the mapping of AFP OpenType fonts to the matching output fonts used in the project.
- “Output Fonts” on page 535. Displays a table showing the output fonts used in your project.

Metacode Input

If your input file is Metacode, then you see the following font table options:

- “Metacode Input Font Family Groups Table” on page 516. Displays a table showing the Metacode font families used in your project.
- “Metacode Input Family Mappings Table” on page 518. Displays a table showing the mapping of the Metacode font families to the various output font typeface names used in your project.
- “Metacode Input Font Mappings Table” on page 519. Displays a table showing the mapping of individual Metacode Font names to the matching output fonts used in the project.
- “Output Fonts” on page 535. Displays a table showing the output fonts used in your project.

PCL Input

If your input file is PCL, then you see the following font table options:

- **“PCL Input Fonts Table” on page 521.** Displays a table showing the mapping of input fonts in PCL 5 printstreams to the matching output fonts used in the project.
- **“PCL XL Input Fonts Table” on page 523.** Displays a table showing the mapping of input fonts in PCL XL (or PCL 6) printstreams to the matching output fonts used in the project.
- **“PCL Soft Input Fonts Table” on page 525.** Displays a table showing the mapping of PCL soft input fonts (fonts whose data is embedded in the PCL file) to the matching output fonts used in the project.
- **“Output Fonts” on page 535.** Displays a table showing the output fonts used in your project.

PDF Input

If your input file is PDF or another file type that contains PDF object containers, then you see the following font table option:

- **“PDF Input Font Mappings Table” on page 533.** Displays a table showing the mapping of individual PDF font names to the matching output fonts used in the project.
- **“Output Fonts” on page 535.** Displays a table showing the output fonts used in your project.

7.3.5.1 AFP Fonts

The topics in this section describe AFP input fonts.

7.3.5.1.1 AFP Typeface Input Fonts Table

The **AFP Typeface Input Fonts table** displays the mapping of the AFP font families to the various output font family names used in your project.

The AFP family input fonts table can be accessed by selecting **AFP Typeface Input Fonts** on the MFCT Fonts Sidebar menu and contains several columns:



Note: For more information on the MFCT Fonts Sidebar menu, see **“MFCT Fonts Sidebar Menu” on page 509.**

- **Typeface.** Indicates the AFP character set name.
- **Style.** Indicates the style of the font:
 - **B** - Bold.
 - **I** - Italic.
 - **BI** - Bold italic.
- **Font Ratio.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less

than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.

- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding an AFP Typeface Input Font

To add an AFP Typeface input font:

1. On the MFCT tab, select the AFP Typeface Input Fonts icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add AFP Typeface** dialog.
3. Edit the available AFP Font Mapping Attributes.
4. Select **OK** to accept the values, or **Cancel** to exit without making any changes. The new font mapping is added to the AFP Typeface Input Fonts table.

Editing an AFP Typeface Input Font

To edit an AFP Typeface input font's resources:

1. On the MFCT tab, select the **AFP Typeface Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the AFP Typeface Input Fonts table.
3. Click **Edit Resource**, , from the MFCT toolbar. Optionally, you can also double-click on the font name to open the **Edit AFP Typeface** dialog.
4. Edit the available AFP Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing an AFP Typeface Input Font

To remove an AFP Typeface input font:

1. Select the font you want to eliminate from the **AFP Typeface Input Fonts** table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove AFP Family** dialog.

3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.5.1.2 AFP Charset Input Fonts Table

The **AFP Charset (Character set) Input Fonts** table displays the mapping of individual AFP character set names or character codepage pairs to the matching output fonts used in the project.

The AFP Character Set Input Fonts table can be accessed by selecting **AFP Charset Input Fonts** on the MFCT Fonts Sidebar menu and contains several columns:



Note: For more information on the MFCT Fonts Sidebar menu, see “[MFCT Fonts Sidebar Menu](#)” on page 509.

- **Charset.** Indicates the AFP character set name.
- **Codepage.** Indicates the AFP codepage name.
- **Point Size.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.
- **Output Color.** Indicates the color to be used in the output for that font. *If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.*
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding an AFP Charset Input Font

To add an AFP Charset (Character Set) input font:

1. On the MFCT tab, select the **AFP Charset Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add AFP Charset** dialog.
3. Edit the available AFP Font Mapping Attributes.
4. Select **OK** to accept the values, or **Cancel** to exit without making any changes.
The new font mapping is added to the AFP Charset Input Fonts table.

Editing an AFP Charset Input Font

To edit an AFP Charset (Character Set) Input Font's resources:

1. On the MFCT tab, select **AFP Charset Input Fonts** from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the AFP Charset Input Fonts table.
3. Click **Edit Resource**, , from the MFCT toolbar. Optionally, you can also double-click on the font name to open the **Edit AFP Charset Font** dialog.
4. Edit the available AFP Font Mapping Attributes.
5. Select **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing an AFP Charset Input Font

To remove an AFP Charset input font:

1. Select the font you want to eliminate from the **AFP Charset Input Fonts** table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove AFP Charset** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT toolbar, see “MFCT Toolbar” on page 508.

7.3.5.1.3 AFP OpenType Input Fonts Table

The **AFP OpenType Input Fonts table** displays the mapping of OpenType fonts used in AFP input streams, to the matching output fonts used in your project. The AFP OpenType Input Fonts table can be accessed by selecting **AFP OpenType Input Fonts** on the MFCT Fonts Sidebar menu and contains several columns:



Note: For more information on the MFCT Fonts Sidebar menu, see “MFCT Fonts Sidebar Menu” on page 509.

- **Full Font Name.** Indicates the entire font name of the OpenType font, such as *Times New Roman Bold Italic*. This name will correspond to the name used in the AFP printstream's font list (MDR records).
- **Font Ratio.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used, then the name of the color table is displayed. *Click on the cell to edit the value.*

- **Output Font.** Indicates the name of the output font mapped to that typeface font.
To edit the font, click on the cell and select one from the dropdown menu.
- **TIFF.** Reveals the font to be used in TIFF output.
- **PDF.** Reveals the font to be used in PDF output.
- **AFP.** The font to be used in AFP output.
- **Metacode.** Reveals the font to be used in Metacode output.
- **PCL.** Reveals the font to be used in PCL output.
- **Postscript.** Reveals the font to be used in Postscript output.
- **HTML.** The font to be used in HTML output.

Adding an AFP OpenType Input Font

To add an AFP OpenType input font:

1. On the MFCT tab, select the **AFP OpenType Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add AFP OpenType Font** dialog.
3. Edit the available AFP Font Mapping Attributes.
4. Click **OK** to accept the values, or **Cancel** to exit without making any changes.
The new font mapping is added to the AFP OpenType Input Fonts table.

Editing an AFP OpenType Input Font

To edit an AFP OpenType input font:

1. On the MFCT tab, select the **AFP OpenType Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the AFP OpenType Input Fonts table.
3. Click **Edit Resource**, , on the MFCT toolbar. Optionally, you can also double-click the OpenType font name to open the **Edit AFP OpenType Font** dialog.
4. Edit the available AFP Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing an AFP OpenType Font

To remove an AFP OpenType input font:

1. Select the font you want to eliminate from the **AFP OpenType Input Fonts** table.

2. Click **Remove Resource**, , on the MFCT toolbar to open the **Remove AFP OpenType Font** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

Importing OpenType Fonts with the AFP Generator

To import an OpenType font using the AFP Generator:

1. Select the **AFP Generator** component in your project to see its properties.
2. Expand the **InputResources** parameter.
3. Click the ellipsis button, , at **FdOpenTypeFonts** and enter the path to where the OpenType fonts are located.

Exporting OpenType Fonts with the AFP Parser

To export an OpenType font using the AFP Parser:

1. Select the **AFP Parser** component in your project to see its properties.
2. On the Properties tab, expand the **ResGroupOption** parameter.
3. Select the **ExportOpenTypeFonts** check box.
4. Click the ellipsis button, , at **FdExportOpenTypeFonts** and enter the path to where you want to export the file. This would usually be in the resource folder within the project file.

7.3.5.1.4 AFP Font Mapping Attributes

The values available to be mapped for **AFP Typeface**, **Charset**, or **OpenType** input fonts may include:

- **AFP Font:**
 - **Typeface.** The AFP typeface name (for Typeface input fonts).
 - **Charset.** The AFP character set name (for Charset input fonts).
 - **Full font name.** The entire font name (for OpenType input fonts).
 - **Style.** The style of the typeface; **Bold** or **Italic** (for Typeface input fonts).
 - **Codepage.** The codepage identification (for Charset input fonts).
 - **Font Ratio adjustment.** Select a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have

overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.

- **Maps To:**

- **Output Font.** Select an output font to which the AFP font will be mapped.
- **Output Color.** Select the radio buttons associated with the color options you want to use:
- **Output Color.** Select a color from the dropdown menu.
- **Output Color Table.** Select an output color table from the dropdown menu.

When you add a new mapping, all values are available to be set. However, when you are editing a mapping, only the Maps To values can be changed. The AFP Font values (with the exception of Font ratio adjustment) cannot be changed once they are set.

7.3.5.2 Metacode Fonts

The topics in this section discuss Metacode input fonts.

7.3.5.2.1 Metacode Input Font Family Groups Table

The **Metacode Font Family Groups** table displays the mappings of the Metacode font family groups used in the project.

The Metacode Font Family Groups table contains several columns:

- **Family Groups.** Indicates the Metacode font family name.
- **Font Name.** Indicates the particular Metacode font name.
- **Point Size.** Specifies the point size of the Metacode font.
- **Orientation.** Indicates the orientation of the Metacode font.

Adding a Metacode Input Font Family

To add a Metacode input font family:

1. On the MFCT tab, select the **Metacode Font Families** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT Toolbar to open the **Add Meta Family dialog**.
3. In the **Family name** field, enter the desired name, and click **OK**. The new Metacode Font Family Group appears in the Metacode Input Families table.
4. Select the new **Metacode Font Family Group**, and click the **Add button at the bottom of the table to open the Add Metacode Family Group dialog**.
5. Edit the available output font definitions for the Metacode Family group. The following fields must be filled out in order to continue:

- **Font name.** Indicates the particular Metacode font name.
 - **Point size.** Specifies the point size of the Metacode font.
 - **Orientation.** Indicates the orientation of the Metacode font. You can select from **Unknown**, **Portrait**, **Landscape**, **InversePortrait**, or **InverseLandscape**.
6. When you are finished defining the output font attributes for the font family, click **OK**.

The Metacode Font Families table opens. Expand the Family Group name to view the output font settings. Multiple output fonts can be added under a Family Group by repeating this procedure as necessary.

Editing a Metacode Input Font Family

To edit a Metacode input font family:

1. On the MFCT tab, select the **Metacode Font Family** icon from the MFCT Fonts Sidebar menu.
2. Do one of the following:
 - If you want to change the Family Group name, select the **Metacode Font Family Group** you want to change from the Metacode Input Font Families table, and click **Edit Resources**, , from the MFCT toolbar.
 - If you want to change the Family Group's font resources, select the **Font Resource** from the Metacode Input Font Families table, and click the **Edit** button below the table.
3. Make your desired changes in the **Edit** dialog that appears.
4. Click **OK** to save your changes, or **Cancel** to discard your changes and exit.

Removing a Metacode Input Font Family

To delete a Metacode font family:

1. On the MFCT tab, select the **Metacode Font Families** icon from the MFCT Fonts Sidebar menu.
2. Do one of the following:
 - If you want to remove an entire Family Group, select the **Metacode Font Family Group** you want to delete from the Metacode Input Font Families table, and click **Remove Resources**, , from the MFCT toolbar.
 - If you want to remove a Family Group's font resources, select the **Font Resource** from the Metacode Input Font Families table, and click the **Remove** button below the table.
3. Confirm the removal of the selected font family or resources. Click **Yes** to confirm the deletion, or **No** to cancel the operation.

7.3.5.2.2 Metacode Input Family Mappings Table

The **Metacode Font Family Mappings** table displays the mapping of the Metacode font families to the various output font families and can be accessed by clicking **Metacode Input Family Mappings** on the MFCT Fonts Sidebar menu.



Note: For more information on the MFCT Fonts Sidebar menu, see “[MFCT Fonts Sidebar Menu](#)” on page 509.

The Metacode Font Family Mappings table contains the following columns:

- **Font Family.** Indicates the Metacode font family name.
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding a Metacode Input Family Mapping

To add a Metacode input family mapping:

1. On the MFCT tab, select the **Metacode Input Family Mappings** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add Metacode Family Mapping** dialog.
3. Edit the available Metacode Font Mapping Attributes.
4. Select **OK** to accept the values or **Cancel** to exit without making any changes.
The new family mapping is added to the Metacode Input Family Mappings table.

Editing a Metacode Input Family Mapping

To edit a Metacode input family mapping:

1. On the MFCT tab, select the **Metacode Input Family Mapping** icon from the MFCT Fonts Sidebar menu.
2. Select the font family you want to change from the Metacode Input Family Mappings table.
3. Click **Edit Resource**, , from the MFCT toolbar to open the **Edit Metacode Family Mapping** dialog.

4. Edit the available Metacode Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing a Metacode Input Family Mapping

To remove a Metacode input family mapping:

1. On the Metacode Input Family Mappings table, select the font family mapping you want to eliminate from the table.
2. Click **Remove Resource**, , from the MFCT toolbar.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information about the MFCT toolbar, see “MFCT Toolbar” on page 508.

7.3.5.2.3 Metacode Input Font Mappings Table

The **Metacode Font Mappings** table displays the mapping of individual Metacode fonts to the various output fonts used in your project.

The Metacode Input Font Mappings table contains the following columns:

- **Font Family.** Indicates the Metacode font family name.
- **Orientation.** Indicates the orientation of the Metacode font.
- **Point Size.** Specifies the point size of the Metacode font.
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding a Metacode Input Font Mapping

To add a Metacode input font mapping:

1. On the MFCT tab, select the **Metacode Input Font Mapping** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add Metacode Font Mapping** dialog.
3. Edit the available Metacode Font Mapping Attributes.

4. Select **OK** to accept the values, or **Cancel** to exit without making any changes.
The new font mapping is added to the Metacode Input Font Mappings table.

Editing a Metacode Input Font Mapping

To edit a Metacode font mapping's resources:

1. On the MFCT tab, select **Metacode Input Font Mapping** from the MFCT Fonts Sidebar menu.
2. Select the input font mapping you want to change from the Metacode Input Font Mappings Table.
3. Click **Edit Resource**, , from the MFCT toolbar to open the **Edit Metacode Font Mapping** dialog.
4. Edit the available Metacode Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing a Metacode Input Font Mapping

To remove a Metacode input font mapping:

1. On the Metacode Input Font Mappings table, select the font mapping you want to eliminate from the table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove Metacode Font Mapping** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT toolbar, see “MFCT Toolbar” on page 508.

7.3.5.2.4 Metacode Font Mapping Attributes

The values available to be mapped in the **Edit Metacode Family Group** dialog window, the **Add Metacode Family Mapping** and **Edit Metacode Family Mapping** dialog windows, and the **Add Metacode Font Mapping** and **Edit Metacode Font Mapping** dialog windows are the following:

- **Metacode Family**
 - **Font family/Family name.** Indicate a Metacode font family name.
 - **Metacode Font**
 - **Font name.** Indicate a Metacode input/output font name.
 - **Orientation.** With the dropdown menu, specify the direction the font should be. You can select from **Unknown**, **Portrait**, **Landscape**, **InversePortrait** or **InverseLandscape**.

- **Point size.** Indicate the point size of the fonts.
- **Maps To:**
 - **Output font.** Select an output font to which the Metacode font is to be mapped.
 - **Output color.** Select from one of the following options:
 - **Output color.** Select a color from the dropdown menu.
 - **Output color table.** Select an output color table from the dropdown menu.
- **Translation table.** Select a Translation Table to use to map the font. For more information, see [“Translation Tables” on page 554](#).

7.3.5.3 PCL Fonts

The topics in this section discuss PCL input fonts.

7.3.5.3.1 PCL Input Fonts Table

The **PCL Input Fonts table** displays the mapping of input fonts in PCL 5 printstreams to the matching output fonts used in the project. The PCL parser compares font references in the PCL 5 printstreams to the entries in this table for the best match, in order to determine the correct output font mapping. The table is displayed by selecting **PCL Input Fonts** from the MFCT Fonts Sidebar menu.



Note: For more information on the MFCT Fonts Sidebar menu, see [“MFCT Fonts Sidebar Menu” on page 509](#).

The PCL Input Fonts table contains the following columns:

- **Typeface.** Indicates the typeface name string in PCL format, or numeric typeface ID value to match. Alternatively, a single asterisk (*) character may be used to match any typeface in the printstream. For a complete list of valid font names in PCL format, see [“PCL Typeface Values” on page 529](#).
- **Spacing Type.** Refers to categories of fonts which are determined by their character spacing or width. Select **Fixed** to associate only PCL fonts having characters of the same width (such as Courier) with that mapping entry. Select **Proportional** to associate only PCL fonts having characters of different widths (such as Times Roman) with that mapping entry. Select **Any** to associate both fixed and proportional PCL fonts with the mapping entry.
- **Style.** Displays a comma delimited list of numeric or textual styles that apply to this font mapping. For the regular default style, use *upright, solid*, or 0. To match any style, use *“*”* (asterisk).
- **WeightMax and Min.** Indicates the maximum and minimum value for weight that this mapping applies to. Setting the Max and Min values to the same value maps to one specific weight value. For normal weighted fonts, use zero (0) as the value. Negative values indicate thin and light fonts (where -7 is the thinnest), and positive values indicate bold fonts (where 7 is the boldest).

- **Font Ratio.** Specifies the **value, in points, used to determine the size of the font.**
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- **TIFF.** Reveals the font to be used in TIFF output.
- **PDF.** Reveals the font to be used in PDF output.
- **AFP.** Reveals the font to be used in AFP output.
- **Metacode.** Reveals the font to be used in Metacode output.
- **PCL.** Reveals the font to be used in PCL output.
- **Postscript.** Reveals the font to be used in Postscript output.
- **HTML.** Reveals the font to be used in HTML output.
- **Trans Table.** Specifies the name of the translation table to be applied.

Adding a PCL Input Font

To add a PCL Input Font:

1. On the MFCT tab, select the PCL Input Font icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add PCL Font Mapping** dialog.
3. Edit the available PCL Font Mapping Attributes.
4. Click **OK** to accept the values, or **Cancel** to exit without making any changes.
The new font mapping is added to the PCL Input Fonts table.

Editing a PCL Input Font

To edit a PCL input font's resources:

1. On the MFCT tab, select the PCL Input Font icon from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the PCL Input Fonts table.
3. Click **Edit Resource**, , on the MFCT toolbar. Optionally, you can also double-click on the input font name to open the **Edit PCL Font Mapping** dialog.
4. Edit the available PCL Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing a PCL Input Font

To remove a PCL input font mapping:

1. On the PCL Input Font table, select the font mapping you want to eliminate from the table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove PCL Font Mapping** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.5.3.2 PCL XL Input Fonts Table

The **PCL XL Input Fonts** table displays the mapping of input fonts in PCL XL (PCL 6) printstreams to the matching output fonts used in the project. The PCL parser compares font references in the PCL XL printstreams to the entries in this table for the best match, in order to determine the correct output font mapping. You can view the PCL XL Input Fonts table by selecting the **PCL XL Input Fonts** icon on the MFCT Fonts Sidebar menu.



Note: For more information on the MFCT Fonts Sidebar menu, see “[MFCT Fonts Sidebar Menu](#)” on page 509

The PCL XL Input Fonts table contains several columns:

- **Typeface.** Indicates the typeface name string in PCL XL format, or numeric typeface ID value to match. Alternatively, a single asterisk (*) character may be used to match any typeface in the printstream. For a complete list of valid font names in PCL XL format, see “[PCL Typeface Values](#)” on page 529.
- **CharAngle.** Allows you to map PCL XL fonts based on whether they have a CharAngle integer value. The valid range of values is from -360 to +360. The default value of “*” (asterisk) indicates that this mapping applies to all CharAngle values. A common use would be to specify !=0 to create a mapping for any non-zero rotations. This would be useful in transforms where the output format does not support non-zero character angles and special mappings (such as rasterizing the font) are needed to accommodate this.
- **BoldValueMax.** Indicates the maximum value allowed for the CharBoldValue in the PCL. Setting both the BoldValueMin and BoldValueMax parameters to the same value will restrict the CharBoldValue parameter to match with only that specific bold value.
- **BoldValueMin.** Indicates the minimum value allowed for the CharBoldValue in the PCL. Setting both the BoldValueMin and BoldValueMax parameters to the same value will restrict the CharBoldValue parameter to match with only that specific bold value.

- **CharShearX**. Allows you to map PCL XL fonts based on whether they have a CharShearX value. The default value of "*" (asterisk) indicates that this mapping applies to all CharShearX values. In transforms where the output format does not support shear values, special mappings (such as rasterizing the font) are needed to accommodate this.
- **CharShearY**. Allows you to map PCL XL fonts based on whether they have a CharShearY value. The default value of "*" (asterisk) indicates that this mapping applies to all CharShearY values. In transforms where the output format does not support shear values, special mappings (such as rasterizing the font) are needed to accommodate this.
- **Font Ratio**. Specifies the value, in points, used to determine the size of the font.
- **Output Color**. Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font**. Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- **Trans Table**. Specifies the name of the translation table to be applied. For more information, see ["Translation Tables" on page 554](#).
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding a PCL XL Input Font

To add a PCL XL input font:

1. On the MFCT tab, select the **PCL XL Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resources**, , on the MFCT toolbar to open the **Add PCL XL Input Font** dialog.
3. Edit the available PCL Font Mapping Attributes.
4. Select **OK** to accept the values, or **Cancel** to exit without making any changes.
The new font mapping is added to the PCL XL Input Fonts table.

Editing a PCL XL Input Font

To edit a PCL XL input font:

1. On the MFCT tab, select the **PCL XL Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the PCL XL Input Fonts table.
3. Click **Edit Resources**, , on the MFCT toolbar. Optionally, you can also double-click on the input font name to open the **Edit PCL XL Input Font** dialog.

4. Edit the available PCL XL Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing a PCL XL Font Mapping

To remove a PCL XL Input Font Mapping:

1. On the PCL XL Input Font table, select the font mapping you want to eliminate from the table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove PCL XL Font Mapping** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information about the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.5.3.3 PCL Soft Input Fonts Table

The **PCL Soft Input Fonts table** displays the mapping of PCL soft input fonts (fonts whose data is embedded in the PCL file) to the matching output fonts used in the project. These mappings apply to both PCL 5 and PCL XL printstreams. You can view the PCL Soft Input Fonts table by selecting the **PCL Soft Input Fonts** icon on the MFCT Fonts Sidebar menu.



Note: For more information on the MFCT Fonts Sidebar menu, see “[MFCT Fonts Sidebar Menu](#)” on page 509

The PCL Soft Input Fonts table contains the following columns:

- **Typeface.** Indicates the typeface name string in PCL 5 or PCL XL format, or numeric typeface ID value to match. Alternatively, a single asterisk (*) character may be used to match any typeface in the printstream. For a complete list of valid font names in PCL 5 and PCL XL format, see “[PCL Typeface Values](#)” on page 529.
- **SoftFontType.** Indicates the soft font type to which this mapping should apply. Possible values are **Any**, **PclBitmap**, **TrueType**, or **Intellifont**.
- **SpacingType.** Refers to categories of fonts which are determined by their character spacing or width. Select **Fixed** to associate only PCL fonts having characters of the same width (such as Courier) with that mapping entry. Select **Proportional** to associate only PCL fonts having characters of different widths (such as Times Roman) with that mapping entry. Select **Any** to associate both fixed and proportional PCL fonts with the mapping entry.
- **WeightMax** and **Min.** Indicates the maximum and minimum value for weight that this mapping applies to. Setting the Max and Min values to the same value maps to one specific weight value. For normal weighted fonts, use zero (0) as the

value. Negative values indicate thin and light fonts (where -7 is the thinnest), and positive values indicate bold fonts (where 7 is the boldest).

- **UnicodeOffsetShift.** Specifies the numeric value (positive or negative) that should be applied to the soft font codepoints to determine the Unicode value. This is sometimes required for soft fonts that have a custom encoding scheme.
- **Font Ratio.** Specifies the value, in points, used to determine the size of the font.
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- **Trans Table.** Specifies the name of the translation table to be applied. For more information, see [“Translation Tables” on page 554](#).
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding a PCL Soft Input Font

To add a PCL soft input font:

1. Select the **PCL Soft Input Fonts** icon from the MFCT Fonts Sidebar.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add PCL Soft Input Font** dialog.
3. Enter the available PCL Soft Font Mapping Attributes.
4. Select **OK** to accept the values, or **Cancel** to exit without making any changes.
The new font mapping is added to the PCL Soft Input Fonts table.

Editing a PCL Soft Input Font

To edit a PCL soft input font:

1. On the MFCT tab, select the **PCL Soft Input Fonts** icon from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the PCL Soft Input Fonts table.
3. Click **Edit Resource**, , on the MFCT toolbar. Optionally, you can also double-click on the input font name to open the **Edit PCL Soft Input Font** dialog.
4. Edit the available PCL Soft Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing a PCL Soft Input Font

To remove a PCL Soft Input Font Mapping:

1. On the PCL XL Input Font table, select the font mapping you want to eliminate from the table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove PCL Soft Font Mapping** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.

For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.5.3.4 PCL Font Mapping Attributes

The font values available to be mapped for PCL 5 are:



Note: For a complete list of valid PCL 5 and PCL XL font names, see “[PCL Typeface Values](#)” on page 529.

- **Typeface.** Indicates the typeface name string or numeric typeface ID value to match. Alternatively, a single asterisk (*) character may be used to match any typeface in the printstream.
- **Weight Max and Min.** Indicates the maximum and minimum value for weight that this mapping applies to. Setting the Max and Min values to the same value maps to one specific weight value. For normal weighted fonts, use zero (0) as the value. Negative values indicate thin and light fonts (where -7 is the thinnest), and positive values indicate bold fonts (where 7 is the boldest).
- **Spacing type.** Refers to categories of fonts which are determined by their character spacing or width. Select **Fixed** to associate only PCL fonts that have characters of the same width (such as Courier) with that mapping entry. Select **Proportional** to associate only PCL fonts with characters of different widths (such as Times Roman) to that mapping entry. Select **Any** to associate both fixed and proportional PCL fonts with the mapping entry.
- **Font style.** Displays a comma delimited list of numeric or textual styles that apply to this font mapping. For the regular default style, use “upright”, “solid”, or “0”. To match any style, use “*” (asterisk).
- **Font ratio adjustment.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.

The font values available to be mapped for PCL XL are:

- **Typeface.** Indicates the typeface name string or numeric typeface ID value to match. Alternatively, a single asterisk (*) character may be used to match any typeface in the printstream.

- **Bold Value Max and Min.** Indicates the maximum and minimum values allowed for the CharBoldValue in the PCL. Setting both maximum and minimum to the same value will restrict the CharBoldValue parameter to match with only that specific bold value.
- **CharShearX.** Allows you to map PCL XL fonts based on whether they have a CharShearX value. The default value of "*" (asterisk) indicates that this mapping applies to all CharShearX values. In transforms where the output format does not support shear values, special mappings (such as rasterizing the font) are needed to accommodate this.
- **CharShearY.** Allows you to map PCL XL fonts based on whether they have a CharShearY value. The default value of "*" (asterisk) indicates that this mapping applies to all CharShearY values. In transforms where the output format does not support shear values, special mappings (such as rasterizing the font) are needed to accommodate this.
- **CharAngle.** Allows you to map PCL XL fonts based on whether they have a CharAngle integer value. The valid range of values is from -360 to +360. The default value of "*" (asterisk) indicates that this mapping applies to all CharAngle values. A common use would be to specify "!=0" to create a mapping for any non-zero rotations. This would be useful in transforms where the output format does not support non-zero character angles and special mappings (such as rasterizing the font) are needed to accommodate this.
- **Font ratio adjustment.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.

The font values available to be mapped for PCL Soft Inputs are:

- **Typeface.** Indicates the numeric typeface ID value. Alternatively, a single asterisk (*) character may be used to match any typeface in the printstream.
- **Weight Max and Min.** Indicates the maximum and minimum value for weight that this mapping applies to. Setting the Max and Min values to the same value maps to one specific weight value. For normal weighted fonts, use zero (0) as the value. Negative values indicate thin and light fonts (where -7 is the thinnest), and positive values indicate bold fonts (where 7 is the boldest).
- **Spacing type.** Refers to categories of fonts which are determined by their character spacing or width. Select **Fixed** to associate only PCL fonts that have characters of the same width (such as Courier) with that mapping entry. Select **Proportional** to associate only PCL fonts with characters of different widths (such as Times Roman) to that mapping entry. Select **Any** to associate both fixed and proportional PCL fonts with the mapping entry.
- **Soft font type.** Indicates the soft font type to which this mapping should apply. You can select from Any, PclBitmap, TrueType, or Intellifont.

- **Unicode offset shift.** Specifies the numeric value (positive or negative) that should be applied to the soft font codepoints to determine the Unicode value. This is sometimes required for soft fonts that have a custom encoding scheme.
- **Font ratio adjustment.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.

PCL 5, PCL XL, and PCL Soft Input fonts all share the same configurable **Maps To** attributes:

- **Output font.** Select an output font to which the Metacode font is to be mapped.
- **Output color.** Select from one of the following options:
 - **Output color.** Select a color from the dropdown menu.
 - **Output color table.** Select an output color table from the dropdown menu.
- **Translation table.** Select a Translation Table to use to map the font.

7.3.5.3.5 PCL Typeface Values

The following is a table of valid PCL typeface values used when mapping PCL 5 input fonts, PCL XL input fonts, and PCL soft input fonts:

PCL 5 Typeface	PCL XL Typeface	Typeface ID Value	Spacing	Style	Weight
Albertus Extra Bold	AlbertusXb	4362	1	0	4
Albertus Medium	AlbertusMb	4362	1	0	1
Antique Olive	AntiqOlive	4168	1	0	0
Antique Olive Bold	AntiqOliveBd	4168	1	0	3
Antique Olive Italic	AntiqOliveIt	4168	1	1	0
Arial	Arial	16602	1	0	0
Arial Bold	ArialBd	16602	1	0	3
Arial Italic	ArialIt	16602	1	1	0
Arial Bold Italic	ArialBdIt	16602	1	1	3
CG Times	CGTimes	4101	1	0	0

CG Times Bold	CGTimesBd	4101	1	0	3
CG Times Italic	CGTimesIt	4101	1	1	0
CG Times Bold Italic	CGTimesBdIt	4101	1	1	3
CG Omega	CGOmega	4113	1	0	0
CG Omega Bold	CGOmegaBd	4113	1	0	3
CG Omega Italic	CGOmegalT	4113	1	1	0
CG Omega Bold Italic	CGOmegaBdIt	4113	1	1	3
Clarendon Condensed	ClarendonCd Bd	4140	1	4	3
Coronet	Coronet	4116	1	1	0
Courier	Courier	4099	0	0	0
Courier Bold	CourierBd	4099	0	0	3
Courier Italic	CourierIt	4099	0	1	0
Courier Bold Italic	CourierBdIt	4099	0	1	3
CourierPS	CourierPS	24579	1	0	0
CourierPS Bold	CourierPSBd	24579	1	0	3
CourierPS Oblique	CourierPSIt	24579	1	1	0
CourierPS Bold Oblique	CourierPSBdIt	24579	1	1	3
Garamond Antiqua	GaramondAntiqua	4197	1	0	0
Garamond Halbfett	GaramondHlb	4197	1	0	3
Garamond Kursiv	GaramondKsr	4197	1	1	0
Garamond Kursiv Halbfett	GaramondKsrHlb	4197	1	1	3
Helvetica	Helvetica	24580	1	0	0
Helvetica Bold	HelveticaBd	24580	1	0	3

Helvetica Oblique	HelveticaOb	24580	1	1	0
Helvetica Bold Oblique	HelveticaBdOb	24580	1	1	3
Helvetica Narrow	HelveticaNr	24580	1	4	0
Helvetica Narrow Bold	HelveticaNrBd	24580	1	4	3
Helvetica Narrow Oblique	HelveticaNrOb	24580	1	5	0
Helvetica Narrow Bold Oblique	HelveticaNrBdOb	24580	1	5	3
ITC Avant Garde Gothic Book	ITCAvantGardBk	24607	1	0	0
ITC Avant Garde Gothic Demi	ITCAvantGardDb	24607	1	0	2
ITC Avant Garde Gothic Book Oblique	ITCAvantGardBkOb	24607	1	1	0
ITC Avant Garde Gothic Demi Oblique	ITCAvantGardDbOb	24607	1	1	2
ITC Bookman Light	ITCBookmanLt	24623	1	0	-3
ITC Bookman Demi	ITCBookmanDb	24623	1	0	2
ITC Bookman Light Italic	ITCBookmanLtIt	24623	1	1	-3
ITC Bookman Demi Italic	ITCBookmanDbIt	24623	1	1	2
ITC Zapf Chancery Medium Italic	ZapfChanceryMdlIt	45009	1	1	0
Letter Gothic	LetterGothic	4102	1	0	0
Letter Gothic Bold	LetterGothicBd	4102	1	0	3

Letter Gothic Italic	LetterGothicIt	4102	1	1	0
Marigold	Marigold	4297	1	0	0
New Century Schoolbook Roman	NwCentSchlbkRmn	24703	1	0	0
New Century Schoolbook Bold	NwCentSchlbkBd	24703	1	0	3
New Century Schoolbook Italic	NwCentSchlbkIt	24703	1	1	0
New Century Schoolbook Bold Italic	NwCentSchlbkBdIt	24703	1	1	3
Palatino Roman	PalatinoRmn	24591	1	0	0
Palatino Bold	PalatinoBd	24591	1	0	3
Palatino Italic	PalatinoIt	24591	1	1	0
Palatino Bold Italic	PalatinoBdIt	24591	1	1	3
SymbolPS	SymbolPS	45358	1	0	0
Times Roman	TimesRmn	25093	1	0	0
Times Bold	TimesBd	25093	1	0	3
Times Italic	TimesIt	25093	1	1	0
Times Bold Italic	TimesBdIt	25093	1	1	3
Times New	TimesNewRmn	16901	1	0	0
Times New Bold	TimesNewRmnBd	16901	1	0	3
Times New Italic	TimesNewRmnIt	16901	1	1	0
Times New Bold Italic	TimesNewRmnBdIt	16901	1	1	3
Univers Medium	UniversMd	4148	1	0	0
Univers Bold	UniversBd	4148	1	0	3
Univers Medium Italic	UniversMdIt	4148	1	1	0

Univers Bold Italic	UniversBdIt	4148	1	1	3
Univers Medium Condensed	UniversCdM d	4148	1	4	0
Univers Bold Condensed	UniversBdCd	4148	1	4	3
Univers Medium Condensed Italic	UniversMdC dIt	4148	1	5	0
Univers Bold Condensed Italic	UniversBdCd It	4148	1	5	3

Related Links

- [“PCL Input Fonts Table” on page 521](#)
- [“PCL XL Input Fonts Table” on page 523](#)
- [“PCL Soft Input Fonts Table” on page 525](#)

7.3.5.4 PDF Fonts

The topics in this section discuss PDF input fonts.

7.3.5.4.1 PDF Input Font Mappings Table

The **PDF Input Font Mappings** table displays the mapping of the PDF font families to the various output font family names used in your project. The table is displayed by selecting **PDF Input Font Mappings** from the MFCT Fonts Sidebar menu.



Note: For more information on the MFCT Fonts Sidebar menu, see [“MFCT Fonts Sidebar Menu” on page 509](#).

The PDF Input Font Mappings table contains several columns:

- **Font Name.** Identifies the name of the PDF input font to map.
- **Font Ratio.** Specifies a multiplier with which to calculate the final font size. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used, then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.

- **Trans Table.** Designates the name of the translation table associated with the particular font, if available.
- Depending on which generator types are used in your project, various columns representing the valid output types (Meta, HTML, TIFF, etc.) are shown in the table, which specifies the font configuration to be used for the output type.

Adding a PDF Input Font

To add a PDF input font mapping:

1. On the MFCT tab, select the **PDF Input Font Mappings** icon from the MFCT Fonts Sidebar menu.
 2. Click **Add Resource**, , on the MFCT toolbar to open the **Add PDF Font** dialog.
 3. Edit the available PDF Font Mapping Attributes.
 4. Select **OK** to accept the values, or **Cancel** to exit without making any changes.
- The new font mapping is added to the PDF Input Font Mappings table.

Editing a PDF Input Font

To edit a PDF input font mapping:

1. On the MFCT tab, select the **PDF Input Font Mappings** icon from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the PDF Input Font Mappings table.
3. Click **Edit Resource**, , on the MFCT toolbar. Optionally, you can also double-click on the font name to open the **Edit PDF Font** dialog.
4. Edit the available PDF Font Mapping Attributes.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Removing a PDF Input Font

To remove a PDF Input font:

1. Select the font you want to eliminate from the PDF Input Font Mappings table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove PDF Font Mapping** dialog.
3. Confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.5.4.2 PDF Font Mapping Attributes

The values available to be mapped for PDF input fonts include:

- **PDF Font:**
 - **Font name.** Identifies the name of the PDF input font to map.
- **Maps To:**
 - **Output font.** Indicates the name of the output font mapped to the PDF input font.
 - **Output color.** Indicates the color to be used in the output for that font. If a color table is to be used, then the name of the color table is displayed.
 - **Translation table.** Designates the name of the translation table associated with the particular font, if available.
 - **Apply before Cmap.** Defines whether the original text byte values are translated prior to or following the Cmap translation process. A Cmap is an encoding table used by the PDF Parser to convert the raw byte values in the PDF text stream into Unicode codepoints. By default, the PDF Parser takes the original text byte values and translates them using the Cmap provided by the PDF's given font resource. If you are using a custom user translation table, then it is applied to the text byte values after the Cmap translation has been completed and these are the final values that are used when building the XifTextElement objects. However, be aware that in some situations the internal Cmap translates more than one glyph into the same Unicode value. In these cases, it will be impossible to correctly map the PDF glyphs into distinct glyphs in the output document. To combat multiple glyphs being translated into the same Unicode value, you can enable this setting to have the PDF Parser use the values within your custom user translation table instead of the Cmap values, effectively allowing you to bypass this scenario and ensuring that distinct Unicode mappings for the desired glyphs are created. Note that any codepoints not defined in the custom translation table will still default to the Cmap translation values.

7.3.5.5 Output Fonts

The topics in this section describe Output Fonts:

7.3.5.5.1 Output Fonts Table

The Output Fonts table displays all of the output fonts used in your project. You can access the table by selecting the **Output Fonts** icon on the MFCT Fonts sidebar menu. It contains the following columns:

- **Name.** Indicates the font name.
- **TIFF.** Reveals the font to be used in TIFF output.
- **PDF.** Reveals the font to be used in PDF output.
- **AFP.** Reveals the font to be used in AFP output.
- **Metacode.** Reveals the font to be used in Metacode output.
- **PCL.** Reveals the font to be used in PCL output.
- **Postscript.** Reveals the font to be used in Postscript output.

7.3.5.5.2 Adding an Output Font

To add an output font:

1. On the MFCT tab, select the **Output Font** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add Output Font** dialog.
3. Edit the available Output Font Mapping Attributes on the font tab which correlates to the output format you are using.
4. Click **OK** to accept the values, or **Cancel** to exit and discard any changes made.

The new font mapping is added to the Output Fonts table.

Related Links

- [“MFCT Fonts Sidebar Menu” on page 509](#)
- [“MFCT Toolbar” on page 508](#)
- [“Output Font Mapping Attributes” on page 543](#)

7.3.5.5.3 Editing an Output Font

To edit an output font's resources:

1. On the MFCT tab, select **Output Font** from the MFCT Fonts Sidebar menu.
2. Select the font you want to change from the Output Fonts table.
3. Click **Edit Resource**, , on the MFCT toolbar. Optionally, you can also double-click the output font name to open the **Edit Output Font** dialog.
4. Edit the available Output Font Mapping Attributes on the font tab which correlates to the output format you are using.
5. Click **OK** to accept the values, or **Cancel** to exit without making any changes.

Related Links

- [“MFCT Fonts Sidebar Menu” on page 509](#)
- [“MFCT Toolbar” on page 508](#)
- [“Output Font Mapping Attributes” on page 543](#)

7.3.5.5.4 Removing an Output Font

To remove an output font mapping:

1. On the **Output Fonts** table, select the font mapping you want to eliminate from the table.
2. Click **Remove Resource**, , from the MFCT toolbar to open the **Remove Output Font** dialog.
3. You must confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information about the MFCT toolbar, see [“MFCT Toolbar” on page 508](#).

7.3.5.5.5 AFP Output Font Families

The AFP Output Font Families table displays all of the AFP output font families used in your project. This table can be accessed by selecting **AFP Output Font Families** on the MFCT Fonts Sidebar menu. It contains the following fields :



Note: For more information about the MFCT Fonts Sidebar menu, see [“MFCT Fonts Sidebar Menu” on page 509](#).

- **Font Family Group.** Indicates the font family name. For example, CourierLatin1.
- **Codepage.** Indicates the AFP codepage name. This appears next to a Charset. For example, T1V10500.

- **Coded Font Name.** Indicates the AFP Coded font name. This appears next to a Charset and Codepage. For example, X04210JC.
- **Byte Scheme.** Indicates whether the font is single byte or double byte.
- **Encoding.** Specifies the encoding to be used. You can select from **Auto** or **None**.
- **Font Type.** Indicates the type of font the font definition uses. You can select from **Auto Detect**, **Outline Font**, or **Raster Font**.
 - With **Auto Detect**, automatically detects the font type.
 - With **Outline Fonts**, you must define the horizontal scale which enables you to condense or expand the glyphs in the AFP output horizontally. The default is 1. As an example, a value of 0.5 would condense the glyphs to 50% of their default width.
 - With **Raster Fonts**, you must define the point size of the AFP characters. This applies only to AFP Raster Fonts. For AFP Outline Fonts, the point size should be zero (0).
- **Point Size.** Specifies the size of the AFP character measured in points. This is only applicable to AFP Raster Fonts; for AFP Outline Fonts, the point size should be zero (0).
- **Horizontal Scale.** Applies to outline fonts only. Enables you to condense or expand the horizontal width of the glyphs in the AFP output. Default value is 1. A value of 0.5 would condense the glyphs to 50% of their default width.

Adding an AFP Output Font Family

To add an AFP output font family:

1. On the MFCT tab, select the **AFP Output Font Families** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add AFP Family** dialog.
3. In the **Family name** field, enter the desired name, and click **OK**. The new AFP Font Family Group appears in the AFP Output Font Families table.
4. Select the new **AFP Font Family Group**, and click the **Add** button at the bottom of the table to open the **Add AFP Family Group** dialog.
5. Edit the available output font definitions for the AFP Family group. Either the **Charset Name** and **Codepage** fields, or the **Coded Font** field must be filled out in order to continue, while the defaults, if they are suitable, can remain for the other fields.

The available fields are:

- **Coded font.** Indicates the AFP coded font name.
- **Charset name.** Indicates the AFP character set name.

- **Codepage.** Indicates the AFP codepage name.
 - **ICU table.** Indicates the name of the ICU table.
 - **Encoding.** Specifies the encoding to be used. You can select from Auto or None.
 - **Byte schema.** Indicates whether the font is single byte or double byte.
 - **Font type.** Indicates the type of font the font definition uses. You can select from Auto Detect, Outline Font, or Raster Font.
 - With **Auto Detect**, automatically detects the font type.
 - With **Outline Fonts**, you must define the horizontal scale which enables you to condense or expand the glyphs in the AFP output horizontally. The default is 1. As an example, a value of 0.5 would condense the glyphs to 50% of their default width.
 - With **Raster Fonts**, you must define the point size of the AFP characters. This applies only to AFP Raster Fonts. For AFP Outline Fonts, the point size should be zero (0).
6. When you are finished defining the output font attributes for the font family, click **OK**.

The AFP Output Font Families table appears. Expand the Family Group name to view the output font settings. Multiple output fonts can be added under a Family Group by repeating this procedure as necessary.

Editing an AFP Output Font Family

To edit an AFP output font family:

1. On the MFCT tab, select the **AFP Output Font Families** icon from the MFCT Fonts Sidebar menu.
2. You must make one of the following selections:
 - If you want to change the Family Group name, select the **AFP Font Family Group** you want to change from the AFP Output Font Families table, and click **Edit Resources**, , from the MFCT toolbar.
 - If you want to change the Family Group's font resources, select the **Font Resource** from the AFP Output Font Families table, and click the **Edit** button below the table.
3. Make your desired changes in the **Edit** dialog that appears, and click **OK** to save your changes, or click **Cancel** to discard your changes and exit.

Removing an AFP Output Font Family

To delete an AFP output font family:

1. Select the **AFP Output Font Families** icon from the MFCT Fonts Sidebar menu.

2. Do one of the following:
 - To remove an entire Family Group, select the **AFP Font Family Group** you want to delete from the AFP Output Font Families table, and click the **Remove Resources** button, , from the MFCT toolbar.
 - To remove a Family Group's font resources, select the **Font Resource** from the AFP Output Font Families table, and click the **Remove** button below the table.
3. Confirm the removal of the selected font family or resources. Click **Yes** to confirm the deletion, or **No** to cancel the operation.

AFP Family Builder Wizard

To access this screen: Click the **AFP Family Builder Wizard** button, , on the MFCT toolbar.

The **AFP Family Builder Wizard** is used to scan font resources (coded and character set files) and automatically generate font families based on properties of the character sets.



Note: You are encouraged to use this wizard rather than creating AFP font families manually. The wizard only adds new entries to the table, so it can be used to build the entire table from scratch, or to add new fonts to existing families.

The wizard scans a given file system location for coded font files, which typically contains named references to a codepage and a character set. For each coded font that is found, the associated charset file is opened and the properties are examined to determine its Family Name and its individual attributes, such as byte scheme and point size.

The name of each generated family is based on the following properties:

- The Typeface Name of the charset.
- The charset's font width description, for example Condensed, Expanded, UltraExpanded.
- The byte scheme, single byte or double byte.

When you open the AFP Family Builder Wizard dialog window you can specify a codepage and encoding method to us as default values. These are required for any charsets that are not referenced by any coded fonts, since the coded fonts contain a codepage reference.

By default, there are very few entries, or no entries at all, in the AFP Output Font Families screen.



Note: For more information on the MFCT toolbar, see [“MFCT Toolbar” on page 508](#). For more information on the AFP Output Font Families screen, see [“AFP Output Font Families” on page 537](#).

Adding AFP Output Families Using the AFP Family Builder Wizard

To automatically add AFP output families:

1. Click the **AFP Family Builder Wizard** button,  to open the **AFP Family Builder Wizard** dialog.
2. Enter a **base director**. Click the **Ellipsis** button, , and navigate to the directory that contains the AFP character sets and coded fonts.
3. Select any of the **default options** to set for the new font families. These default options are used when the actual values cannot be determined.
4. To scan sub directories, select the **Scan Sub Directories** check box.
5. Click **OK**. The selected directory is scanned for charsets. Each of these files are parsed, and the family name, point size and typeface style (i.e., bold, italic, etc.) is identified.

Assuming you have valid charsets in the given directory that are not already defined, you should see a lot more charsets in the table.

7.3.5.5.6 HTML Output Font Families

The HTML Output Font Families table displays all of the HTML output font families used in your web project.

The output fonts window contains two columns:

- **Font Family Group**. Indicates the font family name.
- **Font Name**. Indicates the font name.

Adding an HTML Output Font Family

To add an HTML output font family:

1. On the MFCT tab, select the **HTML Output Font Families** icon from the MFCT Fonts Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add HTML Family** dialog.
3. In the **Family name** field, enter the desired name, and click **OK**. The new HTML Output Font Family Group appears in the HTML Output Font Families table.
4. Select the new **HTML Output Font Family Group**, and click the **Add** button at the bottom of the table to open the **Add HTML Family Group** dialog.
5. Edit the output font name for the HTML Family Group. There is only one field to complete:
 - **Font name**. Indicate the HTML font name.

6. When complete, click **OK**.

The HTML Output Font Families table appears. Expand the Family Group name to view the output font name. Multiple output fonts can be added under a Family Group by repeating this procedure as necessary.



Note: For more information on the MFCT Fonts Sidebar menu, see “**MFCT Fonts Sidebar Menu**” on page 509.

Editing an HTML Output Font Family

To edit an HTML Output font family:

1. On the MFCT tab, select the **HTML Output Font Families** icon from the MFCT Fonts Sidebar menu.
2. Do one of the following:
 - If you want to change the Family Group name, select the **HTML Output Font Family Group** you want to change from the HTML Output Font Families table, and click **Edit Resources**, , button from the MFCT toolbar.
 - If you want to change the Family Group’s font resources, select the **Font Resource** from the HTML Output Font Families table, and click the **Edit** button below the table.
3. Make your desired changes in the **Edit** dialog that appears, and click **OK** to save your changes, or click **Cancel** to discard your changes and exit.

Removing an HTML Output Font Family

To delete an HTML font family:

1. On the MFCT tab, select the HTML Output Font Family icon from the MFCT Fonts Sidebar.
2. Do one of the following:
 - If you want to remove an entire Family Group, select the **HTML Output Font Family Group** you want to delete from the HTML Output Font Families table, and click the **Remove Resources** button, , from the MFCT toolbar.
 - If you want to remove a Family Group’s output font resources, select the **Font Resource** from the HTML Output Font Families table, and click the **Remove** button below the table.
3. Confirm the removal of the selected font family or resources. Click **Yes** to confirm the deletion, or **No** to cancel the operation.

7.3.5.5.7 Output Font Mapping Attributes

From the **Output Fonts** table screen, you can select either **Add Resource**, , or **Edit Resource**, , depending on your desired task, to add or modify the attributes. The Add and Edit Output Fonts dialogs contains tabs for all possible output types, however, only those related to your current project will be shown. Select the tab that corresponds to your output type and the output font mapping attributes for that type are displayed. The available output font type tabs are:

- [“TIFF Tab” on page 543](#)
- [“PDF Tab” on page 543](#)
- [“AFP Tab” on page 546](#)
- [“Metacode Tab” on page 546](#)
- [“PCL Tab” on page 547](#)
- [“PostScript Tab” on page 547](#)
- [“HTML Tab” on page 548](#)

TIFF Tab

The following TIFF output font mapping attributes are available:

- **Name.** Enter a name for the TIFF output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Use this font.** Select this option to map a font and style. The following options are available:
 - **Font name.** Choose a font by clicking the ellipsis button, .
 - **Style.** Select from either bold and/or italic styles.
 - **Translation Table.** Choose a Translation table to use from the dropdown menu. For more information, see [“Translation Tables” on page 554](#).

PDF Tab

The following PDF output font mapping attributes are available:

- **Name.** Enter a name for the PDF output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Adobe Base 14/CJK (Standard Acrobat/PDF fonts).** This option allows you to map to the Standard Base 14 fonts or Chinese/Japanese/Korean (CJK) fonts. You must select the font to use from the **Family name** dropdown menu. The valid

Encoding Scheme is selected from the Encoding scheme dropdown menu in the **Options** area below.



Tip: It is recommended that you use fonts from this category whenever possible because they are supported by Adobe Acrobat Reader by default; however, Acrobat may need to prompt the user to allow automatic downloading of the CJK fonts if an Asian language version of Acrobat is not detected. Using these fonts will also ensure that your generated PDF documents have the smallest file size available.

- **Type 1.** This option maps to Type 1 fonts. When Type 1 is selected, you must provide the names of an **Adobe Font Metrics file** (.afm) and a **Postscript Font Binary file** (.pfb), otherwise the PDF Font settings are not accepted. Click the ellipsis button, , beside each field to define their location.



Important

In addition to providing the names of .afm and .pfb files, you also need to properly configure the **FdAfm** and **FdPfb** parameters in your PDF Generator component. These parameters ensure that the PDF Generator can resolve the designated .afm and .pfb names into the actual resource files at runtime.

- **Type 1 (Dynamically created & embedded).** This option maps to the Type 1 compact font format (CFF), which results in a smaller file size for your PDFs compared to files using the equivalent Type 3 fonts.



Tip: Multiple CFF fonts can be consolidated into one single Character Identifier fonts (CID) CFF font to further optimize the output file size. This is controlled by the PDF Generator through the **CffFontConsolidation** parameter. For more information, see *“PDF Generator” on page 203*.

- **Type 3 (Dynamically created & embedded).** This option maps to Type 3 fonts. When Type 3 is selected, the PDF Generator dynamically builds a Type 3 font using character bitmaps. If the original input font is a:
 - **Raster based font**, such as an AFP or Metacode raster font, then the bitmaps from the original fonts will be used in the Type 3 font.
 - **Outline based font**, such as an AFP outline or PDF Type 1 font, then you must also create a mapping for the TIFF output font as TIFF output fonts are used to render the bitmaps for the Type 3 font.

The dynamically built Type 3 font is a subset of the original font it is modeled after, which means that only the characters that appear in the PDF are embedded. The PDF Generator also ensures that the text appearing in the output PDF document remains searchable by embedding the appropriate encoding information into the document.

Using Type 3 fonts is a convenient way to preserve fidelity and often eliminates the need for a more complex font mapping strategy, such as when finding and correlating to appropriate Base 14 fonts. However, there are certain shortcomings with using Type 3 font mappings like:

- Increased file size of the output PDF documents

PDF files created using Type 3 fonts are often more challenging to repurpose in the future, such as when reparsing with PDF Parser to convert to AFP for high-speed printing. This is due to the technical structure of Type 3 fonts in the Adobe PDF specification.

It is, therefore, recommended that whenever using Type 3 font mappings that you consider these limitations.

- **TrueType.** Select this option to use TrueType fonts. When **TrueType** is selected, you must provide a **TrueType font file** (.ttf) otherwise the PDF font settings will not be accepted. Click the ellipsis button, , beside the field to define the location.

! Important

In addition to providing the names of TrueType font file, you also need to properly configure the FdTtf parameters in your PDF Generator component. This parameter ensures that the PDF Generator can resolve the designated .ttf names into the actual resource files at runtime.

- **Use source font.** This option allows you to map the output font to the source font, allowing the source font to be embedded within the PDF output. An error is generated for any invalid source fonts (i.e. any fonts not currently supported by this feature).
- **Options.** The following options are available to be set (available options depend upon the font type selected):
 - **Inter-character adjustment.** Specify a numeric value to determine how many spaces will be added or subtracted between characters in the output font. The default value of zero (0) units does not modify the output font. This can be used to match the width of the input font to the width of the output font. This field is only enabled if you have selected the **Adobe Base 14/CJK** font option above. Selecting any other type of font or checking the **Auto inter-character adjustment** box below disables this field.
 - **Encoding scheme.** Specify the method of encoding you want to use. This option is not available for dynamically created and embedded Type1 or 3 fonts.
 - **Translation Table.** Choose a Translation table to use from the dropdown menu. For more information, see [“Translation Tables” on page 554](#).
 - **Encoding file.** Specify an encoding file by clicking the ellipsis button and navigating to the file path.
 - **Horizontal scale factor.** Indicate a value with which to stretch or condense the width of the characters in the PDF output. 100 is the default value. This option can prevent gaps or overlapping between words.
 - **Auto inter-character adjustment.** Select this option to have the inter-character adjustment automatically calculated by comparing input and output word widths. This is beneficial for documents where the input

font's character widths are substantially different from the output font's widths. This option is valid for Base 14 fonts with WinAnsi encoding, and is enabled only if you have the **Adobe Base 14/CJK** font option selected. Selecting any other type of font, or entering a value in the **Inter-character adjustment** field disables this check box.

- **Embedded.** Indicate whether mapped fonts are embedded. This option is only available for certain font types, such as TrueType.
- **Subsetting.** The Subsetting option is used in conjunction with the **Embedded** option. It indicates that only a subset of the mapped font is to be embedded into the output document. This feature only works with Type 1 (non-compact), Type 3, and TrueType font mappings.



Tip: When subsetting TrueType fonts, you can improve performance in large document breaking batch jobs by configuring the PDF Generator's **TtfSubsetMode** parameter. When set to the default **Document** value, the subset is created with the smallest possible footprint for each document. Using the **Progressive** value adds characters to the shared subset on an as needed basis, while **Full** creates and shares a complete subset based on the input font's characters. The Full and Progressive subset modes are more efficient than the Document mode since the subset is shared between documents and can also be shared between jobs if you cache the original TTF font.

AFP Tab

The AFP output font mapping attributes are the following:

- **Name.** Enter a name for the AFP output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Standard AFP font.** This option allows you to map an AFP font. Select the font family name from the **AFP Font Family** dropdown menu.
- **Open Type font.** This option allows you to use an Open Type font. Enter the name in the Font name field.
- **Options.** The following preferences can be set when using Standard AFP or Open Type fonts:
 - **Translation Table.** If you want to use an AFP Translation table, select the name of the table from the dropdown menu. For more information, see [“Translation Tables” on page 554](#).
 - **Horizontal scale factor.** Enables the condensing or expansion of the horizontal width of the glyphs in the AFP output. The default value is 1. A value of 0.5 would condense the glyphs to 50% of the standard width.

Metacode Tab

The following Metacode output font mapping attributes are available:

- **Name.** Enter a name for the Metacode output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Use this font.** Select this option if you want to map a Metacode font. You can map it with the following fields:
 - **Metacode Family name.** Select a font family from the dropdown menu.
- **Options.** There is only one option available for Metacode output fonts:
- **Translation Table.** From the dropdown menu, you can select a Translation table to use. For more information, see [“Translation Tables” on page 554](#).

PCL Tab

The following PCL output font mapping attributes are available:

- **Name.** Enter a name for the PCL output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Printer resident font.** This option allows you to use a font located on a PCL printer. The font attributes fields need to be defined: Family ID, Symbol set, Style, Weight and whether it should appear in Fixed or Proportional spacing.
- **External font.** This option allows you to use a font located in an external file. Click the ellipsis button, and navigate to the location of the font file.
- **Options.** There is only one option available for PCL output fonts:
 - **Translation Table.** From the dropdown menu, you can select a Translation table to use. For more information, see [“Translation Tables” on page 554](#).
- **Use Original Codepoints.** Select this check box to preserve the input file's original EBCDIC characters in the PCL output font.

PostScript Tab

The following PostScript output font mapping attributes are available:

- **Name.** Enter a name for the PostScript output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Use this font.** Select this option if you want to map a Postscript font. You can map it with the following field:
 - **Typeface.** Indicate the Postscript font with which you want to map in your project.

- **Options:** There are two options available for Postscript output fonts:
 - **Translation Table.** From the dropdown menu, you can select a Translation table to use. For more information, see [“Translation Tables” on page 554](#).
 - **Encoding.** From the dropdown menu, you can indicate the type of encoding you wish to employ; **ISOLatin1Encoding** or **StandardEncoding**.

HTML Tab

The following HTML output font mapping attributes are available:

- **Name.** Enter a name for the HTML output font.
- **Raster.** Select this option if you want your font to be placed into the output as an image; however, be aware that this can dramatically increase the size of your output file.
- **Use this font.** Select this option to map an HTML font. The following options are available:
 - **HTML Family name.** Select a font family from the dropdown menu.
 - **Translation Table.** From the dropdown menu, you can select an HTML Translation Table to use. For more information, see [“Translation Tables” on page 554](#).
 - **Is Unicode.** Select this check box if the Translation Table you want to use is Unicode.

7.3.6 MFCT Colors Sidebar Menu

The **Colors** sidebar menu contains several options:

	Description
Color Table 	Color Table. Displays a color table showing the names and red, green and blue (RGB) values of colors used in the project.
AFP Color Tables 	AFP Color Tables. Displays the color tables that have been defined within your AFP font correlation table.
Metacode Color Tables	Metacode Color Tables. Displays the color tables that have been defined within your Metacode font correlation table.

7.3.6.1 Color Table

The **Color Table** displays a table showing the names and red, green and blue (RGB) values of colors used in the project. The table has several columns:

- **Color.** Indicates the approximate color for the particular definition and also the color name. The color names are not case sensitive, and can be up to 32 characters long with no spaces.



Note: The fidelity of the colors, as seen on your monitor, may be different from the color as it appears when printed due to variances in displaying colors in various media. As long as your color values have been defined correctly, the output should produce the color you expect. If you have additional questions about getting correct colors in your output, please contact customer support.

- **Red.** Specifies the amount of red that is in the color, as indicated by a value from 0-255.
- **Green.** Specifies the amount of green that is in the color, as indicated by a value from 0-255.
- **Blue.** Specifies the amount of blue that is in the color, as indicated by a value from 0-255.

Color Table			
Color	Red	Green	Blue
BLACK	0	0	0
BLUE	0	0	255
RED	255	0	0
FUSCHIA	255	0	255
LIME	0	255	0
AQUA	0	255	255
YELLOW	255	255	0
WHITE	255	255	255
NAVY	0	0	128
DARKBLUE	0	0	170
ORANGE	255	128	0
PURPLE	128	0	128
AFPPURPLE	170	0	170
GREEN	0	128	0
DARKGREEN	0	146	0
OLIVE	128	128	0
DARKTURQUOISE	0	146	170
MUSTARD	255	204	0
AFPMUSTARD	196	160	32
GRAY	190	190	190
AFPGRAY	131	131	131
BROWN	255	128	64
AFPBROWN	144	48	0

Figure 7-3: Color Table

7.3.6.1.1 Adding a Color Table

Every project has a color table with a set of default colors already defined in the table. If you do not want to use the default colors table, you have the ability to add your own color table.



Note: There can only be one instance of a Metacode Color Table. The Metacode Color Table cannot be renamed from Default, but the color resources on the Default table can be edited to suit your requirements.

To add a color table:

1. On the AFP Color Table screen, click the **Add Resource** button, , on the MFCT toolbar to open the **Add Table** dialog.
2. Enter a name for your table in the **Table name** field, and click **OK**.

The new Color Table is added to your project.

For information on adding a color to input data type color tables, see the topic appropriate to your data type:

- [“Adding a Color to an AFP Color Table” on page 552](#)
- [“Adding a Color to a Metacode Color Table” on page 553](#)

7.3.6.1.2 Adding a Color to a Color Table

Every project has a color table with a set of default colors already defined. If you do not want to use the default colors or they do not meet your requirements, you can add a color to a previously defined color table.

1. To add a color to a color table:
2. On the MFCT tab, select the **Color Table** icon from the MFCT Colors Sidebar menu.



Note: For more information about the MFCT Colors Sidebar menu, see [“MFCT Colors Sidebar Menu” on page 548](#).

3. Click the **Add Resource** button, , on the MFCT toolbar to open the **New Output Color** dialog.
4. Enter the desired name for the new color definition in the **Color name** field.
5. Select the color parameters. There are three different ways to perform this task:
 - On the **Swatches** tab, select the color from the predefined color swatches.
 - On the **HSB** (Hue, Saturation, and Brightness) tab, move the slider on the Luminosity bar, and click on an area of the Color Chart to choose a new color. This method offers a more accurate color selection process.
 - On the **RGB** tab, use either the sliders for each color or enter a value into the field to complete a set of RGB values.

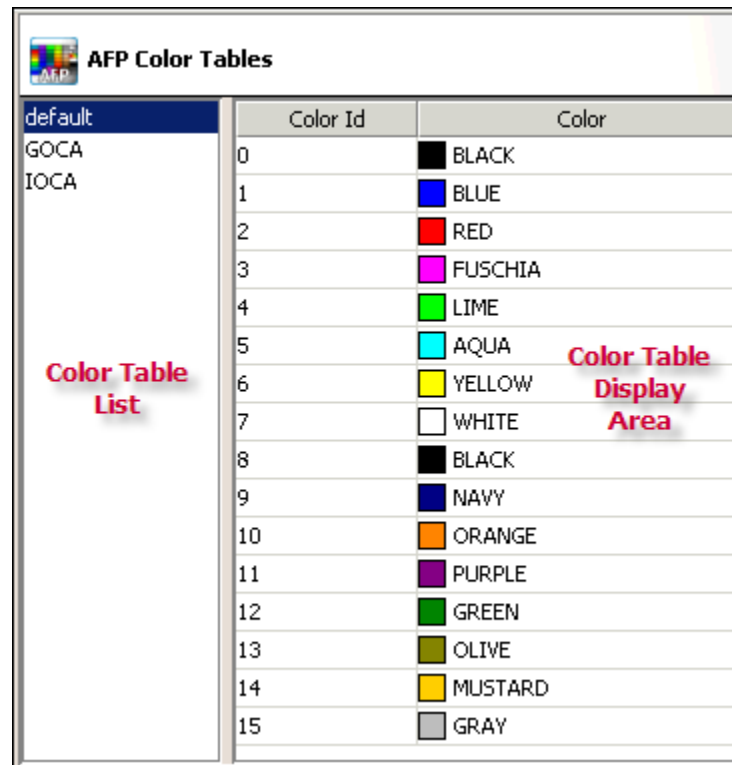
The **Preview** area at the bottom will change dynamically to show you an example of the currently selected color.

6. When you have defined your desired color, click **OK**.

7.3.6.2 AFP Color Tables

The **AFP Input Color Tables** screen lists all of the color tables that have been defined in your project. You can access this table by clicking **AFP Input Color Tables** icon on the MFCT Colors Sidebar menu. The table has two areas:

- **Color Table List.** A list of the names of the color tables associated with your current project.
- **Color Table Display Area.** The Color Table itself. Select a table from the Color Table list area, and the Color Table is shown in this pane.



AFP Color Tables		
	Color Id	Color
default	0	BLACK
GOCA	1	BLUE
IOCA	2	RED
	3	FUSCHIA
	4	LIME
	5	AQUA
	6	YELLOW
	7	WHITE
	8	BLACK
	9	NAVY
	10	ORANGE
	11	PURPLE
	12	GREEN
	13	OLIVE
	14	MUSTARD
	15	GRAY

Figure 7-4: AFP Color Table

The Color Table display area has the following columns:

- **Color ID.** The ID number associated with the color.
- **Color.** Indicates the approximate color for the particular definition and also the color name. The color names are not case sensitive, and can be up to 32 characters long with no spaces.



Note: For more information about the MFCT Colors Sidebar menu, see “MFCT Colors Sidebar Menu” on page 548.

7.3.6.2.1 Adding a Color to an AFP Color Table

To add a color to an AFP Color Table:

The table already contains separate Color IDs. Select one of the cells in the **Color** column and from the dropdown menu that appears, select the color that you want to correlate to the particular color ID. If you want to create a new custom output color, select {New} from the dropdown menu to produce the color.

The color is added to the AFP Color Table.

7.3.6.3 Metacode Color Tables

The **Metacode Color Tables** window lists all of the color tables that have been defined in your project. The table has two areas:

- **Color Table List.** A list of the names of the color tables associated with your current project.
- **Color Table display area.** The Color Table itself. Select a table from the Color Table list area, and the Color Table is shown in this pane.

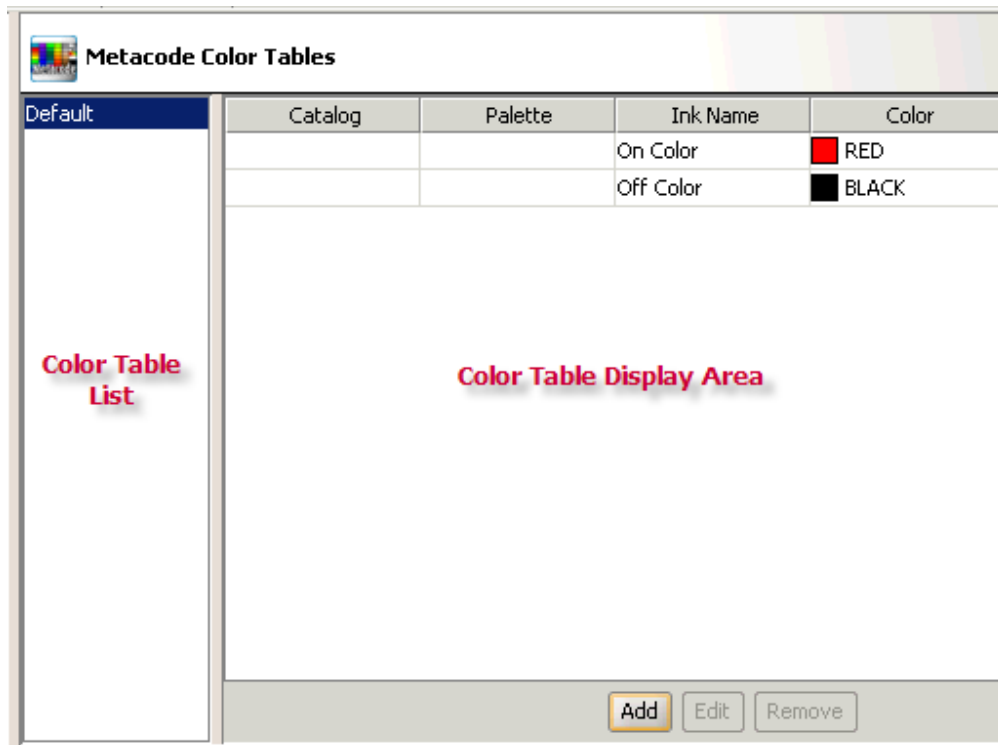


Figure 7-5: Metacode Color Tables

The Color Table display area has the following columns:

- **Catalog.** Specifies the name of the main index of colors in use.
- **Palette.** Indicates the name of the core color.
- **Ink Name.** Specifies whether the printer's highlight colors are on or off. Valid values are **On Color** and **Off Color**.
- **Color.** Indicates the approximate color for the particular definition and also the color name. The color names are not case sensitive, and can be up to 32 characters long with no spaces.



Note: An asterisk in the **Catalog** or **Palette** fields allows for all applicable catalogs and color entries to be used.

7.3.6.3.1 Adding a Color to a Metacode Color Table

To add a color to a Metacode Color Table:

1. On the Metacode Color Table screen, click the **Add** button located below the table to open the **Add Metacode Color** dialog.
2. Complete the following fields:
 - **Catalog.** Specifies the name of the main index of colors in use.
 - **Palette.** Indicates the name of the core color.
 - **Ink Name.** Specifies whether the printer's highlight colors are on or off. Valid values are **On Color** and **Off Color**.
 - **Color.** Indicates the approximate color for the particular definition and also the color name. The color names are not case sensitive, and can be up to 32 characters long with no spaces.



Note: An asterisk in the **Catalog** or **Palette** fields allows for all applicable catalogs and color entries to be used.

3. When complete, click **OK**.

The new color is added to the Metacode Color Table.

7.3.7 MFCT Translations Sidebar Menu

The  sidebar menu contains the following options:

	Description
Translation Tables 	Translation Tables. Displays a table showing the hex value of the output character you match to each input character. You can rasterize an individual character, if it is not contained in your output font; and you can build and name many different translation tables, for many different font combinations.
AFP Glyph Table 	AFP Glyph Table. Displays a table showing the UNICODE mapping of a glyph. A glyph name causes custom encoding for PDF formats.
Codepage ICU Table 	Codepage ICU Table. Displays a table of AFP codepages that have been overridden and are not using the default ICU table. For more information, see “ICU Tables” on page 505 .

7.3.7.1 Translation Tables

The **Translation Tables** screen displays a registry indicating the hex value of the output character you matched to each input character. You can rasterize an individual character if it is not contained in your output font; and you can build and name different translation tables for different font combinations. You can view the Translation Tables by selecting **Translation Tables** on the MFCT Translations Sidebar menu. For more information about this menu, see [“MFCT Translations Sidebar Menu” on page 554](#).

The Translation Tables screen provides you with character-level control over your font correlation by permitting the following functions:

- **Map a Glyph** by specifying the hex value of the output glyph you want to match to each input character.
- **Rasterize an individual glyph**, if it is not contained in your output font. An individual character may have to be rasterized when it cannot be mapped to any known output font such as a line draw character. This may happen if the input font contains custom characters such as a signature or a logo.
- **Build and name many different translation tables** for different font combinations.

Glyph mapping is used to translate into the Unicode equivalent. If the input font's code point does not match the output font's code point then this can be tweaked by use of a translation table. These tables map Unicode to output font, whereas the glyph mappings map input font to Unicode.

The translation table has two areas:

- **Translation Table List.** A list of the names of the translation tables associated with your current project.
- **Translation Table display area.** The Translation Table itself. Select a table from the Translation Table list area, and the Translation Table is shown in this pane.

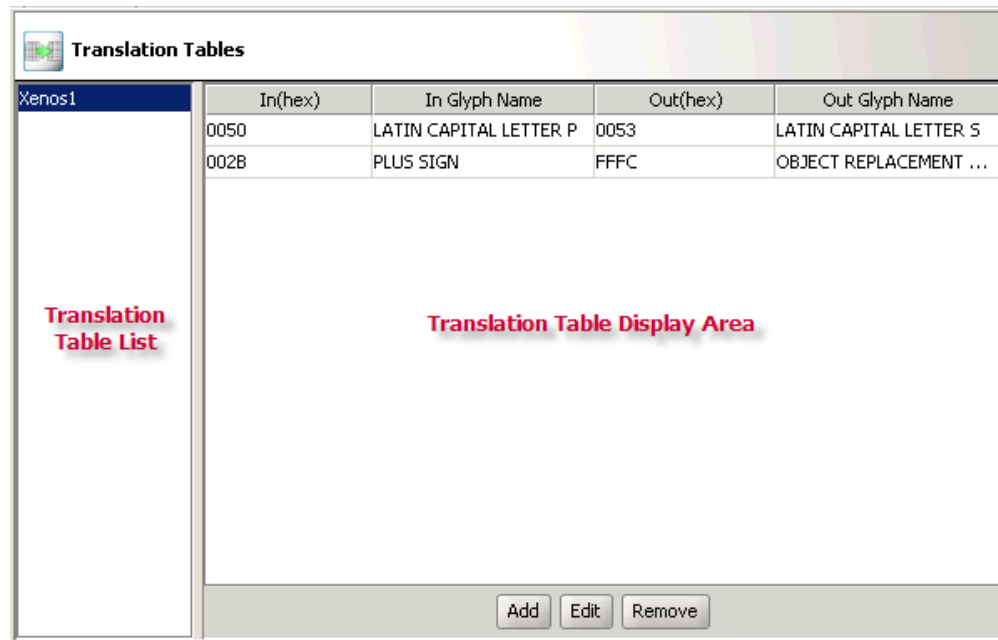


Figure 7-6: Sample Translation Table

The translation table contains four columns:

- **In(hex).** Displays the input hex value of the glyph.
- **In Glyph Name.** Displays the input name of the glyph being mapped.
- **Out(hex).** Displays the output hex value of the glyph.
- **Out Glyph Name.** Displays the output name of the glyph being mapped.

7.3.7.1.1 Adding a New Translation Table

To add a translation table:

1. On the MFCT tab, select the **Translation Tables** icon from the MFCT Translation Sidebar menu.
2. Click **Add Resource**, , on the MFCT toolbar to open the **Add Table** dialog.
3. In the **Name** field, enter a name for the new translation table.
4. Click **OK** to create your new translation table.



Note: For information on adding a translation, see [“Adding a Translation” on page 556](#).

For more information on the MFCT Translations Sidebar menu, see [“MFCT Translations Sidebar Menu” on page 554](#).

7.3.7.1.2 Removing a Translation Table

To remove a translation table:

1. On the MFCT tab, select the **Translation Tables** icon from the MFCT Translation Sidebar menu.
2. From the tables list, select the name of the Translation Table you want removed, and click **Remove Resource**, , to open the **Remove Translation Table** dialog appears.
3. You must confirm the removal of the selected table. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the MFCT Translations Sidebar menu, see [“MFCT Translations Sidebar Menu” on page 554](#).

7.3.7.1.3 Adding a Translation

To add a translation:

1. On the Translation Table screen, click the **Add** button below the table to open the **Add Translation** dialog.
2. Complete the following fields:
 - **Input Unicode.** Enter the Unicode value for the glyph in your input. You can also click the ellipsis button, , to open the Unicode Browser, and find the Unicode value for a specific glyph.
 - **Rasterize and Map to.** Select how you want the glyphs to be processed. Select **Rasterize** if you want to rasterize the glyph as a bitmap, or **Maps to** if you want to map the input glyph to a specific output glyph.

- *<Output Unicode.>* Available only when **Map to** is selected. Enter the UNICODE value for the glyph in your output. You can also click the ellipsis button, , to open the Unicode Browser and find the Unicode value for a specific glyph.
3. Click **OK** to accept the values and create the new translation.



Note: For more information on the Unicode Browser, see “Unicode Browser” on page 557.

7.3.7.1.4 Editing a Translation

To edit a translation:

1. On the Translation Table screen, click the translation you want to edit, and select the **Edit** button below the table to open the **Edit Translation** dialog.
2. The fields show their current values, which you can edit to your desired settings.
3. Click **OK** to accept the new values, or **Cancel** to exit without making any changes.

7.3.7.1.5 Removing a Translation

To remove a translation:

1. On the **Translation Table** screen, select the translation you want to delete.
2. Click **Remove** at the bottom of the table to open the **Remove Translation** dialog.
3. You must confirm the removal of the selected font. Click **Yes** to confirm the deletion, or **No** to cancel the operation.

7.3.7.1.6 Unicode Browser

The Unicode Browser displays the Unicode value of a glyph under a graphical representation of that glyph. Glyphs that are unable to be graphically represented on screen are represented by an empty box.

You can view the Unicode Browser from either the Add Translation or Edit Translation dialog windows by clicking the **ellipsis**, . For more information about these dialogs, see “Adding a Translation” on page 556 or “Editing a Translation” on page 557.

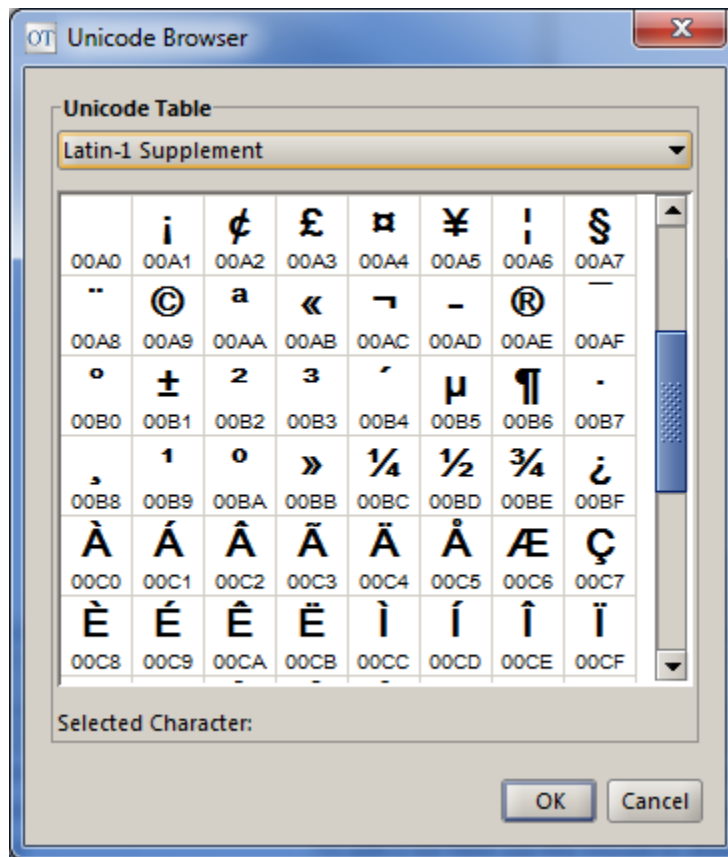


Figure 7-7: Unicode Browser dialog box

To select a glyph:

1. Select a character set from the dropdown menu located at the top of the Unicode Browser.
2. Click the glyph you wish to map.
3. Click **OK** to accept the selection.

The Unicode Browser closes and the Unicode value of the selected glyph is automatically entered into the field on the dialog window from where the Unicode Browser was opened.

7.3.7.2 AFP Glyph Table

The Glyph Table is used to map glyphs for instances where the input encoding does not automatically correspond with an output encoding. When a project is initially set up, you should use the glyph mapping feature to verify that the input encoding has a suitable output encoding counterpart; if not, you must manually adjust each glyph so that it matches, or select matching ICU Tables if applicable. You can access the AFP Glyph Table by selecting **AFP Glyph Table** icon on the Translations Sidebar menu. For more information about this menu, see “[MFCT Translations Sidebar Menu](#)” on page 554.



Note: Glyph mapping is not required if your output is TIFF, PNG, or AFP.

The AFP Glyph Table displays three columns:

- **AFP Glyph name.** Indicates the name of the AFP Glyph.
- **Unicode Mapping.** Indicates the Unicode value to which the AFP Glyph has been mapped.
- **Force Raster.** Specifies whether the AFP Glyph is forced to be a raster image. It is a **True** or **False** value.

7.3.7.2.1 Adding a Glyph Mapping

To add a glyph to the AFP Glyph Table:

1. On the **AFP Glyph Table** screen, click **Add Resource**, , on the MFCT toolbar to open the Add AFP Glyph Mapping dialog.
2. Complete the following fields:
 - **AFP Glyph Name.** Indicates the name of the AFP Glyph.
 - **Unicode Mapping.** Indicates the Unicode value to which the AFP Glyph is being mapped.
3. With the **Force Raster** check box, specify whether you want the AFP Glyph to be a raster image.
4. Click **OK** to accept the values and create the new mapping.



Note: You can also map glyphs from the **Map Glyph Fonts** tab of the Job Manager window.

For more information on the AFP Glyph table screen, see “[AFP Glyph Table](#)” on page 559. For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.7.2.2 Editing a Glyph Mapping

To edit a glyph in the AFP Glyph Table:

1. On the **AFP Glyph Table** screen, click the mapping you want to change, and select **Edit Resource**, , on the MFCT toolbar to open the **Edit AFP Glyph Mapping** dialog.
2. The fields show their current values, which you can edit to your desired settings.
3. Click **OK** to accept the new values, or **Cancel** to exit without making any changes.



Note: For more information on the AFP Glyph table screen, see “[AFP Glyph Table](#)” on page 559. For more information on the MFCT toolbar, see “[MFCT Toolbar](#)” on page 508.

7.3.7.2.3 Removing a Glyph Mapping

To remove a glyph in the AFP Glyph Table:

1. On the **AFP Glyph Table** screen, select the mapping you want to delete.
2. Click **Remove Resource**, , on the MFCT toolbar to open the **Remove AFP Glyph Mapping** dialog.
3. You must confirm the removal of the selected mapping. Click **Yes** to confirm the deletion, or **No** to cancel the operation.



Note: For more information on the AFP Glyph table screen, see “[AFP Glyph Table](#)” on page 559.

7.3.8 Finding Unicode Values in MS Windows



Note: The following instructions are for Windows XP, however, the procedure for other versions of the Windows operating system will be similar. Follow these directions as a general guideline.

This can be useful when setting up Translation Tables.

To find a Unicode value using Microsoft Windows:

1. Click the **Start** button on the Windows taskbar.
2. Navigate to **All Programs > Accessories > System Tools > Character Map** to open the **Character Map** application.
3. Choose a font typeface from the **Font** dropdown menu.
4. Select a character from the map.

The Unicode value (and the Windows shortcut, if available) is displayed in the status bar at the bottom of the window.

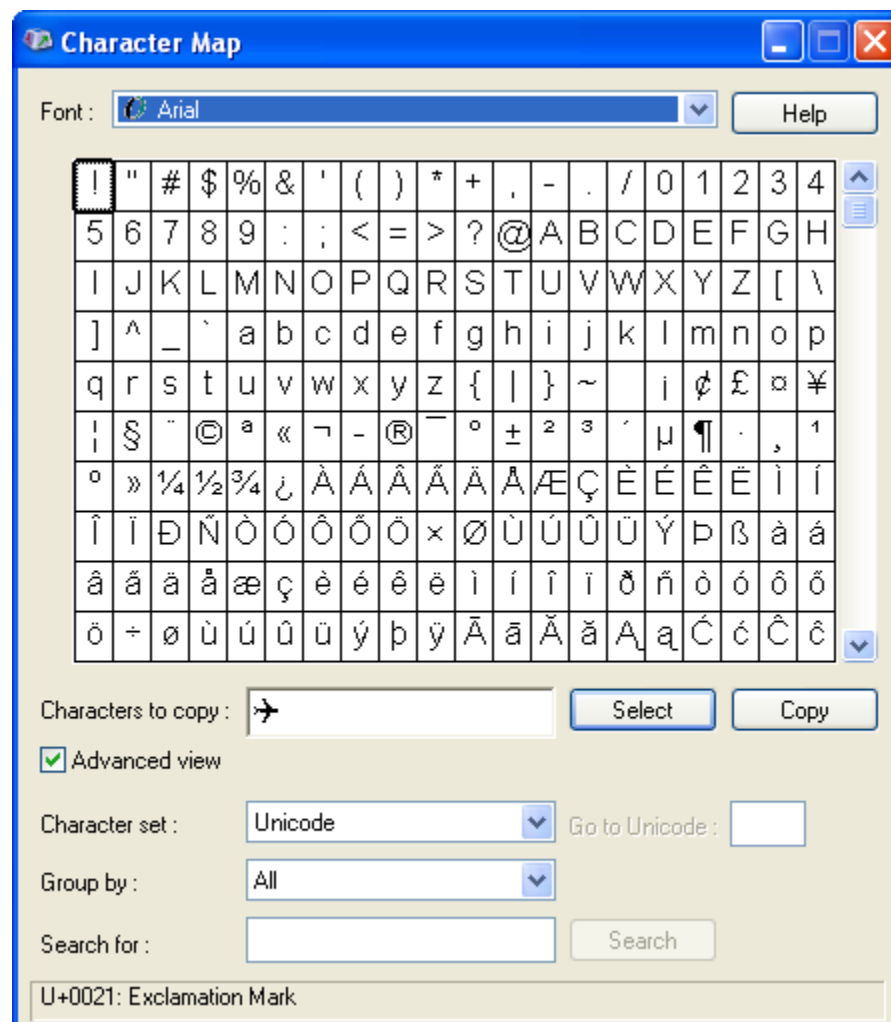


Figure 7-8: Windows Character Map

7.3.9 Manage Fonts

The Manage Fonts tab on the MFCT screen allows you to edit fonts needed for the current project, and can be thought of as a condensed version of the MFCT. It displays only the fonts used in the current project. Attributes of the font mappings can be altered without opening the MFCT window.



Note: The displayed data does not refresh automatically when a new input file is selected. Either close and reopen the MFCT window, or click to another tab of the MFCT window and then back to this tab to refresh the contents to reflect the new input data.

The Manage Fonts tab has several columns:

- **Typeface/Font Family.** Indicates the name of the input AFP typeface or Metacode font family. Expand the group by clicking the + symbol to the left of the name to display the different character sets.
- **Style.** Indicates the style of the font:
 - **B** – Bold
 - **I** – Italic
 - **BI** – Bold Italic
- **Font Ratio.** Specifies a multiplier to calculate the final font size with. This value will be applied to all output font types mapped to this font. A value less than 1.0 should be used if your output documents have overlapping text, or a value greater than 1.0 if the text in your output documents appear smaller than desired.
- **Output Color.** Indicates the color to be used in the output for that font. If a color table is to be used then the name of the color table is displayed. Click on the cell to edit the value.
- **Output Font.** Indicates the name of the output font mapped to that typeface font. To edit the font, click on the cell and select one from the dropdown menu.
- **TIFF.** Reveals the font to be used in TIFF output.
- **PDF.** Reveals the font to be used in the PDF output.
- **AFP.** Reveals the font to be used in AFP output.
- **Metacode.** Reveals the font to be used in Metacode output.
- **PCL.** Reveals the font to be used in PCL output.
- **Postscript.** Reveals the font to be used in Postscript output.
- **HTML.** Reveals the font to be used in HTML output.
- **Trans Table.** Reveals the name of the translation table.



Note: Not all columns appear with every combination of input and output types.

7.3.10 Map Glyph Fonts

The Map Glyphs tab gives you powerful character-level control over your font correlation. It enables you to do the following:

- Identify the hex value of the Unicode point you want to match to each input character.
- Rasterize an individual character if it is not contained in your output font.

You can access the Map Glyphs tab from the MFCT screen. The Map Glyphs tab has several aspects:

- **Font.** The drop-down menu contains the fonts you can select from. Information on your chosen font (Typeface, Font style and type, Byte size, Mapping, and Point size) is displayed to the right.
- **Encoding:**
 - If your input is **AFP**: The available ICU tables are determined by the FdIcuTables location specified by the FdIcuTables parameter in the JobManager section of the .d2esys file. By default, the ICU tables are in `<install_home>/initialFiles/common/_sample/OutputTransformation/applications/sample/icutables`.
 - **ICU.** From the dropdown menu, select the name of the ICU table you want to use.
 - **Default ICU.** Displays the default ICU defined in the code page.
 - If your input is **Metacode** or **PDF**: From the list of available tables in the drop-down menu, select the Translation Table you want to use.
Be aware that the font must be mapped to an output font before the encoding can be changed.
- **Glyph Mappings Table.** The Glyph Mappings table displays the AFP glyph mappings, with each row representing a different AFP glyph.

The table contains several columns:

- **Mapped.** Indicates, with a **True** or **False** value, whether the AFP glyph has been mapped to a Unicode Glyph.
- **Used.** Indicates, with a **True** or **False** value, if the AFP glyph is used in the current project.
- **Available.** Indicates, with a **True** or **False** value, whether the AFP glyph is available in code page. If it is false, then the Code Point value will be undefined, and appears as (FFFF).
- **Glyph.** Displays an image of the AFP glyph.
- **Glyph Name.** Indicates the name of the AFP glyph from the AFP Glyph Table.

- **Code Point.** Specifies the hexadecimal value from the code page that represents the AFP glyph.
- **Unicode Glyph.** Displays the Unicode glyph image.
- **Unicode ID.** Indicates the ID number of the Unicode glyph from the Unicode Table.
- **Force Raster.** Indicates, with a **True** or **False** value, whether the AFP glyph should be rasterized in the output.
- **Unicode Name.** Specifies the Unicode glyph name to which the AFP glyph is mapped.
- **Changed.** Indicates whether or not a particular mapping has been altered. When the mapping has been modified, this value changes to **True**.
- **Unicode Table.** Displays the Unicode Glyph Table mappings. The drop-down menu contains a list of available Unicode tables that you can select.

7.3.10.1 Glyph Mappings Controls

The buttons below the Glyph Mappings Table are:

- **Search.** Searches for a glyph in the table by either: Glyph/Unicode name or ID.
- **Filter.** Launches the **Glyph Mapping Search Filter** to customize which glyph mappings are displayed in the Glyph Mappings Table. The items you can choose to filter by are **Mapped**, **Used**, **Available**, **Has Raster**, and **Forced Raster**. Select **True** or **False** to determine whether you want the particular item(s) shown in your table.



Tip: If you select **Mapped** and **True**, then all glyphs that are mapped will appear in the table. Select **False**, and only unmapped glyphs will show. Any combination of filter options may be used.

- **ICU Best Fit.** Searches ICU Tables for the one that is the best fit to your glyph mappings. To perform a search:
 1. On the **Map Glyphs** tab of the MFCT, click the **ICU Best Fit** button.
The **ICU Best Fit** dialog appears.
 2. Confirm the start of the search by clicking **Yes**.
 3. A list of ICU tables appears, showing the most appropriate tables. Beside the file path of the table is the percentage that the ICU table matches your glyph mappings. Select one from the list, and click **OK**.



Tip: The more glyphs you map, the more precise your search results. If no tables return with a 100% match, then map some more glyphs until you find a table with a satisfactory match percentage.

7.4 Font Environment Map

The **Font Environment Map** is a project-level component designed to simplify the complex management of font resources for users.

The Font Environment Map is stored in an XML file with the file extension `.d2efontenv` and is linked by the project, `.d2eproject`, in the same manner as Master Font Correlation Table (MFCT) files, `.d2emfct`. While the MFCT and the Font Environment Map are both XML files that provide font configuration information to the project at runtime, there are key differences.

The MFCT is primarily a mapping table, which describes how input fonts used by parser components are correlated to output fonts used by generator components. In other words, the MFCT describes the relationships between particular fonts that are to be observed by the transform at runtime. This mechanism provides very powerful and robust functionality to ensure users are able to achieve maximum fidelity and optimal presentation in all document types generated by .

On the contrary, the Font Environment Map does not describe relationships between fonts. Its primary function is to allow users to graphically index the fonts in their environment using a user-friendly interface. This helps eliminate some of the more complex steps involved in building the MFCT resources.

In brief, the MFCT demonstrates relationships between input and output fonts, whereas the Font Environment Map shows the fonts themselves, but does not show the relationship between the fonts.

The Font Environment Map is accessed from the **Transformation Projects** tab via the Font Environment Map node. Right-clicking on the node brings up a shortcut menu with the following options:

- **Add Font Environment Map.** Adds a new Font Environment Map to the project. The Select Resource Path/File dialog appears where you can select the map you wish to add to your project.
- **Save Active Font Environment Map As.** Saves a copy of the current Font Environment Map. Enter a file path and name in the dialog window that appears.
- **Show Active Font Environment Map.** Launches the Font Environment Map tab in the Development window, showing the current map.
- **New Default Font Environment Map.** Sets up a new Font Environment Map for your project. The Select Resource Path/File dialog appears where you can enter the location and name to save your new map under.



Note: For more information on the Transformation Projects tab, see [“Transformation Projects Tab” on page 83](#).

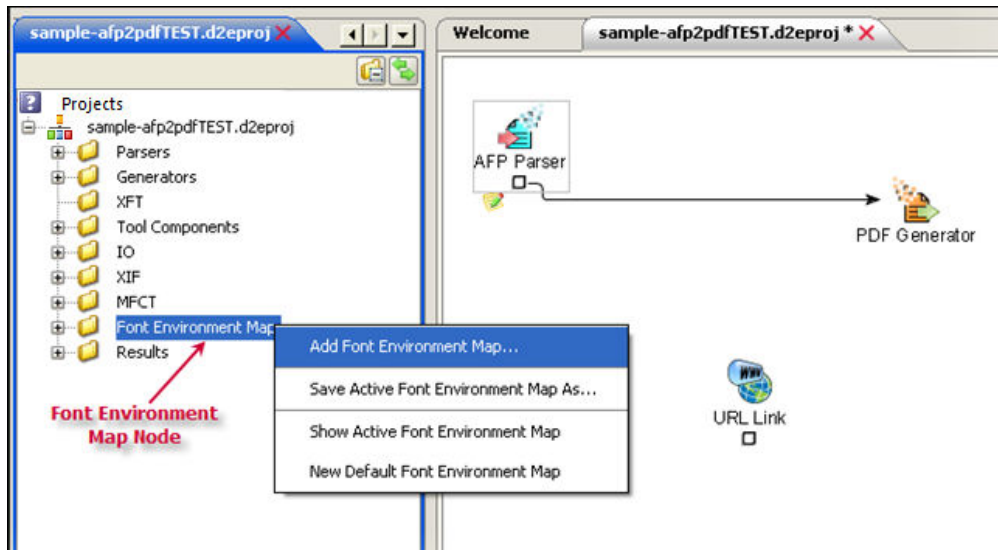


Figure 7-9: Font Environment Map node on Transformation Projects tab

Once you create and activate a Font Environment Map file (.d2efontenv), you can double-click on it to open the Font Environment Map window.

The Font Environment Map consists of three components:

- Catalog Manager
- Repository Manager
- Font Convert Tool

7.4.1 Catalog Manager

The Catalog Manager is comparable to a font catalog as it contains an index of font entries. Each entry has information about an individual font in the project environment, and is comprised of the following elements:

- Font name.
- File names of associated resources files, such as `times.ttf`.
- Font descriptor attributes, such as Serif/Sans Serif, Fixed/Proportional, Bold, and Italic.

Keep in mind that a catalog entry describes the properties of a font, but does not describe its location. Although the catalog entries contain the names of the resource files for the actual font itself, it does not contain any path information to indicate where the font is located.



Note: The Font Environment Map supports individual catalogs for each compatible font type: OpenType, TrueType, Adobe Type 1, AFP, and Metacode

Fonts. However, at this time, the graphical interface only supports OpenType fonts for use with the AFP Parser.

Operations within the Catalog Manager can all be performed from the default screen:

- Import Folder
- Import Fonts Wizard
- Catalog Table

7.4.1.1 Importing a Folder

The **Import Folder** feature on the Font Environment Map screen allows you to quickly import a folder containing fonts. It launches a file chooser dialog window which enables you to select a folder for import. This folder is scanned for all supported font file types and catalog entries are created for fonts that do not already exist in the catalog, while repository entries are created for the folder.

7.4.1.2 Using the Import Fonts Wizard

The **Import Fonts Wizard** guides you through the procedure for importing specific fonts and folders into the Font Environment Map, and is launched by selecting **Import Wizard**. It also provides customizable options for creating font catalog and repository entries.

The steps of the Import Fonts Wizard are:

1. Choosing the fonts files and/or directories to import
2. Choosing import options in the Import Fonts Wizard

7.4.1.2.1 Choosing the fonts files and/or directories to import

To add font files and/or directories:

1. In the Font Environment Map on the Catalog Manager screen, click **Import Wizard** to open the **Import Fonts Wizard**.

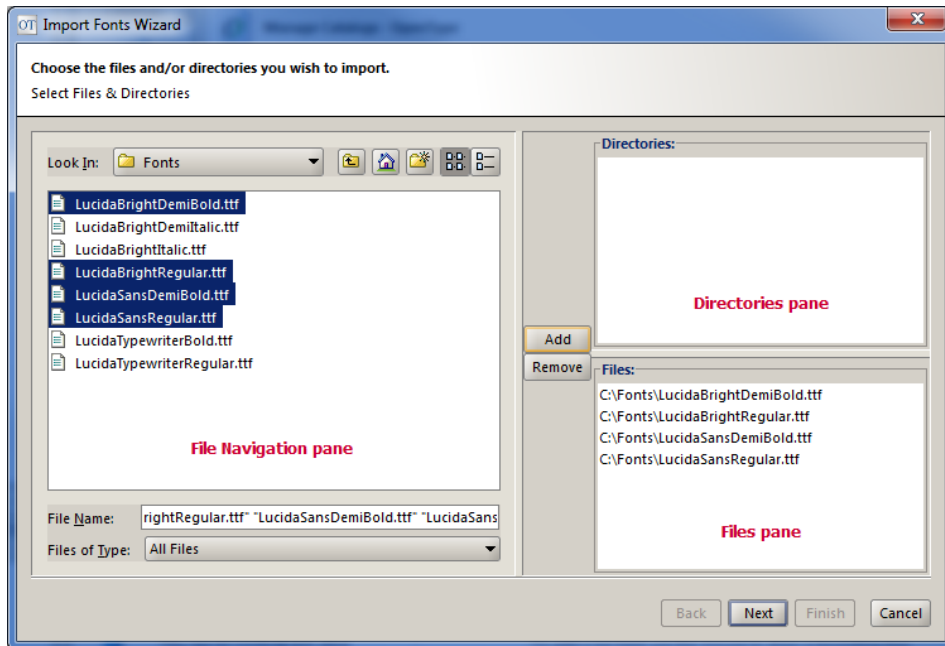


Figure 7-10: Import Fonts Wizard screen

2. From the **File Navigation** pane on the left, navigate to and select the font files and/or directories you want to import. You can use CTRL + click to select several individual files or directories at the same time.
3. Click the **Add** button to add the selected files and directories to the right-hand panes. Your selected directories are added to the Directories pane, while the font files are added to the Files pane.
4. When you have finished making your selections, click **Next**.

7.4.1.2.2 Choosing import options in the Import Fonts Wizard

1. Select the desired import options:

Folder Options:

 - **Create repositories.** Select this option to create repository entries. You must also indicate whether to **Include only folders containing font resources**, or to **Include all folders (even empty ones)**.
 - **Recurse into subfolders.** Select this option to have all subfolders imported as well.

File Options:

 - **Import font resources.** Select this option to import specific font resources, and then indicate the font types to import with the check boxes below.



Note: The Font Environment Map supports individual catalogs for each compatible font type: OpenType, TrueType, Adobe Type 1, AFP, and Metacode Fonts. However, at this time, the graphical interface only supports OpenType fonts for use with the AFP Parser.

2. When you have finished selecting your options, click **Finish**.

The wizard begins the importing process, and provides a dialog box showing its progress. When complete, the Catalog Table appears. For more information, see [“Catalog Table” on page 569](#).

7.4.1.2.3 Removing font files and/or directories from the Import Fonts wizard

1. In the Import Fonts wizard, select the font files and/or directories you no longer want to import from their respective panes on the right side.
2. Click the **Remove** button to delete them from the list.

7.4.1.3 Catalog Table

After you have imported your font resources using the Import Folder or Import Wizard methods, the **Catalog Table** will be populated. The table provides the font names as well as columns representing some of the more common font descriptor attributes for each font. Additionally, a preview of the font’s appearance is shown in the area underneath the table; simply click on a font and the sample automatically appears.



Note: For more information on the Import Folder, see [“Importing a Folder” on page 567](#). For more information on the Import Wizard, see [“Using the Import Fonts Wizard” on page 567](#).

You can remove fonts for the Catalog Table by selecting them, and clicking the **Delete Selected** button.

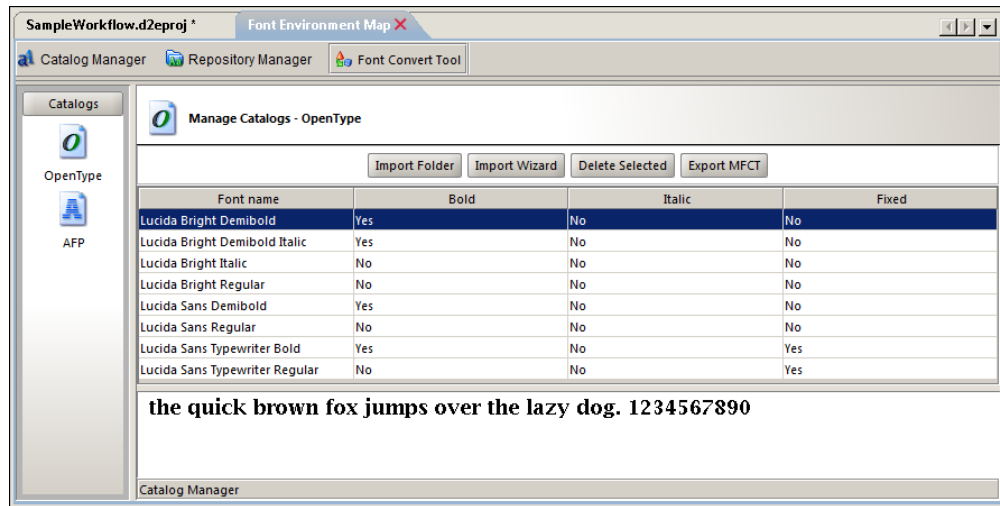


Figure 7-11: Catalog table with fields populated and preview of selected font showing

7.4.2 Repository Manager

The **Repository Manager** presents the physical location where font resources are located and consists of two properties:

- A file path or directory location.
- A listing of font file types that the location contains, such as OpenType, TrueType, Adobe Type 1, AFP, and/or Metacode Fonts.

To open the Repository Manager, on the Font Environment Map screen, select **Repository Manager**. From the Repository Manager, the following functions can be performed:

- Adding a Repository
- Editing a Repository
- Removing a Repository

7.4.2.1 Adding a Repository

To add a new repository:

1. In the Font Environment Map on the Repository Manager screen, click **Add New** to open the **Add Repository** dialog.
2. Enter the folder location in the **Repository Location** field. You may click on the ellipsis button to aid in this process.
3. When you are finished making your selection, click **OK**.

The repository is added to your list.

7.4.2.2 Editing a Repository

To edit a repository entry:

1. In the Font Environment Map on the Repository Manager screen, select the repository entry you want to edit, and then click **Edit Selected** to open the **Edit Repository** dialog.
2. Edit the folder location in the **Repository Location** field. You may click on the ellipsis button to aid in this process.
3. Click **OK after** making your selection.

The new repository location is saved.

7.4.2.3 Removing a Repository

To remove a repository entry:

1. Select the repository entry you want to delete in the **Font Environment Map** on the **Repository Manager** screen.
2. Click **Delete Selected**.
3. Confirm the removal of the repository. Click **Yes** to delete the repository, or **No** to return to the Repository Manager screen.

7.4.3 Font Convert Tool

The **Font Convert Tool** enables you to convert TrueType fonts into AFP raster fonts, which can then be used in the generated output (such as for PDF2AFP projects). This is applicable for any transformations that use the AFP Generator, and is especially useful when the user does not have the desired AFP fonts, but does have some TrueType fonts that are a close match. Since most AFP printers do not support TrueType fonts natively, the Font Convert Tool allows you to build the AFP fonts you require.



Note: Some TrueType fonts contain licensing information that may prevent them from being converted to AFP. Attempting to convert these fonts will result in an error message.

To use the Font Convert Tool:

1. In the Font Environment Map, click the **Font Convert Tool** button to open the **Font Convert Tool** dialog.
2. Complete the following fields:
 - **TrueType font.** Indicates the location path and name of the TrueType font that you want to convert to AFP. You can either enter the values manually, or click **Browse** to locate the font.
 - **Desired point size.** Specifies the desired point size of the converted AFP font. The default is 10.0.
 - **AFP name tag.** Indicates the name for the converted AFP font. You can enter up to 6 alphanumeric characters for single byte fonts, and up to 3 alphanumeric characters for double byte fonts.
 - **Single byte or Double byte.** Specifies the byte value for the font. Double byte is appropriate for fonts that contain more than 256 characters in their output fonts, such as for many Asian languages.
 - **Output directory.** Indicates the directory where the converted AFP font will be located.
3. Click **Convert** to begin the font conversion process.

The converted AFP raster font generates in the output directory you specified, and is comprised of 3 files, each with one of the following prefixes:

- **C0.** The Character Set, which holds the glyph character information and bitmaps.
- **T1.** The Codepage, which provides encoding information.
- **X0.** The Coded Font, which contains and links the names of the Character Set and Codepage files.

Now, using the Import Wizard from the Font Environment Wizard's Catalog Manager, you can import the converted AFP fonts for use in your project. For more

information, see “Using the Import Fonts Wizard” on page 567 and “Catalog Manager” on page 566.

Chapter 8

Project Review

8.1 Reviewing Your Project

Once your project is loaded into Output Transformation Designer, you should review your parameters, resources and components to ensure the information is complete and accurate before running your project.

The best way to do this is to work your way down the Transformation Projects tab, checking the parameters, values, and files for each parser, generator and other resources.



Note: For more information, see “Transformation Projects Tab” on page 83.

8.1.1 Reviewing Font Resources



Note: For more information, see “Project Properties Tab” on page 87.

When a project is loaded into Output Transformation Designer, the Output Transformation Engine tab shows all of the parsers, generators, and components associated with that project. Double-click the MFCT path shown in your project tree and the MFCT tab will open in the **Development window**.

Typeface	Style	Font Ratio	Output Color	Output Font	Tiff
TIMES NEW ROMAN LATIN1		1.0		Default-Proportional	*
TIMES NEW ROMAN LATIN1	BI	1.0		Default-Proportional-BoldItalic	*
TIMES NEW ROMAN LATIN1	I	1.0		Default-Proportional-Italic	*
TIMES NEW ROMAN LATIN1	B	1.0		Default-Proportional-Bold	*
HELvetica LATIN1	B	1.0		Helvetica-Bold	*
HELvetica LATIN1	BI	1.0		Helvetica-BoldItalic	*
HELvetica LATIN1		1.0		Helvetica	*
HELvetica LATIN1	I	1.0		Helvetica-Italic	*
COURIER LATIN1		1.0		Default-Fixed	*
COURIER LATIN1	BI	1.0		Default-Fixed-BoldItalic	*
COURIER LATIN1	I	1.0		Default-Fixed-Italic	*
COURIER LATIN1	B	1.0		Default-Fixed-Bold	*

Figure 8-1: Master Font Correlation Table

For more information about the MFCT tab, see [“MFCT” on page 86](#).

8.2 The Input File

This section describes the Input file.

8.2.1 What is the input file?

The input file is the document you want to transform. It is in the printstream format for the particular printer for which it was created. For example, if the file is to be printed on an IBM printer, the input file is in AFP format.

8.2.1.1 Acceptable AFP Input

AFP documents processed by the AFP Parser are the same documents you would normally send to an IBM Print Services Facility (PSF) for transformation into IPSF that prints on your printer. However, the AFP Parser does not support the output of PSF (IPDS).

These AFP documents may be pure linedata, a mixture of linedata and AFP structured fields, or fully composed AFP. MO:DCA-P files created by ACIF are also supported.

For more information, see [“AFP Parser” on page 132](#).

8.2.2 AFP Resources

The following AFP resources are required by the AFP Parser. These are the same resources used by PSF and the printer during printing.

For more information, see [“AFP Parser” on page 132](#).

8.2.2.1 Page Definition Resources

The **PageDef** and **fdPageDefs** parameters tell the AFP Parser the name and the location of the page definition resources, respectively.

For linedata or mixed-mode AFP files, a page definition is needed. Page definition resources define how the linedata will be placed on the page using one or more Data Maps.

The **CC=True** parameter is almost always needed when a page definition resource is used. It tells the AFP Parser that the first byte in linedata records is a Carriage Control byte, which defines spacing between lines and page ejects.

The **TRC=True** parameter may also be needed. It tells the AFP Parser that the second byte in linedata records is a Table Reference Character (Font Index Byte), which defines the font to be used for the data.

Page definition resource names will typically start with **P1**.

8.2.2.2 Form Definition Resources

A form definition resource is almost always required. With the use of Medium Map definitions, form definition resources define the following:

- Size of the page
- Initial fonts used
- Overlays that are automatically included on the page.

The **FormDef** and **fdFormDefs** parameters tell the AFP Parser the name and the location of the form definition resources, respectively.

If the page is a standard 8.5“x11” and the AFP input file contains all font definitions and resource calls, the form definition is not required.

Form definition resource names will typically start with **F1**.

8.2.2.3 Font Resources

AFP font resources include:

- Coded Fonts
- Character Sets (charset)
- Code Pages (codepage)

A font reference within AFP is accomplished through a single CodedFont reference, or by referencing the CharSet and CodePage directly. When a CodedFont is referenced, that resource will determine the CharSet and the CodePage.

CodedFont resource names will typically start with **X1**.

CharSet resources are required for proper processing. They contain the raster patterns and spacing information for each character, as well as the typeface name, point size, and bold/italic attributes needed for proper font mapping.

CharSet resource names will typically start with **C0**.

CodePage resources are required for proper processing. They determine which characters within the CharSet resource are being used.

Although processing can occur without CharSet and CodePage resources, the resulting output is rarely acceptable.

The **fdAFPfonts** parameter determines the location of the fonts. The fonts used by the AFP document are defined in the form definitions, other AFP resources, or the AFP document.

CodePage resource names will typically start with **T1**.

8.2.2.4 Overlay Resources

Any overlay resources referenced by the AFP input file or the form definitions are required. Overlay resources are separately composed document elements that act as preprinted forms for the page data. They are commonly referenced as electronic forms, form overlays, or forms. Overlay Generation Language (OGL) is an example of an application that builds overlays.

If overlays are not present during parsing, their data will not be placed on the page, but processing will continue.

The **fdOverlays** parameter determines the location of the overlays.

Overlay resource names will typically start with **O1**.

8.2.2.5 Page Segment Resources

Any page segment resources referenced by the AFP input file or the overlay are required.

Page segments are separately composed document elements that typically contain images, although they can contain text and line drawing commands.

If page segments are not present during parsing, their data will not be placed on the page, but processing will continue.

The **fdPageSegs** parameter determines the location of the page segments.

Page segment resource names will typically start with **S1**.

8.2.2.6 ACIF Resource Group

ACIF resources are sometimes contained in resource groups that are located at the beginning of the AFP file (inline resource group) or in a separate resource file that is referenced by the **fdACIF** parameter.

If no resource group is used, the resources must reside in a separate file (for MVS, these are members of a PDS).

8.3 AFP Font Resources

There are three types of font resource files:

- Character Set (charset) resource names will typically start with "C0".
- Codepage resource names will typically start with "T1".
- Coded Font resource names will typically start with "X1".

A font reference within AFP is accomplished through a single coded font reference, or by referencing the charset and codepage directly. When a coded font is referenced, that resource will determine the CharSet and the CodePage.

The **fdAFPfonts** parameter determines the location of the fonts. The fonts used by the AFP document are defined in the form definitions, other AFP resources, or the AFP document.

Related Links

- ["Character Sets" on page 580](#)
- ["Codepage" on page 582](#)
- ["Coded Font" on page 583](#)

How are font resources used?

A charset is paired with a codepage to define a set of glyphs (the charset) that occurs at incoming code points (as defined by the codepage). This defines a font that can be used by an AFP printstream.

Double-Byte Fonts

A series of charset/codepage pairs can be defined to build a double-byte font. When this occurs, each charset/codepage pair is tied to a section number. The section number is the first byte of the code point.

8.3.1 Character Sets

Character Set (charset) resources are required for proper processing. Although processing can occur without charset and codepage resources, the resulting output is rarely acceptable.

Charsets contain information about the font including:

- Typeface name.
- Point size.
- Style attributes, such as bold and italic.
- Glyphs. A glyph contains the information needed to represent a character on the page: its shape and size. Each glyph has an 8-byte glyph name, which is typically uppercase.

Charsets can be:

- **Raster fonts**, where the image for each glyph is stored in a bitmap in 300 or 240 dots per inch.
- **Outline fonts**, where the glyphs are drawn at various point sizes (Postscript Type1 fonts).

Charset resource names will typically start with C0.

Character Codes

Characters are represented by character codes. Character codes are generated and stored when a user inputs a document.

Single-Byte Character Sets

Single-Byte character sets (SBCS) provide 256 character codes. This is an adequate number to encode most of the characters needed for Western Europe.

Double-Byte Character Sets

SBCS, 256 character codes, are not enough to represent all the characters needed by multilingual users in a single font, or by users in Asia, where over 12,000 characters may need to be addressed at any one time. Multi-Byte character sets (commonly known as Double-Byte character sets) are necessary. Double-Byte

character sets (DBCS) are a mixture of Single-Byte and Double-Byte character encodings and provide over 65,000 character codes (2 to the 16th power).

Unicode

Unicode is a 16-bit encoding that encompasses many characters used in general text interchange throughout the world. Each Unicode index refers unambiguously to a given character. Unicode allows a larger range of characters to be addressed than is possible using a Single-Byte character encoding. All Unicode values are Double-Byte, which simplifies the way a Unicode-based system reads a string of text. In comparison, a Double-Byte system must determine which values in a string are Single-Byte character codes and which are Double-Byte character codes.

8.3.2 Output Transformation Engine Character Sets

The following tables describe Output Transformation Engine character sets.

8.3.2.1 Character Set Attributes

The Output Transformation Engine software uses the following character set attributes:

Attribute	Description
Charset.name	The name of the charset resource file, up to 8 characters, typically starts with a C0 (or CZ for outline fonts).
Charset.typefaceName	The name of the typeface, which refers to style in which the font glyphs appear. This is also known as the family.
Charset.isBold	A Boolean (true/false) flag indicating if the font is bold.
Charset.isItalic	A Boolean flag that indicates if the font is italic.
Charset.isRaster	A Boolean flag indicating if the font is a raster font or an outline font. Raster fonts contain glyph bitmaps that are at 240 or 300 dpi. Outline fonts (Postscript Type1 fonts) contain drawing instructions for each glyph and can be scaled to any point size.
Charset.pointSize	The point size in which the raster font is displayed.
Charset.dpi	The dots per inch resolution at which a raster font is displayed.
Charset.glyphCount	The number of glyphs contained in the charset.

8.3.2.2 Glyph Attributes

Each glyph inside the charset has the following attributes:

Attribute	Description
Glyph.name	The unique name of the glyph, 8 characters, usually uppercase.
Glyph.spacingWidth	The amount of space that the character and its surrounding space takes up.
Glyph.bitmapRows	The height of the actual character.
Glyph.bitmapCols	The width of the actual character.
Glyph.bitmapRaster	The actual bits that make up the character.

For example, character set C0A075Z0 has the following attribute values:

<code>Charset.name = C0A075Z0</code>
<code>Charset.typefaceName = SONORAN SANS SERIF</code>
<code>Charset.isBold = true</code>
<code>Charset.isItalic = false</code>
<code>Charset.isRaster = true</code>
<code>Charset.pointSize = 36</code>
<code>Charset.dpi = 300</code>
<code>Charset.glyphCount = 300</code>
<code>Glyph.name = LA010000 draws the lowercase letter "a".</code>
<code>Glyph.name = LA020000 draws the uppercase letter "A".</code>
<code>Glyph.spacingWidth = ?????</code>
<code>Glyph.bitmapRows = ?????</code>
<code>Glyph.bitmapCols = ?????</code>
<code>Glyph.bitmapRaster = ?????</code>

8.3.3 Codepage

A codepage is used by the system to encode and interpret string characters. When a new codepage is loaded, different character codes are provided to the application. It contains the encoding used for the font. It maps a hex point to a glyph name within the charset.

8.3.3.1 Codepage Formats

Codepage formats are not the same for each language. Each codepage is represented by a unique integer. A codepage is a list of selected character codes in a certain order. Codepages are usually defined to support specific languages or groups of languages which share common writing systems. Some languages, such as Japanese and Hindi, have multi-byte characters, while others, such as English and German, need only one byte to represent each character.

A codepage can be a:

- **Double-byte codepage**, where each incoming code point is represented by 2 bytes.
- **Single-byte codepage**, where only 1 byte represents the incoming code point.

The order of character codes in a codepage enables the system to provide the appropriate character code to an application when a user presses a key on the keyboard. Codepages are typically organized in an EBCDIC encoding, where the letter A is at code point C1. ASCII encodings can occur, where the letter A is at code point 41; however, codepages may be encoded in any non-standard encoding.

For example, `codepage.Name = T1D0BASE` includes the following mappings:

- Hex C1 maps to the Glyph named LA020000 (The letter A)
- Hex F1 maps to the Glyph named ND010000 (The digit 1)

8.3.3.2 Codepage Resources

Codepage resources are required for proper processing. They determine which characters within the CharSet resource are being used. Although processing can occur without CharSet and Codepage resources, the resulting output is rarely acceptable.

Codepage resource names will typically start with "T1".

8.3.4 Coded Font

Coded fonts contain a reference to one or more character sets or codepage pairs.

A font reference within AFP is accomplished through a single coded font reference, or by referencing the charset and codepage directly. When a coded font is referenced, that resource will determine the charset and the codepage.

CodedFont resource names will typically start with "X1".

8.4 AFP File Transfer Issues

AFP is a record-based architecture usually created on IBM mainframe computers, such as MVS, which has a record-based file system. When processing these files on a PC or UNIX system, take care when transferring the AFP files and AFP resources since PC and UNIX file systems, unlike MVS, have no logical understanding of records.



Note: Never translate AFP documents or resources from EBCDIC to ASCII or from ASCII to EBCDIC, because the translation may not be reliable.

8.4.1 AFP Resources and Fully Composed AFP Documents

These files can be transferred without any data changes. Because AFP has the length field definition within its records, the files transferred by binary will still have a logical means in which the software can determine records. However, sometimes records on the MVS host will have an inaccuracy between the true record length and the record length within the AFP structured field. In these cases, the AFP Parser requires some additional information to determine record breaks. Adding CRLFs to resources and input files will sometimes correct the situation, but not always.

8.4.2 For AFP Line Data and Mixed Mode Files

No record length information exists on the records. The easiest and most common method for transferring these files is to add CRLFs at the end of the records during transfer. But the problem indicated in the above paragraph remains.

If RDW, or some other type of length field, can be added to the beginning of each record during transfer, this method is preferred.

See the I/O Utility information on file transfer options.

8.5 File Extension Recognition

The following tables list the file extensions recognized by Output Transformation Designer.

8.5.1 Output Transformation Engine File Extensions

Extension	File Type
.d2eproj	Project file
.d2emsg	Message file
.d2emfct	Font correlation tables and color definition tables
.d2eesys	System configuration file
.jar	Java archive file

.tif	TIFF image file
.bmp	Bitmap image file
.bat	Batch files and UNIX script files

8.5.2 AFP File Extensions

Extension	File Type
.ndx	AFP Page Index file
.fil	AFP ACIF resource file
.fdf	AFP Form Definition
.ovr	AFP Overlay
.pde	AFP Page Definition
.psg	AFP Page Segment Resource
.fnt	Font files for the generator and the parser
.afp	AFP file
.240	AFP font file
.300	AFP font file

8.5.3 PDF File Extensions

Extension	File Type
.afm	Adobe Font Matrix file
.pfb	Adobe Bitmap font
.pdf	Adobe Portable Document Format file

Chapter 9

Running a Project

The following topics are available on running a project.

9.1 Processing Overview

Output Transformation Designer executes according to the parameters, components, and font correlation data you have specified for your particular project.

9.1.1 Running the Project

Provided you have configured at least one parser and one generator, you can begin the process to run your project by performing one of the following actions:

- Click the **Execute** button, , on the toolbar.
- From the **File Systems** tab, right-click the project and select **Execute**.
- In the **Output Transformation Engine** tab, right-click the root project file, and select **Run**.
- Select **Build > Execute** from the menu.

When a project is executed, it is automatically saved if there are any unsaved changes to the file and the **Execution Options** dialog box opens, where you can provide some further job variable and source settings for your parser and generator. (For more information on the options available to be set, see [“Job Execution Options Window” on page 588](#).) After your options are configured, click **Execute** to run the project.



Note: You can use the **Undo** function to revert changes to the saved project.

9.1.2 Locating Your Output

The output will be stored with the project, in the project directory or in the output file and location you specified, for subsequent retrieval.

For more information on viewing your output, please see the section [“Reviewing the Output” on page 615](#).

9.2 Job Execution Options Window

Your configured project can be executed from the Output Transformation Engine tab by right-clicking on the project and selecting **Run** from the context menu. The Execution Options window has a few separate tabs which are detailed in this section.

After configuring your execution options, click the **Execute** button and your project will be run. Information relating to the job appears in the Output tab, which is located below the Development window.

Related Links

- [“Reviewing the Output tab” on page 615](#)

9.2.1 Job Options Tab

With the options on this screen, you can set additional execution options for your component.

- **Rasterize All Fonts.** Specifies whether or not all of the input fonts used should be rasterized into images for the output during processing.
- **Start Page / Stop Page.** Specifies the page to begin processing from, and the last page where processing will stop. An entry of 0 as the Stop Page denotes no end and thus, the document’s last page will be the final page processed. The default setting is to transform the entire document.

If you are testing a project, you may prefer to run a short range of pages until you are certain your project settings are correct.



Note: The page range values can also be changed in the parser Parameters, or its associated wizard. The most recently changed setting will always be used. For example, if you change the Stop Page on the Job Options tab of the Execute Options form to run a test, it will remain at this setting until it is overridden by either changing it on the same form, or by changing it in the Parameters, despite what may be displayed in the Parameters. Similarly, if you change these values in the Parameters, the new values will override what’s entered in the Job Options tab of the Execute Options form until a change is made to either the Job Options tab or the Parameters.

9.2.2 Job Variables Tab

Job variables can be specified on this tab to provide dynamic substitutions within projects. They can be used to redirect output for a specific run, or to set variables used in extension components. For more detailed information about job variables and their use, please see *OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-H-PTE)*.

9.2.2.1 Adding New Job Variables

To add a new job variable to your project:

1. In the **Executions Options** screen, on the Job Variables tab, click the  button to add a new line in the table below.
2. Double-click in the **Name**, **Description**, or **Value** cells to enter information.
3. Press **Enter** to save the cell information.

The new job variable is used, if applicable, during your next project run. Otherwise, it is saved for subsequent executions of your project.

9.2.2.2 Removing a Job Variable

To remove a job variable from your project:

1. In the Job Variables table, click the job variable that you want removed to highlight the row.
2. Click  to delete the selected job variable.

9.2.3 Sources Tab

Each entry on the Sources tab represents a separate input that is used to configure the selected components for your particular project. You can view the sources in a table by clicking the **Table View** button, , or in a structured tree view, by clicking the **Tree View** button, . In either view, the following headings are shown:

- **Name.** Indicates the name of the input source/parameter. In Table View, the names appear as the full package name, while in Tree View, the names are broken down into their file path structure.
- **Description.** Provides a brief explanation of the input source's function.
- **Value.** Contains the file path that is the source for the particular parameter to read.



Note: If you change the fdInput Value, it will override the path shown in the Properties window. To re-enable the path listed in the Properties window, click the ellipsis, re-select the path, and save the project before running it. The most recent setting will always take precedence.

9.3 Input Monitors

A Output Transformation Engine **Input Monitor** watches designated directories for input files. When an input file is written to a watch directory, Output Transformation Engine runs the specified project to process the input file.

For detailed information on using input monitors, see the following topics:

- “Input Monitor Setup” on page 599
- “System Configuration Parameter File” on page 599
- “Watch Directory Configuration File” on page 600
- “Input Monitors” on page 598

9.4 Running in Production

The topics covered in this section discuss issues that apply to running in a production environment.

9.4.1 Transferring Configuration Files

The topics in this section discuss transferring various types of configuration files.

9.4.1.1 Files from Development

Output Transformation Designer builds two files:

- The MFCT (Master Font Correlation Table) file, with the file extension `.d2emfct`.
- The Project file, with the file extension `.d2epro.j`.

In Output Transformation Engine, the system configuration file (`.d2esys`), project files (`.d2epro.j`) and font table files (`.mfct`) are XML based files. They can be transferred to the other platforms (such as a test server or production machine) through normal file transfer processes such as FTP.

Many times these hosts are UNIX or Mainframe platforms, so it is important to be aware of specific requirements of these platforms in the context of Output Transformation Engine project files. You need to become familiar with viewing and modifying these configuration files in an XML or plain text editor. These modifications are generally straightforward and tend to be related only to file system related variables (path names and file names). Often, these only apply to values of `<filedef>` tags in the XML.

9.4.1.2 UNIX Servers

If the host environment is on a UNIX-based platform, then keep in mind the following points:

- The file paths in the system configuration (.d2esys) and project files (.d2eproj) need to be modified to use the forward slash, /, instead of the backslash, \.
- UNIX is case sensitive with all filenames, therefore, all file paths and file extensions, such as: .fnt, .psg, .jsl, **must** reflect the exact case of the files on disk they apply to.

9.4.1.3 IBM Mainframe (z/OS) Servers

On Windows platforms, where all configuration files are typically created using Output Transformation Designer, XML data is written using that platform's native ASCII/UTF-8 encoding. If the files need to be transferred to an IBM z/OS mainframe, where the native encoding is EBCDIC rather than ASCII, the correct file transfer procedures must be followed to ensure compatibility.

When you view an XML file in a text editor on a Windows platform, the first line of the file looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>
```

This example specifies that the encoding of the file is the UTF-8 (ASCII) format.

9.4.1.3.1 Properly Transferring XML Files from Windows to z/OS

Using FTP, there are two methods to properly transfer the UTF-8/ASCII formatted XML files from Windows to the z/OS mainframe.

- Leave the encoding as UTF-8 and FTP the files in BINARY mode from the Windows platform to the mainframe. This will keep the native ASCII bytes as is and Output Transformation Engine's XML parser will process them as ASCII/UTF-8 data. However, this method is sometimes not viable because it means that you will not be able to view or modify the file as text on the mainframe. In other words, this method is only useful if you know for certain that no modifications will need to be made to files from the mainframe console.
- Manually change the encoding="UTF-8" statement to be encoding="ebcdic-cp-us" and FTP the file in ASCII/TEXT mode. In ASCII/TEXT mode, the FTP process automatically converts ASCII bytes into EBCDIC bytes. The encoding specified (ebcdic - cp - us) will be correct and you will be able to view and modify the XML files as normal text files from the mainframe console.

9.5 Command Line Operation

Output Transformation Designer includes many ways to define your parameters within their GUIs. However, you can still run jobs and batch file operations from the command line. For example, you may want to start and stop input listeners as you run that job.

9.5.1 Usage

```
java -classpath com.xenos.d2e.d2eMain %D2E_CLASSPATH%  
  
-config=<system config file>  
  
-app=<application name>  
  
-input={IO Directive}<input file>  
  
-output=<output file>  
  
-fontmode=<FontModeEnum>  
  
-debug=<debug config file>  
  
-jobvariable=<variable name>=<variable value>
```

Where the variable:

- <system config file> defines the **system configuration XML** file.
- <application name> defines the **application name** (Run single job mode). There can be multiple -app= points to a **.d2epro** file.
- <input file> defines the **input file** (Run single job mode). It overrides the FdInput parameter of the parser.



Note: The {IO Directive} value specifying the input file format is highly recommended. If this directive is not provided, the parser will use the directive that is specified by the FdInput directive in the project file.

- <output file> identifies the **output file** (Run single job mode). It overrides the FdOutput parameter of the generator.
- <FontModeEnum> specifies **MFCT, RASTERIZE** (Run single job mode).
- <debug config file> defines the **debug configuration file**.
- <variable name> identifies the **name of a variable** to set.
- <variable value> identifies the **value of the variable** to set.

9.5.1.1 Specifying the ClassPath File

The **ClassPath** (or **cp**) argument sets the path to the Java class files used by Output Transformation Engine.

Argument	Description
-classpath (or -cp)	It usually points to a variable like %D2E_CLASSPATH% or %D2ETEMPCLASSPATH%. The ClassPath file contains pointers to all the Output Transformation Engine JAR files in the engine\version\directory and the d2evision.jar in the bin subdirectory. It may use variable %D2E_ENGINEPATH% or %D2ETEMPENGINEPATH% to help simplify upgrading from one engine to another without having to change many lines of code. For example, <pre>java -cp "C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\engine\version\bin\d2evision.jar"; "C:\OpenText\OTS\<version>\lib\ common\Xenos-framework-util. jar" com.xenos.d2e.d2eMain</pre>  Note: When copying the example into the command line, remember to change the version name (<version>) and ensure the command is one line and not word wrapped.

9.5.1.2 Specifying the Configuration File

The Java class that is used to run Output Transformation Engine from the command line is called d2eMain and is found in the com.xenos.d2e package. To start the engine from the command line you first need to specify your system configuration file, which is usually located in the following directory:

```
-config <install location>\engine\version\config
```

Argument	Description
-config	This is a required argument. If you specify only the <code>-config</code> argument, Output Transformation Engine will start up and wait for jobs to be submitted to it through an input monitor, such as the DirectoryListener. <pre>-config="C:\OpenText\OTS \<version>\dev-studio\build\vision\ staging\engine\version\config\ ivtest.d2esys"</pre> For watchdir examples see: <pre><install_home>\engine\version\ sampletests</pre>  Note: When copying the example into the command line, remember to change the version name (<version>) and ensure the command is one line and not word wrapped.

9.5.1.3 Specifying the Project File

If you wish to run a job from the command line, you must also specify the *application* you wish to run using the **-app** argument.

Argument	Description
-app	This is a required argument. It specifies pointers to the input file, resources, and job parameters. This is done by specifying a path and project file name as follows: <pre>-app=C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\workspaces\default\sample\sample-afp2pdf.d2eproj</pre> You may specify more than one -app argument to run multiple jobs in sequence. For example: <pre>-app=C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\workspaces\default\sample\sample-afp2pdf.d2eproj</pre> <pre>-app=C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\workspaces\default\sample\sample-afp2tiff.d2eproj</pre>  Note: When copying the example into the command line, remember to change the version name (<version>) and ensure the command is one line and not word wrapped.

9.5.1.4 Optional Command Line Arguments

You can also customize some of the parameters for the job by using the following optional arguments:

Argument	Description
-input	This argument sets the full file name and path of the input file to be processed. It overrides the value of the FdInput parameter in the .d2eproj file.  Note: The I/O directive specifying the input file format at the beginning of the input file path is highly recommended. If this directive is not provided, the parser will use the directive that is specified by the FdInput directive in the project file. The example below contains {afp} to specify that the input file is in AFP format: <pre>-input="{afp}C:\OpenText\OTS \<version>\dev-studio\build\vision\ staging\workspaces\default\sample\ input\sample.afp"</pre>
-output	This argument sets the full file name and path of the output file to be created. It overrides the value of the FdOutput parameter in the .d2eproj file. This argument does not require an I/O directive at the beginning of the output file path in order to specify the output type, unless you want the data formatted in a special way, such as when the data is to be loaded into mainframe (z/OS) datasets. For example: <pre>-output="C:\OpenText\OTS\<version> \dev-studio\build\vision\staging\ output\sample.pdf"</pre>

Argument	Description
-fontmode	This argument specifies how you would like to perform your font correlation for this transform. This argument has three settings: • MFCT. This is the default, which uses the MFCT table (* .d2emfct) specified in your project file to correlate the fonts. • MAP. The same as MFCT. • RASTERIZE. This option does not perform any correlation, but instead rasterizes all of the input fonts used into images in the output. If you select this option, then: – Text in the input file will not be searchable. – Output file size will be larger. – You will need bitmap input fonts.
-jobvariable	This argument sets any job variables, such as {d2eOutputPath}. It can be used to redirect output for a specific run or to set variables used in extension components. For example: <pre>- jobvariable=myJobVariable=myValue</pre> <pre>- jobvariable= d2eoutputPath="C:\testout\"</pre> With this example, myValue will be substituted in any parameter that uses {myJobVariable}. For more information about job variables, see <i>OpenText Embedded Output Transformation Engine - User Guide (VDTOTS-H-UTE)</i> and <i>OpenText Embedded Output Transformation Engine - Developer's Guide (VDTOTS-H-PTE)</i>.

Argument	Description
-debug	This argument gives the full path and file name of the debug configuration file (debug.config). The -debug argument is used for internal diagnostics. If you are experiencing trouble with your installation then your Professional Services representative or consultant (see Customer Support) may provide you with a debug configuration file that must be specified using this argument. For example: <pre>- debug="C:\OpenText\OTS<version>\dev-studio\build\vision\staging\engine\version\config\debug.config"</pre> The information gathered using the debug argument will be used to determine the problem that you are experiencing.

9.5.1.5 Command Line Example

Below is an example command line.

```
java -cp "C:\OpenText\OTS<version>\dev-studio\build\vision\staging\engine\version\bin\
d2evision.jar";"C:\OpenText\OTS<version>\lib\common\Xenos-framework-util.jar"
com.xenos.framework.tools.d2e -config="C:\OpenText\OTS<version>\dev-studio\build\vision
\staging\engine\version\config\ivtest.d2esys"
-app="C:\OpenText\OTS<version>\dev-studio\build\vision\staging\workspaces\default\sample
\sample-afp2pdf.d2eproj"
-input="{afp}C:\OpenText\OTS<version>\dev-studio\build\vision\staging\workspaces\default
\sample\input\sample.afp"
-output="C:\OpenText\OTS<version>\dev-studio\build\vision\staging\output\sample-
afp2pdf.pdf"
```



Note: When copying the example into the command line, remember to change the version name (<version>) and ensure the command is one line and not word wrapped.

9.5.2 Input Monitors

An Input Monitor watches designated directories for input files. When an input file is written to a watch directory, the application runs the specified project to process the input file.

9.5.2.1 Input Monitor Setup

To set up input monitors, you must do the following:

1. Identify the type of input monitor you want to have.
2. Edit the system configuration parameter file (.d2esys) used to start up the Output Transformation Engine system. The system configuration parameter file defines the input monitor as well as all input sources. For more information, see [“AFP Resources and Fully Composed AFP Documents” on page 584](#)
3. Create the Watch Directory configuration file (watchdir.config). The XML-based Watch Directory configuration file defines which directories the input monitors watch for incoming input files. It also specifies which projects to run when an input file is detected. You can find this file in the file directory where your engines are stored. For more information on the Watch Directory file, see [“AFP Generator Parameters” on page 180](#).

An example of the file directory is:

```
C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\engine\
version\config
```



Note: When copying the example into the command line, remember to change the version name (<version>) and ensure the command is one line and not word wrapped.

9.5.2.2 System Configuration Parameter File

Each instance of the application is started with its own (xml-based) **system configuration parameter file** (watchdir.d2esys). To set up the system with input monitors, an **input monitor** element, <inputmonitor>, should be included in the system file.

Input Monitor

The **InputMonitor** element, <inputmonitor>, includes settings to control different types of input monitors. It has a child element **InputSource**, <inputsource>.

Sleep Interval

The **SleepInterval** attribute of the <inputmonitor> element represents the number of milliseconds the input monitor will wait before it cycles through the input sources to collect the external command requests. It is set at 1000 milliseconds, or 1 second, in the example above.

You can also define the SleepInterval element independently of InputSource in your Properties window.

Input Source

The <InputSource> element, InputSource[0], is a child of <inputmonitor> and is the configuration string for an input monitor element. It is a comma delimited list where

the first token will indicate the input monitor type to use, and the remaining tokens are the configuration parameters for that monitor.

Multiple `<inputsource>` elements can be defined in the configuration file.

In the example above, the first `<inputsource>` element defines a `DirectoryListener`. This indicates which directory to watch for command files such as shut down commands or flush commands. The second `<inputsource>` element defines an `InputFileListener`, `watchdir.config`, for the watch directory configuration file. For more information on the watch directory configuration file, see [“Watch Directory Configuration File” on page 600](#).

For more information regarding the system configuration file, see *OpenText Embedded Output Transformation Engine - User Guide (VDTOTS-H-UTE)*.

Example System Configuration Parameter File

```
<inputmonitor sleepinterval="1000">
<inputsource>DirectoryListener,{d2eSysPath}../sampletests/watchdir/cmd,500,.d2ecmd</
inputsource>
<inputsource>InputFileListener,{d2eSysPath}watchdir.config</inputsource>
</inputmonitor>
```

9.5.2.3 Watch Directory Configuration File

This is an XML-based configuration file that defines which directories to watch for incoming input files. It also specifies which projects to run when an input file is detected.

Input File Listener

The root element of the Watch Directory configuration file is **InputFileListener**. It has the following attributes:

searchdelay	Represents the number of milliseconds the input file listener will wait before it cycles through the watch directories to pick up input files.
filechangeinterval	Specifies the interval in milliseconds to check if a file has changed its size in order to determine if the file is stable or not. If the size of the file is found to have changed during the specified time, it will not be processed until it stops changing.  Note: If the file transfer connection is slow, this value should be increased.
queuedext	The file name extension of the input files only while in the process waiting queue. The default value is <code>.que</code> , but can be renamed in the configuration file to your choosing.

finalfilenamepattern	Allows customization of the final file name of a processed input file. The values of the attribute can be any alphanumeric character or symbol, except for the few symbols that are restricted by operating system file name rules. The default value is {d2eJobId}, but you may use any number of valid “Adding a Color to a Metacode Color Table” on page 553. The {d2eJobId} parameter will automatically add a mandatory time stamp (in milliseconds) to the file name. If the parameter {d2eJobId} is not included inside finalfilenamepattern, a time stamp will still be added as a suffix to the file name. Since the time stamp is included in {d2eJobId}, it is suggested that the parameter {d2eJobId} should be placed as the last part of the finalfilenamepattern attribute. This is done in order to more easily match the processed input file with the corresponding report file. For example: <pre>finalfilenamepattern="CompanyName_{InputFileName}-{d2eJobId}"</pre> The processed input file will then be renamed to: <pre>CompanyName_InputFileName-ProjectName_ProjectRunNumber_TimeStamp</pre>
finishedext	The file name extension of the input files which are processed successfully. The default value is .fin, but can be renamed in the configuration file to your choosing.
erroredext	The file name extension of the input files which are not processed successfully. The default value is .err, but can be renamed in the configuration file to your choosing.
jobvarfiletype	Designates the job variable support file used during the job. For more information, see “Job Variable Support” on page 602.



Note: For more information on Input Listeners, see [“AFP Color Tables”](#) on page 551.

Application Directory

Below the **InputFileListener** element, one or more child **ApplicationDir** elements are defined. This is an empty element that has the following attributes:

- **name**. A user-defined name for the application.
- **dirwatch**. The directory to watch for incoming input files to be processed by this application.
- **fileext**. The file extensions to watch for on the input files. Only one file extension can be defined per application. It is also case insensitive. For example, the `.afp` file extension defined above would pick up the file `input.afp`.
- **application**. The file path to the project file.
- **inputiodirective**. The I/O directive to be used on the input file(s).

Sample Watch Directory Configuration File

```
<?xml version="1.0" encoding="UTF-8"?>
<inputfilelistener searchdelay="0"
  filechangeinterval="0"
  queuedext=".queued"
  finalfilenamepattern="{d2eInputFileName}"
  finishedext=".finished"
  erroredext=".error"
  jobvarfiletype="json"
  >
  <applicationdir name="pcl_to_pdf"
    dirwatch="{d2eSysPath}../otos/watchdir/input/pcl_to_pdf"
    fileext=".pcl2pdf"
    application="{d2eSysPath}../ote-project/pcl2pdf.d2eproject"
    inputiodirective="{nocrlf}"
  />
  <applicationdir name="pcl_to_ps"
    dirwatch="{d2eSysPath}../otos/watchdir/input/pcl_to_ps"
    fileext=".pcl2ps"
    application="{d2eSysPath}../ote-project/pcl2ps.d2eproject"
    inputiodirective="{nocrlf}"
  />
  <applicationdir name="pcl_embed_font"
    dirwatch="{d2eSysPath}../otos/watchdir/input/pcl_to_pcl"
    fileext=".pcl2pcl"
    application="{d2eSysPath}../ote-project/pcl_embed_font.d2eproject"
    inputiodirective="{nocrlf}"
  />
</inputfilelistener>
```

9.5.2.3.1 Job Variable Support

The **InputFileListener** element also supports the use of job variables in transform jobs through inclusion of a second file along with the input file. This extends the listed job variables as a means of customizing projects.

Enabling the Job Variable Support File

To use a job variable support file in your job, you must define the **InputFileListener** > **jobvarfiletype** property in the Watch Directory configuration file. It is an optional property, so the value can be left blank in the configuration file, which is the default. When a property definition is provided, the input file listener looks for the

corresponding job variable file. The support file must be in JSON format so to enable the feature, set the **jobvarfiletype** property to **JSON**.

When adding the JSON job variable support file to the directory, the support file name must match the file name of the InputFileListener input file and contain the proper file type extension. For example, if the input file is named:

```
input-123.pcl
```

Then the corresponding job variable support file must be named:

```
input-123.pcl.json
```

If the job variable support file cannot be found or the file contains syntax errors, then the job is not submitted and an error is thrown. You can consult the system log messages to determine the cause of the failure.



Note: After the job variable file is processed and the transform job is submitted, the job variable file is automatically deleted.

Configuring the JSON Job Variable Support File Structure

When creating the job variable support file, you must ensure it uses UTF-8 encoding. Furthermore, it must be structured with all values in quoted strings or unquoted null strings. All unquoted null values are ignored and not provided to the job as an available job variable.

The following code snippet demonstrates the proper file structure:

```
1  {
2      jobVariables: {
3          "jobVar1": "value1",
4          "jobVar2": "value2",
5          "jobVar3": null
6          "jobVar4": "value4"
7      }
8  }
```

Figure 9-1: Sample JSON job variable file

9.5.2.4 Starting an Input Listener

Ensure that the System Configuration parameter file and the Watch Directory configuration file have both been edited to define the input monitors, watch directories, and projects to run.

For more information, see [“AFP Resources and Fully Composed AFP Documents” on page 584](#), [“AFP Generator Parameters” on page 180](#), and [“Adding a Glyph Mapping” on page 559](#).

To start an input listener:

1. Start Output Transformation Engine using the `-config` command line option to pass the System Configuration parameter file (`*.d2esys`).
2. Type the following at the command prompt to run a Windows batch file or Unix script:

```
start_watchdir
```



Note: A sample Windows batch file and Unix script is provided in the installation directory:

```
C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\engine\version\sampletests
```

Some initialization messages will appear on the screen to notify you of the various listeners the application has added. Once initialization is complete, you can begin placing input files in the watch directories.

When files are detected, the application will issue messages about the status of the job.

9.5.2.5 Stopping an Input Listener

To stop the input listeners, you have to copy command files into the DirectoryListener defined in the system configuration file. There are sample command files in the `sampletests` directory,

```
C:\OpenText\OTS\<version>\dev-studio\build\vision\staging\engine\version\sampletests
```



Note: When copying the example into the command line, remember to change the version name (`<version>`) and ensure the command is one line and not word wrapped.

To stop an input listener:

1. Type one of the following sample command files at the command prompt:
 - `shutdown_request.d2ecmd`. This causes the system to shut down, provided it is not currently running a job.

- `shutdown_now.d2ecmd`. This causes an immediate shutdown of the system after any running jobs are cancelled.
 - `flush.d2ecmd`. This causes a shutdown after all jobs are complete.
2. Type the following batch script at the command prompt to stop a Windows batch file or Unix Script:

```
stop_watchdir
```

You will also find this sample command file in the `sampletests` directory.

9.5.3 Resource Collection IORES

The IORES utility provides an easy method to package all resources used by a parser into one single binary archive file. Once the resources are collected into a single file, a project can be run with an IORES file as the only resource.

This enables archived printstreams to be easily transferred as one file stream for on demand conversions, thus reducing complexity and network and/or system overhead.

The popular ZIP format is supported by the standard Java libraries, but it is costly in terms of parsing performance. The IORES Resource Collection format alleviates those concerns.

9.5.3.1 IORES Parameters

The IORES parameters provide an easy method to package all resources used by a parser into one single binary archive file.

To ensure that you have all of the settings you need, we recommend you review the IORES parameters in the **Properties window** before running your project.

There are two types of resource collection parameters:

InputCollection	Specifies the file definition for the <i>input</i> IORES file.
OutputCollection	Specifies the file definition for the <i>output</i> IORES.
Compression type	Indicates the type of compression to use; either Deflated or None .
Maximum buffer size	Specifies the maximum buffer size for an individual IORES entry output stream. If the output bytes for an entry exceed the maximum buffer size, a temporary file is used to store the bytes before they are written to the new IORES file. For more information about temporary files, see “tempPath Job Variable” on page 606 .

tempPath Job Variable

Temporary files are used to store the resource data while the IORES file is being built.

The **tempPath** job variable specifies the directory in which to create any temporary files that are needed. If **tempPath** is not specified, it is set to the Java system property `java.io.tmpdir`.

The temporary files are deleted on normal termination of the JVM.



Note: To see an explanation of a parameter's use and syntax, select the parameter in the **Properties window** and the message area at the bottom of the window will display the explanation. Place your mouse pointer on the line between the parameters list and the definitions area and drag up or down to alter the size of the **Message** window.

IORES Parameter Examples

```
<iores>
<object name="InputCollection">
  <filedeflist name="FdInput">
    <filedef directives="" location="{d2eAppPath}font-resources.res"/>
  </filedeflist>
</object>
<object name="InputCollection">
  <filedeflist name="FdInput">
    <filedef directives="" location="{d2eAppPath}overlay-resources.res"/>
  </filedeflist>
</object>
<object name="OutputCollection">
  <!-- Compression type, default is "none" -->
  <enum name="Compression" value="deflated"/>
  <!-- Buffer size in kilobytes -->
  <int name="MaxBufferSize" value="20"/>
  <filedeflist name="FdOutput">
    <filedef directives="" location="{d2eOutputPath}resources-output.res"/>
  </filedeflist>
</object>
</iores>
```

9.5.3.2 IORES I/O Directives

There are two I/O Directives related to resource collection:

{IORES-IN}

Specifies the **input** stream and internal path for the file definition, which is to be obtained from one of the input IORES files. The internal path can be entered in the **path** field of the **Select Resource Path/File** window. For more information about working with the Select Resource Path/File window, see [“Select Resource Path/File Window” on page 102](#).

{IORES-OUT:<path>}

Specifies that the file definition is to be written to the **output** IORES file. The <path> specifies the file path to use within the resource file.

If this directive is specified on an input resource, the input stream will be written to the resource file and then passed on to the requester, the AFP Parser for example.



Note: For file definitions that are defined to have an {IORES-IN} directive and an I/O Handler, the I/O Handler takes precedence and the {IORES-IN} parameter has no affect. Although this I/O directive can be used for FdOutput parameters, keep in mind that entries can only be written one at a time.

For more information on **I/O Directives**, see [“I/O Directives” on page 118](#).

9.5.3.3 Creating a Resource Collection

To collect the project resources into an IORES file by running a job:

1. Set up an IORES parameter, using output as the resource type.
2. Add the IORES-OUT directive to any input resources to be added to the IORES file. For more information, see [“Index record name” on page 292](#).
3. Run the job.

After running the job, the new IORES file can be found in the location specified in the IORES file definition.

9.5.3.4 IORES and Resource Paths

When setting up a new project using IORES files, you must be aware of the different file paths required.

To properly set the resource paths using an IORES file, follow these steps for each resource type contained in the IORES file:

1. Create a new project with a parser and generator of your choice.
2. From the **Output Transformation Engine** tab, right-click the parser.
3. Select **View Resources** from the shortcut menu. The **Resources** tab opens.
For more information about the Resources tabs, see “Resources Tab” on page 98.
4. Select a resource, such as fonts, page segments, or file definitions.
5. Click the **Add** button, , on the **Resources toolbar**.
The **Select Resource Path/File** window opens.
6. Complete the following fields:

- **Path.** Enter the internal resource path within the IORES file, not the path to the IORES file.

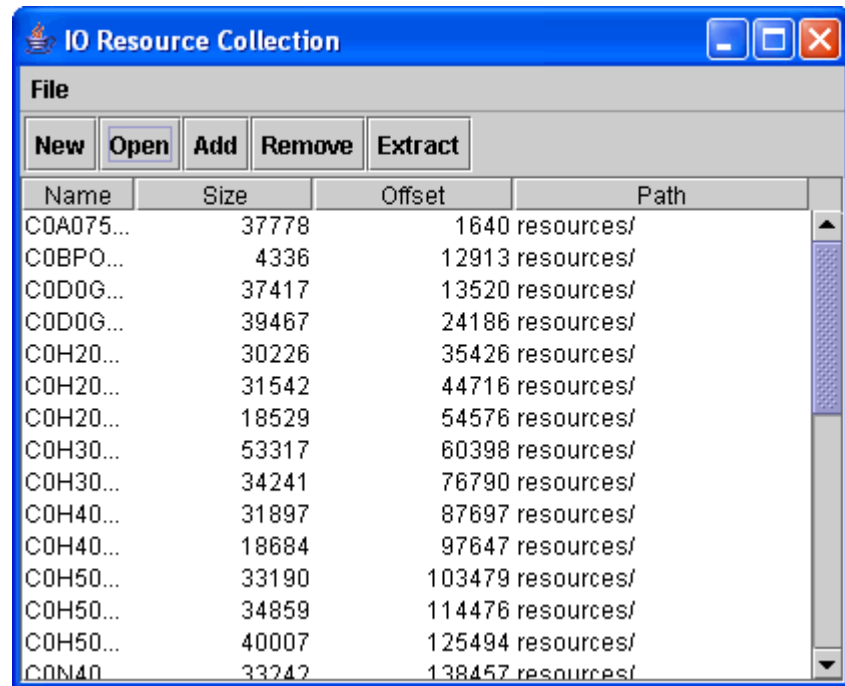
The internal resource path may no longer exist or be accessible, which is acceptable. It is a relative path within the IORES file.

Each resource can have a different internal path so each resource type must have its path set in this manner.

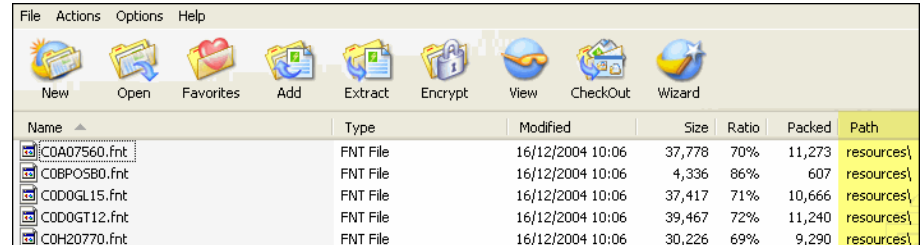


Note: To see what is the correct resource path to use, view the IORES file in the IORES Builder Utility or, using the Zip2IoRes Utility, convert the IORES to a zip file and view the resulting zip file. The path for each resource contained in the IORES file will be listed in the path column.

- *IORES file in the IORES Builder Utility*



- *IORES file as a ZIP file*



- **Files of Type.** Select the file type from the drop-down list.
 - **Directives.** Select the {IORES-IN} I/O directive.
7. When you are finished, click **OK** to accept any changes you have made, or **Cancel** to exit this dialog without making any changes to the resource path.

9.5.3.5 Running a Job with an IORES File

There are two methods for running a job when the project file is part of the resource collection. The IORES files passed in using these methods are used for input only:

- IORES-IN I/O Directive
- d2eMain



Note: For detailed information about running from the command line, see [“Command Line Operation” on page 592](#).

IORES-IN I/O Directive

1. Set up an input type IORES parameter.
2. Set the {IORES-IN} I/O directive for any input resources that are available in the IORES file. For more information, see [“Index record name” on page 292](#) and [“Adding a Form Field” on page 409](#).
3. Run the job.

d2eMain

Run d2eMain with the `-iores` argument.



Example 9-1: d2eMain example

```
java com.xenos.d2e.d2eMain -app=sample-afp2pdf.d2eproject -iores=sample.iorest
```



9.5.3.6 IORES Utilities

This section includes topics related to IORES utilities.

9.5.3.6.1 IORES Builder Utility

The IORES Builder Utility, **IoResBuilder**, is a graphical interface that provides the following functionality:

- Create a new IORES file.
- Open and view an existing IORES file.
- Modify an existing archive (add/remove entries).
- Extract individual entries from an IORES file.

Starting IoResBuilder

To run IoResBuilder, use the following command line (assuming `d2eVision.jar` and `XenosFramework.jar` are in the CLASSPATH). (For more information about CLASSPATH, see [“AFP Fonts” on page 510](#)):

```
java com.xenos.framework.tools.iores.IoResBuilder
```

Example with d2eVision.jar and XenosFramework.jar classpaths defined on the command line:

```
java -cp "C:\OpenText\OTS\<version>\lib\common\d2evision.jar";"C:\OpenText\OTS\<version>\lib\common\Xenos-framework-util.jar"
com.xenos.framework.tools.iores.IoResBuilder ZIP2IORES
```

Creating an IORES File

To create a new IORES file:

1. Start the **IoResBuilder Utility**.
2. Select **Add**.
3. Select the files and/or directories to add to the archive.
4. Select **File > Close** or close the main window.
5. When prompted to save the archive, select **Yes**.
6. Browse to the desired location and type in the name of the new IORES file (or select an existing file to overwrite).

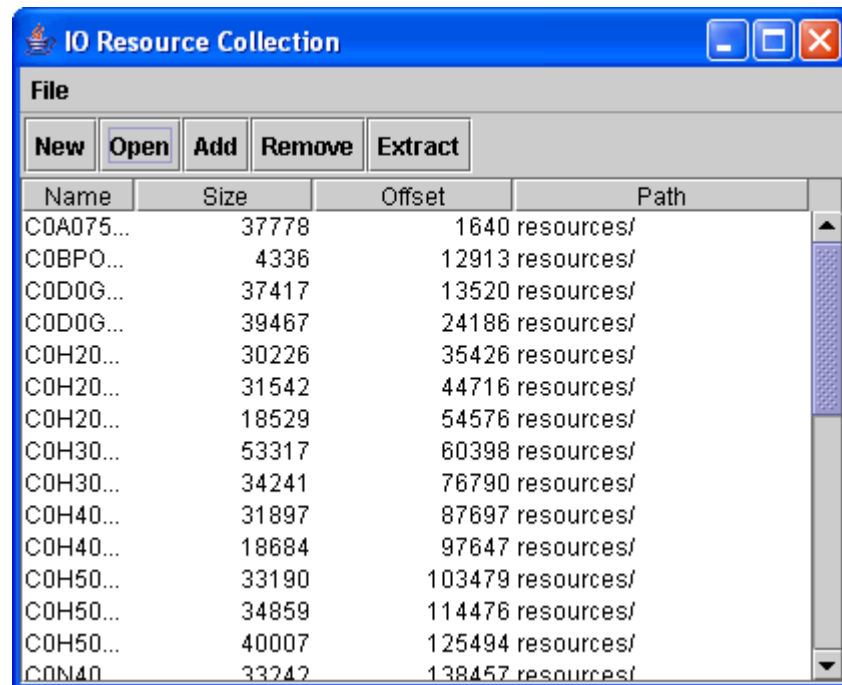


Figure 9-2: IoResBuilder Utility

Modifying an IORES File

To open/modify an existing IORES file:



Note: Individual entries may be selected by clicking on them. Multiple entries may be selected by holding down the shift key while selecting consecutive entries or by holding down the control key while selecting random entries.

1. Start the **IoResBuilder Utility**.
2. Select **Open** or select **File > Open**, and select the IORES file.
3. Perform one of the following tasks:
 - To **add entries**, click **Add** and select the files and/or directories to add to the IORES file.
 - To **extract entries**, select one or more entries, click **Extract**, and then browse to the desired location.
 - To **remove entries**, select one or more entries, and click **Remove**. Confirm when prompted.

9.5.3.6.2 Zip2IoRes Utility

The ZIP to IORES Utility, **Zip2IoRes**, converts a standard ZIP file into an IORES file. All entries from the ZIP file are copied over to the new file, and the original path and name are used from each entry.

Usage

```
java com.xenos.framework.tools.iores.Zip2IoRes <input ZIP file> <output IORES file>
```

Example

```
java com.xenos.framework.tools.iores.Zip2IoRes resources.zip resources.iores
```

Example with `d2eVision.jar` and `XenosFramework.jar` classpaths defined on command line:

```
java -cp "C:\OpenText\OTS\<version>\staging\engine\version\bin\d2evision.jar";  
"C:\OpenText\OTS\<version>\lib\common\Xenos-framework-util.jar"  
com.xenos.framework.tools.iores.Zip2IoRes Q:\d2evision\2_0\TestWorkspace\afpin\nova  
\resources.zip Q:\d2evision\2_0\TestWorkspace\afpin\nova\resources.iores
```

9.5.3.6.3 IoRes2Zip Utility

The IORES to Zip Utility, **IoRes2Zip**, converts an IORES file into a standard ZIP file. All entries from the IORES file are copied over to the new file, and the original path and name are used from each entry.

Usage

```
java com.xenos.framework.tools.iores.IoRes2Zip <input IORES file> <output ZIP file>
```

Example

```
java com.xenos.framework.tools.iores.IoRes2Zip resources.iores resources.zip
```

Example with `d2eVision.jar` and `XenosFramework.jar` classpaths defined on command line:

```
java -cp "C:\OpenText\OTS\<version>\staging\engine\version\bin\d2evision.jar";  
"C:\OpenText\OTS\<version>\lib\common\Xenos-framework-util.jar"  
com.xenos.framework.tools.iores.IoRes2Zip  
Q:\d2evision\2_0\TestWorkspace\afpin\nova\resources.iores  
Q:\d2evision\2_0\TestWorkspace\afpin\nova\resources_new.zip
```


Chapter 10

Reviewing the Output

10.1 The Output File

The output file's formats will match the generator(s) specified in your project. For example, if you select a PDF Generator, the output file will be in PDF format.

10.1.1 Reviewing Your Output File



Note: If your output needs error correction, see [“Troubleshooting” on page 647](#) for assistance.

When you run a Output Transformation Engine project, there are three possible results:

- The output is perfect, with full fidelity when compared to the original.
- The output has some problems that need to be fixed before running the project again. Usually, this has to do with font correlation, or other parameters such as color, shading, position of text, and output file attributes such as TOC, URLs, and document splitting.
- There is no output, due to an error.

10.2 Reviewing the Output tab

When a project is run, an **Output** tab corresponding to the project opens in the **Events** window. The Output tab displays information about the output file being generated and stores messages written during the operations conducted by in the debug logs. These operations messages include activities related to the loading, running, and saving of projects.

There are four sub-tabs on the Output tab:

- Job Stats
- Log
- Variables
- Results

10.2.1 Job Stats tab

The **Job Stats** tab displays details about the job, including its current status, job ticket ID, processing time, and any messages, warnings, or errors that occurred during processing.

The buttons at the top of the Job Stats tab are shortcuts to help you perform the following actions:

Icon	Description
	Runs the job or resumes a paused job.
	Pauses the job. (This option is unavailable when the job is not running, and appears blue when the job is running.)
	Stops the job. (This option is unavailable when the job is not running, and appears red when the job is running.) You can click this button to cancel an executed job that appears to be hanging or running for an unusually long time. This allows you to avoid restarting the engine to remove the job. Note that if a job contains an uninterruptable component then the job cannot be stopped.

The Job Stats table displays the following statistics:

- **Current Status.** Indicates the present standing of your job. There are three categories:
 - **Running.** Indicates that the job is currently running.
 - **Finished.** Indicates that the job is no longer running. This does not indicate that there were no errors, but rather that the job did not fail.
 - **Errored.** Indicates that the job was unable to run to completion due to an error. There will be a corresponding entry at **Failure Message** to help with identifying the cause of the error.
- **Job Ticket ID.** Specifies the unique ID assigned for the execution of this job. The format is YYYYMMDD_HHMM_IncrementalNumber. (Note that the HHMM time variable uses a 24 hour clock.)
- **Processing Time (ms).** Specifies the total time, in milliseconds, that elapsed while processing the job.
- **Failure Message.** Specifies the failure message if the result of the job is Errored. Either double-click the field entry or click the ellipsis to see the log file detailing the error.
- **Warnings.** Indicates the number of warnings from your current job. A warning usually triggers a non-fatal error. For example, if a job is configured to write to a specific file that already exists and the job does not have permission to overwrite,

the warning would be that the file already exists and the non-fatal error would be that the job did not complete successfully. Either double-click the field entry or click the ellipsis button to see the log file detailing the error.

- **Non-fatal Errors.** Indicates the number of non-fatal errors from your current job. A non-fatal error is an error condition that may or may not fail a job. Either double-click the field entry or click the ellipsis button to see the log file detailing the error.
- **Fatal Errors.** Indicates the number of fatal errors from your current job. Fatal errors are absolute error conditions where the job fails. This can be caused by things like a component unable to initialize, a job not able to be executed, etc. Either double-click the field entry or click the ellipsis button to see the log file detailing the error.

10.2.2 Log tab

The **Log tab** communicates project-related messages to indicate what actions were carried out as the job ran. Log files are created each time a job is run to help you understand what happened during the processing. This includes a report of:

- Font correlation.
- Processing of the different parsers, generators, and components (if applicable).
- Details important for problem reporting and analysis.

Double-clicking on any message in the Log tab displays the message in a separate dialog window. You can right-click in this window to copy the message text to the clipboard, which may be useful when reporting an error.

The last message written to the log indicates whether the job was successful or unsuccessful. However, it is helpful to review all the log messages after running each job until you are comfortable understanding the product's operation.

See [“Messages” on page 620](#) for a list of possible log messages.

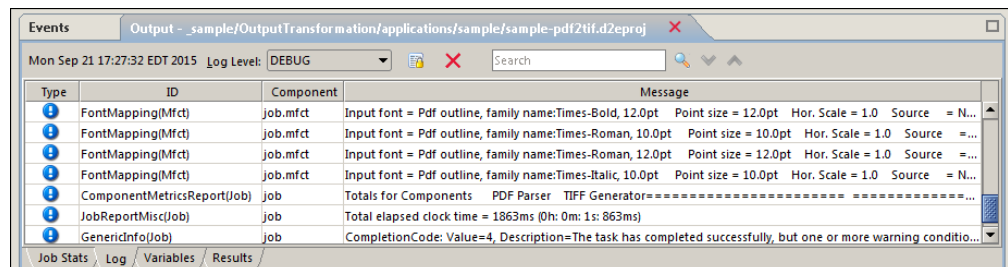


Figure 10-1: Output tab with Log tab active

The following actions can be performed on the Log tab:

-  **Prevents automatic scrolling.** Determines whether automatic scrolling during population of the log is enabled. If enabled, the default view is the

beginning, or top, of the log whereas usually, the default view is the end, or bottom, of the log.

- , **Clear**. Clears the contents of the Log tab.
- **Search**. Conducts a case-insensitive search of the messages in the Log tab. Enter your search criteria in the **Search** field and click **Find**, . If there are any matching search results, the first instance of a matching string is highlighted. If you have more than one search result, subsequent matches can be located by using the **Next**, , and **Previous**, , buttons.

Log levels

The **Log Level** list is used to filter the log messages by type. On the Log tab, using the dropdown menu, set the log filter to your desired level, and then run the job again in order for the table on the Log tab to be populated with only the classes of messages you want included.



Note: For more information on message severity, see [“Message Severity” on page 622](#).

The following log level options are available:

- **DEBUG**. Displays error messages of all types (DEBUG, INFO, WARN, ERROR, and FATAL).
- **INFO**. Displays all error messages except for messages designated as Debug.
- **WARNING**. Displays all error messages except for messages designated as Debug and Info. These messages are issued when a situation arises that may affect your final results.
- **ERROR**. Displays all error messages except for messages designated as Debug, Info, and Warning. These messages indicate a fault that may still produce an output, but that the output file is unsatisfactory due to these errors.
- **FATAL**. Displays only those messages designated as FATAL, which are typically critical error messages.

10.2.3 Variables tab

The **Variables** tab displays a list of variables used in your project, including their values and scope, following the execution of a job. You can consult this list to troubleshoot issues you are having that may be caused by variables.

The table on the Variables tab contains the following information about the job variables used in your job:

- **Name**. Specifies the name used to identify the job variable.
- **Value**. Specifies the value for the particular job variable.
- **Scope**. Indicates the area of the application the job variable relates to, which are classified as job, engine, or system variables.

If you are looking for a job variable of a certain type, you can use the buttons at the top of the Variables tab to filter the contents in the table by the variable's scope. The following options are available:

Icon	Description
	Toggles the appearance of job variables in the table on or off.
	Toggles the appearance of engine variables in the table on or off.
	Toggles the appearance of system variables in the table on or off.

To further facilitate organization of the variables used in your project, the data within the columns of the Variables table can also be sorted by clicking any of the column headers to sort the variables in either an ascending or descending ordered sequence.

10.2.4 Results tab

The **Results** tab provides a summary of the outputs from your run job. Listed are the location, name, size, and modification date of the output file.

To view one of the output files, either double-click on the name of the file or right-click anywhere in the specific file's row and from the context menu that appears, select **Open File**. If you prefer to bring up the Windows Explorer window see the full directory where the output files are contained, select **Explore** from the context menu and the directory's contents appear.

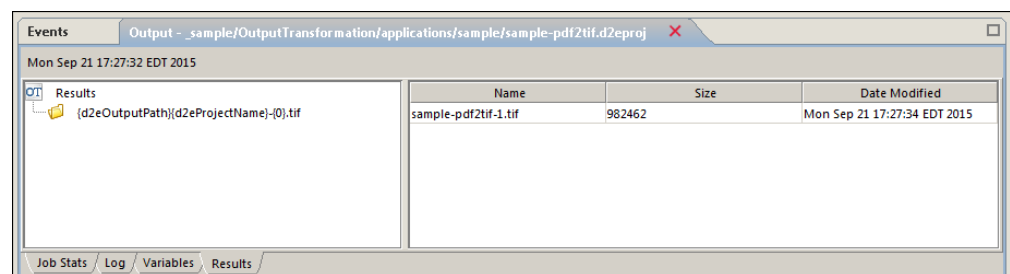


Figure 10-2: Output tab with Results tab active

10.3 Messages

communicates commands and information to its components. For example, sends a series of commands to the parsers and generators in order to execute a job. The parsers and generators in turn send informational messages back to report their progress, and if any errors have occurred.

When appropriate, certain messages are issued to the Report Log for each job that is executed. You should review these messages to ensure the program functioned as expected. They may also help you determine the cause of any problems in the processing or your output.

10.3.1 Message Format

The default message format is **XenosFramework**. This message format provides all the message details within the entry, making the log easy to read and interpret.

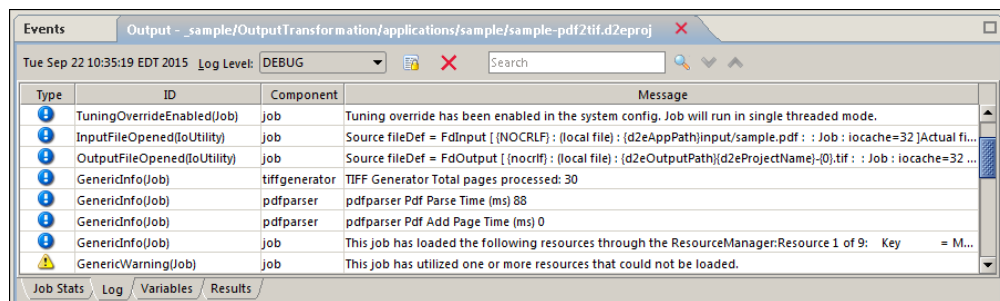


Figure 10-3: Example of XenosFramework log message format

Legacy system users may prefer to view the log messages in the **Classic** format, which uses error numbers that need to be investigated with the help of the online product documentation. The **XenoxFramework** set of messages includes the entire library of Classic messages, but they are simply displayed more accessibly.

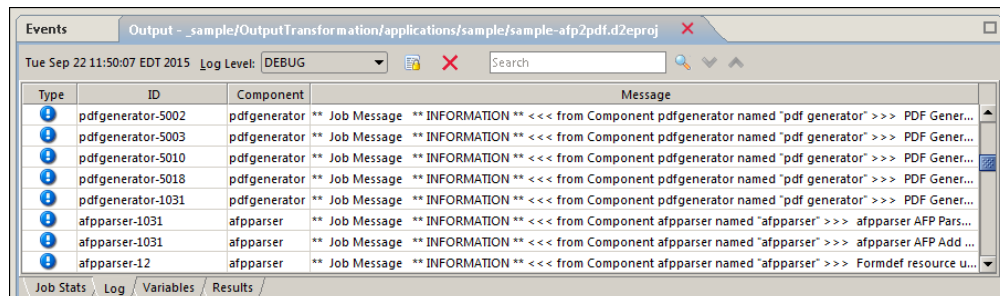


Figure 10-4: Example of Classic log message format

To change the default message format:

1. In the **File Systems** window, navigate to the folder where your system configuration file is stored.
2. Double-click the system configuration file to open the file in the Development window.
3. Expand the **LogMessageOptions** parameter and select your preferred log message format.
4. **Save** your settings.
5. **Restart** for the changes to take effect.

10.3.2 Message Categories

The following list contains all the message categories that are displayed:

- “AFP Parser Messages” on page 622
- “Command Messages” on page 625
- “Generator Messages” on page 627
- “Input Listener Messages” on page 628
- “Job Manager Messages” on page 628
- “Job Messages” on page 630
- “License Messages” on page 632
- “MFCT Messages” on page 633
- “Parameter Messages” on page 634
- “Parser Messages” on page 635
- “PDF Generator Messages” on page 636
- “Resource Manager Messages” on page 637
- “System Messages” on page 638
- “TIFF Generator Messages” on page 640
- “Table of Contents (TOC) Messages” on page 640
- “XFT Messages” on page 642

10.3.3 Message Severity

Messages are categorized by severity as follows:

- **Informational Messages**

Informational messages provide non-critical information about the program's execution, such as indicating the number of pages processed or listing the font resources used by the input file.

- **Warning Messages**

Warning messages are issued to inform users of an abnormal situation during the program's execution. The abnormal situation does not cause the program to halt processing, but may affect the final results. For example, a warning message is issued when a font does not appear in the font correlation table, causing the default correlation to be used. The resulting output may not be acceptable, but the processing can be completed. Warning messages should be verified and corrected if possible.

- **Error Messages**

Error messages indicate a serious failure during the program's execution, such as a missing font resource, or a syntax error in the font correlation table. These conditions may cause the program to halt and almost always indicate the program results are not acceptable.

- **Severe Messages**

Severe messages indicate a very serious condition occurred, causing the program to halt. Output files created by the program are not usable.

- **Fatal Messages**

Fatal messages, like severe messages, indicate a very serious condition occurred, causing the program to halt. Fatal messages are typically machine or operating system related, such as a lack of memory.

10.3.3.1 AFP Parser Messages

ID	Name	Severity	Description

1	RES NAME DOES NOT MATCH	INFORMATION	The AFP Parser encountered a resource with an unexpected name. The resource file will be used to create the output but it may be incorrect. This can occur when missing resources, typically fonts, are created from other resources to get a “clean run”.
2	RES CANNOT BE USED	WARNING	The AFP resource cannot be used due to previous errors, which could be an I/O error, a file not found, or an invalid resource.
3	UNEXPECTED SFID IGNORED	WARNING	This message will only be issued 5 times per input file or AFP resource. If more than 5 unexpected SFIDs occur, the messages will be suppressed.
4	UNSUPPORTED OBJECT IGNORED	WARNING	An object, such as BCOCA, was encountered and is not supported in the context in which it occurred. This message will only be issued 5 times per input file or AFP resource. If more than 5 unsupported objects occur, the messages will be suppressed.
5	RECORD TOO SHORT	ERROR	An AFP Structured Field record appears to be truncated or invalid. More data was expected but was not there.

6	SFID DIFFERENT THAN EXPECTED	ERROR	While parsing an AFP resource or MOD:CA input file, an out of place SFID was encountered. This may be caused by a newer release of MOD:CA that contains a new SFID, by new use of an existing SFID, or an invalid/corrupt resource/input file.
9	RES OVERLAY USAGE	REPORT	Reports the usage statistics to the report log.
10	RES PAGESEG USAGE	REPORT	Reports the usage stats of a page segment to the report log after the job is complete.
11	RES PAGEDEF USAGE	REPORT	Reports pagedef usage to report log.
12	RES FORMDEF USAGE	REPORT	Reports formdef usage to the report log.
7	BAD SFID ENCOUNTERED	ERROR	The SFID could not be parsed due to invalid, inconsistent and/or unrecognized values.
18	SHORT IRD	WARNING	This message is issued when IRD data is shorter than it should be.
13	RES FONT USAGE	REPORT	Reports font usage to the report log.
14	RES CODED FONT USAGE	REPORT	Lists the coded fonts that were used during the job.
15	RES CHARSET USAGE	REPORT	Lists the names of all character sets used during the job.
16	RES CODEPAGE USAGE	REPORT	Lists all codepages used during the job.

17	RES GROUP OVERRIDE	WARNING	A resource has been found in an inline resource group, but was not used because the parameter file specified a different one. Seldom used, so this message informs users if they use the parameter incorrectly.
19	NO DEFAULT FONT DEFINED	WARNING	Occurs when the input stream selects a font entry for a font that has matching inline font and no corresponding default font added to the project file. In many cases (especially when rasterizing fonts) this is something the user will want to correct.

10.3.3.2 Command Messages

ID	Name	Severity	Description
2013	JOB REQUEST ERROR	FAIL RESPONSE	This message is raised by Input monitors when more than one action is requested on the same job at the same time.
2001	SHUTDOWN SYSTEM	REQUEST	This message is used to instruct the system to immediately close down. Any queued jobs or jobs being processed will be lost.
2002	RESET SYSTEM	REQUEST	This message ID signifies that the system should reset/reload its global configuration parameters.

2003	SUBMIT JOB	REQUEST	This message signifies a job is ready for the system to process.
2004	SHUTDOWN REQUEST	REQUEST	This message is a request for the system to shut down. If there are no jobs running, the system will comply. If there are jobs running then the system will respond with a SYSTEM BUSY message.
2005	SHUTDOWN D2E	REQUEST	This message specifies an unconditional shut down of the system. Note: This command cannot be aborted.
2007	FLUSH SYSTEM	REQUEST	This Command Message ID signifies that the system should not accept any more jobs, continue processing any active or queued jobs, then shut down without further user intervention.
2008	SUSPEND JOB	REQUEST	This message is a request to suspend the processing of a previously submitted job. This will pause a running job if it is active, or if the job has not yet been started it will be held in the queue. First Parameter is Job Name.

2009	STOP JOB	REQUEST	This message is a request to STOP an active job. This will cause the active job to terminate immediately and the job will be placed back in the queue in a suspended state (so it will not be processed again until it is resumed).
2010	RESUME JOB	REQUEST	This message is used to request that a job previously suspended be resumed.
2011	GET JOB STATUS	REQUEST	This message is used to request the status of a given job.
2012	GET JOB MANAGER STATUS	REQUEST	This message is used to request the status of the Job Manager.

10.3.3.3 Generator Messages

ID	Name	Severity	Description
27000	GENERATE SUCCESSFUL	SUCCESS RESPONSE	The named generator has completed processing successfully and generated the number of pages/ documents indicated in the message. Note that the number of pages generated may not equal the number of pages parsed if code has removed or inserted extra pages.
27002	GENERATE STATUS	INFORMATION	This is an informational message that the generator can send from time to time to report on its progress by indicating the number of pages it has processed so far.

27003	GENERATOR ABORTED	ERROR	The Generator has been aborted by some external agent such as an operator or by an externally submitted command.
27001	GENERATE EXCEPTION	ERROR	The named Generator encountered an error while processing. An error message indicating where and what error occurred will also be included in this message.

10.3.3.4 Input Listener Messages

ID	Name	Severity	Description
8000	SEND EVENT FAILURE	ERROR	This message is sent to the report log if an event cannot be sent to an RMI listener class.

10.3.3.5 Job Manager Messages

ID	Name	Severity	Description
3001	REPORT HEADER	REPORT	Sent to begin a Job Manager Status Report.
3002	REPORT QUEUE HEADER	REPORT	Sent after a REPORT HEADER message. Storing this as a message enables the report headers to be translated into different languages.
3003	REPORT QUEUE ENTRY	REPORT	This message enables the report entries in the report to be sent to any component which has subscribed to the message.
3004	REPORT JOB WAITING	REPORT	Sent after 0 or many REPORT QUEUE ENTRY messages.

3005	REPORT RUNNING HEADER	REPORT	Contains the Header information for a report. Held in this format so that it can be translated.
3006	REPORT RUNNING ENTRY	REPORT	For internal use only: Enables the report entries to be sent as Messages to each of the registered subscriber components.
3007	REPORT FOOTER	REPORT	For internal use only: Enables the end of report string to be sent to each of the components which has subscribed for the report.
3008	REQUEST TOO MANY THREADS	FAIL RESPONSE	The Job has requested more threads than the total number of threads that have been configured to run with the Job Manager.
3009	NOT ENOUGH THREADS YET	INFORMATION	Not enough threads currently available to run the job. The job will be queued until threads are freed up by a currently running job.
3010	EXCEPTION	ERROR	Job Manager has encountered an Exception.
3011	JOB START FAILED	FAIL RESPONSE	Response indicating that a job has failed to start.

10.3.3.6 Job Messages

ID	Name	Severity	Description
1001	STARTED	INFORMATION	The specified job has been started. The message shows the name of the person or system making the request and a reference to the job provided by the requester.
1005	QUEUED	SUCCESS RESPONSE	The job specified has been put in the processing queue and will be picked up when sufficient resources become available.
1007	RESUMED	SUCCESS RESPONSE	The specified job has resumed processing after it has been suspended.
1011	SUSPENDED	SUCCESS RESPONSE	The specified job has received a request to suspend itself. The job will not continue until a 'RESUME' command is received.
1015	NO NEW JOBS ALLOWED	FAIL RESPONSE	The system is currently allowing jobs in progress to finish and is not accepting new jobs.
1017	DELETED	SUCCESS RESPONSE	This message indicates that the job has been deleted from the queue in response to a STOP JOB request.
1002	JOB STATUS	INFORMATION	This message shows the status of the job.
1004	FINISHED	INFORMATION	The job specified has been finished.
1008	JOB REPORT MISC	REPORT	This message reports miscellaneous properties and events.

1010	STOPPED	SUCCESS RESPONSE	This message can occur either when the job reaches a successful conclusion or when a request from an external agent to stop the job is received.
1014	ERROR	ERROR	The specified error has occurred in the job.
1016	EXCEPTION	ERROR	The specified error has occurred in the job.
1020	APPLICATION PARM VALUES	INFORMATION	This message lists all the project parameters.
1022	INVALID APPLICATION	ERROR	Issued when a job file contains an invalid application name.
1021	COMPONENT ERROR	ERROR	This message is generated if a component passes an error to the component manager. The message contents will depend on the class of component that errored.
1023	INVALID MFCT	ERROR	This message is issued when a job fails on startup because the MFCT cannot be loaded. Either it is missing or there was an error processing it.

10.3.3.7 License Messages

ID	Name	Severity	Description
7000	LICENSE CORRUPT	ERROR	Indicates that the license file failed to load for some reason. Could be physically missing or corrupted because fields have been modified or removed.
7001	LICENCE WILL EXPIRE	WARNING	Warns the user that the license will expire in xxx days.
7002	LICENSE EXPIRED	ERROR	The license has expired, and the customer must obtain a new license.
7003	FEATURE NOT LICENSED	ERROR	Notifies the user that a feature was encountered that is not licensed.
7004	FEATURE REQUIRED	ERROR	Some components require the existence of another component in order to run.
7005	SOFTWARE LICENSED	INFORMATION	Provides notification that the initial phase of the license authenticator passed. The license appears to be valid at the time.
7006	INVALID HOST	ERROR	Issued when the current host name does not match the host specified in the license file.
7007	INVALID PLATFORM	ERROR	Message raised when the platform is invalid.

10.3.3.8 MFCT Messages

ID	Name	Severity	Description
15001	UNMAPPED FONT	WARNING	Describes an input font that was not mapped by the MFCT.
15003	UNKNOWN COLOR	WARNING	The input referred to a color name that is not defined in the MFCT.
15005	INVALID OUTPUT FONT	WARNING	An unknown output font name was encountered in an input font.
15007	INVALID TRAN TAB	WARNING	An invalid translation table name was encountered for an output font.
15000	MAPPED FONT	INFORMATION	Describes the mapping of an input font to an output font by the Master Font Correlation Table (MFCT).
15002	UNKNOWN AFP COLOR TABLE	WARNING	The input tried to use an AFP Color Table that is not defined in the MFCT.
15004	INVALID COLOR NAME	WARNING	An unknown color name was encountered in an AFP color table.
15006	EXCEPTION	ERROR	An exception occurred within the MFCT.

10.3.3.9 Parameter Messages

ID	Name	Severity	Description
9000	UNKNOWN PARM	WARNING	An unknown parameter was encountered while parsing an Application XML file.
9001	UNKNOWN OBJECT	WARNING	An unknown object was encountered while parsing an Application XML file.
9002	UNKNOWN ELEMENT	WARNING	An unknown element was encountered while parsing the Application XML.
9003	INVALID BOOLEAN	WARNING	An invalid boolean value was encountered while parsing the Application XML.
9004	INVALID INT	WARNING	An invalid integer number was encountered while parsing the Application XML.
9005	INVALID FLOAT	WARNING	An invalid float value was encountered while parsing the Application XML.
9006	INVALID ENUM	WARNING	An invalid enum value was encountered while parsing the Application XML.
9007	INVALID FILEDEF	WARNING	An invalid filedef was encountered while parsing the Application XML.
9008	INVALID MIN	WARNING	A value was encountered that was smaller than the minimum allowable value.

9009	INVALID MAX	WARNING	A value was encountered that was greater than the maximum allowable value.
9010	INVALID MINMAX	WARNING	A value was encountered that was greater or smaller than the maximum/minimum allowable values.
9011	INVALID PARSER	WARNING	An invalid parser type was encountered while parsing the Application XML.
9012	INVALID GENERATOR	WARNING	An invalid generator type was encountered while parsing the Application XML.
9013	MISSING REQUIRED PARM	WARNING	A required parameter was not set while parsing the Application XML.

10.3.3.10 Parser Messages

ID	Name	Severity	Description
26000	SOURCE FILE NOT FOUND	ERROR	The path may be incorrect or the source file may be corrupt.
26002	PARSER EXCEPTION	ERROR	An error has occurred in the parser while processing the input file.
26004	PARSER ABORTED	ERROR	The parser has been aborted by some external agent such as an operator or by an externally submitted command.

26001	PARSE SUCCESSFUL	REPORT	The parser has successfully parsed the source file. This message indicates the number of pages and the number of bytes processed.
26003	PARSE STATUS	REPORT	This message describes the current state of the parser in terms of the number of whole pages processed and the total number of bytes which make up those pages.

10.3.3.11 PDF Generator Messages

ID	Name	Severity	Description
5000	DOCUMENTS CREATED	REPORT	Number of documents created.
5001	FONTS USED	REPORT	Number of fonts used in run.
5002	FONT NAME	REPORT	Name of a font used in run.
5003	FORMS USED	REPORT	Number of forms used in run.
5004	FORM NAME	REPORT	Name of a form used in run.
5007	NON BASE 14	REPORT	Reports if PDF font is not a Base 14 font.
5009	WRONG AFM	REPORT	PDF font is invalid for AFM file.
5012	MISSING LINK TARGET	WARNING	The target for an internal link was not found. The link was directed to the page on which it was defined.
5014	MISSING FONT FILE	ERROR	A font metrics file (AFM, PFB, etc.) was not found.
5016	FONT CREATION ERROR	ERROR	Unable to create PDF font.

5018	IMAGES CACHED	REPORT	Number of images cached by generator.
5006	PATTERN NAME	REPORT	Lists the Pattern name used.
5008	NON BASE 14 NO AFM	REPORT	PDF font is not a Base 14 font and no AFM was specified.
5010	PATTERN REPLACEMENT LIST	REPORT	Pattern replacement list for this generator.
5011	ELEMENT EXCEPTION	ERROR	There was an error processing this specific element. Processing will continue.
5013	INVALID FONT FILE	ERROR	Format of font metrics file is invalid.
5015	FONT FILE ERROR	ERROR	I/O error occurred while reading a font metrics file.

10.3.3.12 Resource Manager Messages

ID	Name	Severity	Description
12000	RESOURCE FILE LOAD ERROR	ERROR	The Resource Manager was unable to load the file of persisted resources.
12001	RESOURCE FILE LOAD OK	INFORMATION	The Resource Manager successfully loaded a persisted resource file.
12002	RESOURCE FILE SAVE ERROR	ERROR	The Resource Manager was unable to save the persisted resource file.
12003	RESOURCE FILE SAVE OK	INFORMATION	The Resource Manager successfully saved the persisted resource file.
12004	CACHE CONTENTS	INFORMATION	A description of the contents of the Resource Manager cache.
12005	FLUSH SYSTEM RESOURCES	REQUEST	Request to flush all system resources.

12006	FLUSH ALL RESOURCES REQUEST	REQUEST	Request to flush all resources.
12007	FLUSH APP RESOURCES REQUEST	REQUEST	A request to flush the resources for a single application.

10.3.3.13 System Messages

ID	Name	Severity	Description
4002	RESET SUCCESSFUL	INFORMATION	System has been successfully reset.
4004	RESET FAIL	FAIL RESPONSE	The request to reset the system has failed.
4001	STARTUP SUCCESSFUL	INFORMATION	The system has been successfully started.
4007	SYSTEM STATUS	INFORMATION	This message contains details of the status of the system as requested in the system configuration parameters.
4008	TRANSFORMATIO N NOT LICENSED	ERROR	This message indicates that the SUBMIT JOB request was trying to utilize unlicensed features.
4009	LICENSE AUTHENTICATION ERROR	ERROR	This message indicates that the license authentication failed. The reason could be an invalid license file. Further details are in the message.
4010	SYSTEM ERROR	ERROR	This message indicates that an error has occurred within a system function.
4011	COMPONENT SHUTDOWN	INFORMATION	This message indicates that a system component is shutting down.
4012	JOB NOT FOUND	ERROR	This message indicates that the job was not found.

4013	INVALID COMMAND SEQUENCE	ERROR	Commands in a command file sometimes have to be processed in a certain order. If the sequence of commands cannot be processed then this message will be generated.
4014	INPUT MONITOR LISTEN ERROR	ERROR	This message indicates that an exception has occurred while listening for requests.
4015	REPORT LISTENER ERROR	ERROR	An error occurred in the Report Listener while processing a message. The listener will no longer receive messages from this report.
4016	INVALID REPORT LISTENER	ERROR	This message indicates that an error has occurred while adding a listener to a report.
4017	INPUT MONITOR ADD LISTENER ERROR	ERROR	This message indicates that an exception occurred while adding a listener to the Input Monitor.
4018	COMPONENT STARTED	INFORMATION	This message indicates that the named component has started.
4019	INPUT MONITOR ADD LISTENER	INFORMATION	This message indicates that the named input monitor source is being added.
4003	SYSTEM BUSY	WARNING	The system is currently busy.
4005	COMMAND FAIL	FAIL RESPONSE	This message indicates that the system has failed to process the command specified in the message.

4006	COMMAND SUCCESS	SUCCESS RESPONSE	This message indicates that the system has successfully completed the command detailed in the message.
4020	JOB HISTORY CLASS INVALID	ERROR	This message is thrown if Job Manager is unable to create the class specified in the Job History Data parameter of the System parameter file.
4021	JOB HISTORY ERROR	ERROR	This message is raised by a subclass of Job History Mail Subscriber which is responsible for writing and querying job history information to a particular storage medium. The type of storage medium involved depends on the class.

10.3.3.14 TIFF Generator Messages

ID	Name	Severity	Description
16000	DOCUMENTS CREATED	REPORT	Total number of documents created for this job.

10.3.3.15 Table of Contents (TOC) Messages

ID	Name	Severity	Description

28001	TOC ENTRY MISSING VALUES	WARNING	The Table of Contents entry was defined with embedded field values which were not encountered before the end of the document, so the field placeholders remain in the entry text. The entry will exist in the output but the placeholders will remain.
28003	TOC EXCEPTION	ERROR	Gives detail of an exception that occurred in the TOC component while processing an event.
28000	TOC ENTRY MISSING PARENT	WARNING	The XFT Table of Contents builder had a TOC entry which did not have any parent assigned to it by the end of the document. This means that you may have a trigger for a parent TOC entry set to something that does not occur before the end of the document.
28002	TOC DOC INFO	INFORMATION	Information message from the Table of Contents component relating to the end of a document. This information may include a text version of the bookmark output tree.

10.3.3.16 XFT Messages

ID	Name	Severity	Description
6000	SECTION ALREADY EXISTS	ERROR	The section given already exists within the document.
6002	EXCEPTION ENCOUNTERED	ERROR	An exception that has been thrown by the Java Virtual Machine. These exceptions occur when no localized version of the exception message is available or the exception is caused by an outside issue. These exceptions should be reported back.
6004	SET COLUMN LOCATOR ERROR	ERROR	An exception was encountered when trying to add the XFT Column Locator defined in the parameter file. Column Locator Class {0}, Exception: {1}
6005	XFT EXTENSION CLASS NOT FOUND	ERROR	The class for performing certain actions such as table row validation or locator pattern matching can be specified by the user in the parameters. This message indicates that the class as specified cannot be found.
6006	COLUMN ALREADY EXISTS	ERROR	The column name provided for a new column already exists in this table.

6008	TABLE ROW BREAK NO ROWS	WARNING	This warning message is raised by the system when it cannot find any rows on a table which has been identified within the current page.
6010	THRESHOLD does not EXIST	ERROR	The specified threshold was not defined within the XFT application.
6011	XFT FIELD VALUE INVALID	WARNING	This text field has been validated against the rules for validation and is found to be invalid. The action taken when an invalid field is encountered is configured by a separate parameter. Possible actions are; ignore, log, or stop job.
6013	XFT ROW VALIDATION FAILED	WARNING	Warns that results of the table row validation is misleading due to missing or invalid numerical values.
6014	XFT TABLE ROW INVALID	WARNING	This message occurs when invalid data is encountered. The user specifies validation criteria in the parameters, and data which does not meet these criteria are logged or cause the job to halt.
6015	XFT TABLE CELL INVALID	WARNING	This message occurs if a column within a row of a table does not meet its validation rules and the user has configured either a log or halt job action in the parameters.

6016	XFT PARAMETER MISSING	ERROR	This message occurs when an XFT parameter, such as Whole Page Threshold, is found to be undefined.
6017	XFT PARAMETER INVALID	ERROR	This error is raised by the system when a parameter for XFT has an erroneous value.
6001	SECTION NOT FOUND	ERROR	The given section was not found in this document. This error occurs when the specified section name is not found. The cause may be incorrect case, as the name is case sensitive.
6003	ADD EVENT HANDLER EXCEPTION	ERROR	One of the XFT application event handler classes in the application parameter file caused an exception while trying to add an instance of this class to the XFT application. Illegal Access Exception - if the class or initialize is not accessible.
6007	TABLE PROCESSING EXCEPTION	WARNING	XFT encountered an exception processing Table: {0} at row number {1}, column {2}. The encountered exception was :- {3}
6009	XFT EXTENSION SETTING INVALID	ERROR	This message is raised by the system when a properties object passed to one of the XFT extension classes does not contain a required named value, or the value is invalid.

6012	XFT EXTN SETTING UNKNOWN	WARNING	This message is added to the report log as a warning to the user that the property is not recognized, and will be ignored.

Chapter 11

Troubleshooting

The following information will help you evaluate your output and solve the most common problems or questions related to difficulties encountered during processing.

By reviewing these suggestions and taking the recommended actions, you may be able to solve the problems yourself without needing to call Customer Support. However, our Customer Support staff are glad to assist you at any time.

11.1 First Steps

If output is being generated, determine exactly how it is different from the original input. Compare the hard copy printout of the input file with the output received. If a printout of the input file is not available, you need to obtain one. That is the only way to know for sure what it should look like. If the input file will not print, it will not transform either. Bad input causes bad output.

Here's a checklist for reviewing your output:

- Was an output file written?
- Is any text missing? Whole lines, or individual characters?
- Are lines and boxes visible? Are they positioned correctly?
- Are graphics visible? Are they positioned correctly?
- Is shading visible? Is it positioned correctly? Is it the correct color?
- Are page breaks occurring in the right places? Are any pages missing, or extra blank pages added?
- Do the fonts match? Are the characters too big or too small?
- Are the typeface(s) those that were expected?
- Are the characters correct? Are only some characters wrong, or all?
- Is the text in the right places? Too far left or right? Too far up or down?
- Is the word spacing correct? Are the spaces between words too big or too small?
- Are the colors correct?

11.2 Look for Possible Causes

Try to determine simple causes for the problem. Many problems are caused by typing errors, or errors in downloading the input or resource files.

Check the following:

- “Check the Report Log” on page 648
- “Check the Parameters” on page 649
- “Review Font Table” on page 650
- “View the Diagnostic Log” on page 651
- “Specific Problems and Solutions” on page 651
- “Frequently Asked Questions” on page 656

If your problem still persists after troubleshooting, report the problem to Customer Support.

11.3 Check the Report Log

The report log is the first step in troubleshooting. It will record messages that may be helpful in finding a solution. This file actually echoes the settings the program is using. A review will indicate whether or not the program is using the parameters and settings that you defined, versus using default settings. For more information, see “Reviewing the Output” on page 615.

Check for any error or warning messages. These may indicate an unrecognized keyword or invalid value. See “Messages” on page 620.

A message of error on open, record 0 usually indicates the file could not be found.

An error may also state that the font is not in the font correlation table or that the .afm filename was not given for a non-Base 14 font requested for use in PDF output.

The log file contains a list of all the fonts used. This is useful for identifying problem fonts. Unmapped fonts may be mapped by double-clicking a **font message** from the report log.

11.4 Check the Parameters

Verify all of the parameters contained in your project.

- Are the parameter keywords spelled correctly?

If the report log says “Unknown Keyword ignored” the keyword is probably misspelled, or in the wrong section. This is likely to occur if you have entered incorrect input into the parameters window.

- Are all the resource files that are called by this job present?

(Fonts; forms, overlays, or macros; page segments, IMGs or LGOs; or PageDef and FormDef).

- Missing fonts can cause missing text or poor text positioning. Missing graphics will not be displayed in the output.
- Missing forms can cause missing text or graphics, and incorrect positioning of other data.

- Are the resources really located in the directories specified in the resource path parameters?

(Fdfonts, Fdimgs, Fdpagesegs, and so on.)



Note: Do not forget to check that the paths and file names are entered correctly. Output Transformation Designer requires absolute file paths, not relative file paths, so the drive letter must be specified.

- Are the file extensions specified correctly in the resource path parameters?
- Is there a variable %s for the filename in the resource path parameters?
- Is the font correlation table specified correctly?
- Is the parameter keyword, path, and file name complete and correct?
- Do the I/O directives correctly match the input file?
- Does the file contain fixed length records, CRLF, or length fields?

If fixed length line data stair-steps down the page, the value you gave for the record length is probably incorrect. If the text has numbers at one or both ends of the lines, it may indicate length fields that are being read as data because the program does not know there were length fields that it needed to process. Wrong I/O settings will sometimes cause multiple errors in the output.

Check the FDOOutput parameter to verify the path and filename for the output if the output file is missing.

If the directory to write to does not exist, the file cannot be written.

- Is the CC set correctly?

If odd characters are displayed in the left margin, and line spacing is wrong, those may be carriage control values that the program was not told to read. If the first character of each line is missing, the program may be trying to read a

carriage control that is not there, and processing the first byte of the data as though it was a carriage control.

- Are the starting PageDef/FormDef values set correctly?

These should match what is normally started on the printer.

11.5 Review Font Table

If problems with fonts are identified, a review of the font table may indicate the source of the problem.

11.5.1 Is the font being mapped?

If not, a default may have been used.

11.5.2 Is the font mapped twice by family and by font/character set?

If so, the mapping for the individual font will override the family mapping.

11.5.3 Is the point size correct?

If the output font is too large, the words may overlap on the output, or extend too far to the right. If the output font is too small, the words may be too widely spaced, or too far to the left on the page. Changing the point size or the ratio value will change the size of the output font selected.

11.5.4 Is the font being rasterized?

If fonts are being rasterized and the input font file is not present, the fonts will disappear.

11.5.5 Is a color table defined?

To get color output, color mappings must be set in the font table. To use any color in a font mapping, the RGB values for that color must first be defined in the color table section of the font correlation table.

11.5.6 Are the colors defined correctly?

Is the color name spelled correctly or at least consistently? Black is 0,0,0 in RGB. Check the color of the font. If it is set to white, text may disappear in the output. Form color, image and graphic colors, and shading colors are all affected by color table settings.

11.5.7 Are external filenames specified?

External non-Base 14 PDF Type 1 fonts have different internal font names and external file names. These must both be specified and paired in the font correlation table.

11.6 View the Diagnostic Log

To provide information that is useful both in problem diagnosis and for Customer Support, the diagnostic log in Output Transformation Designer can be enabled. Select **View > Options**, and click the Debug tab. On this tab you can enable/disable the diagnostic log and change its location, as well as view the log.



Note: Enabling the log may cause Output Transformation Engine to run more slowly.

11.7 Specific Problems and Solutions

This section discusses some typical problems that you may encounter and their associated troubleshooting information.

11.7.1 Output File Does Not Exist

There are several possible reasons this occurs:

- The output filename or path may not be correct.
- A tool component error may be causing the project to run without creating output.
- A fatal error may have occurred. Check the report log (for more information on reports and logs, see [“Reviewing the Output tab” on page 615.](#))

11.7.2 Output Text is Messy

You may find that the text in your output document overwrites itself or the words are too closely jammed together. There could be a couple of reasons for this:

- If you have used Relative position instead of Word position in your project file for the Position parameter, the text may be starting in the wrong place. Check your project file and try using Word position instead.
- If the fonts in your font correlation table are not mapped correctly, the generator may be using the information for the default font. These settings are usually Helvetica red. If your output appears as in the figure below, check your font correlations in the font table.

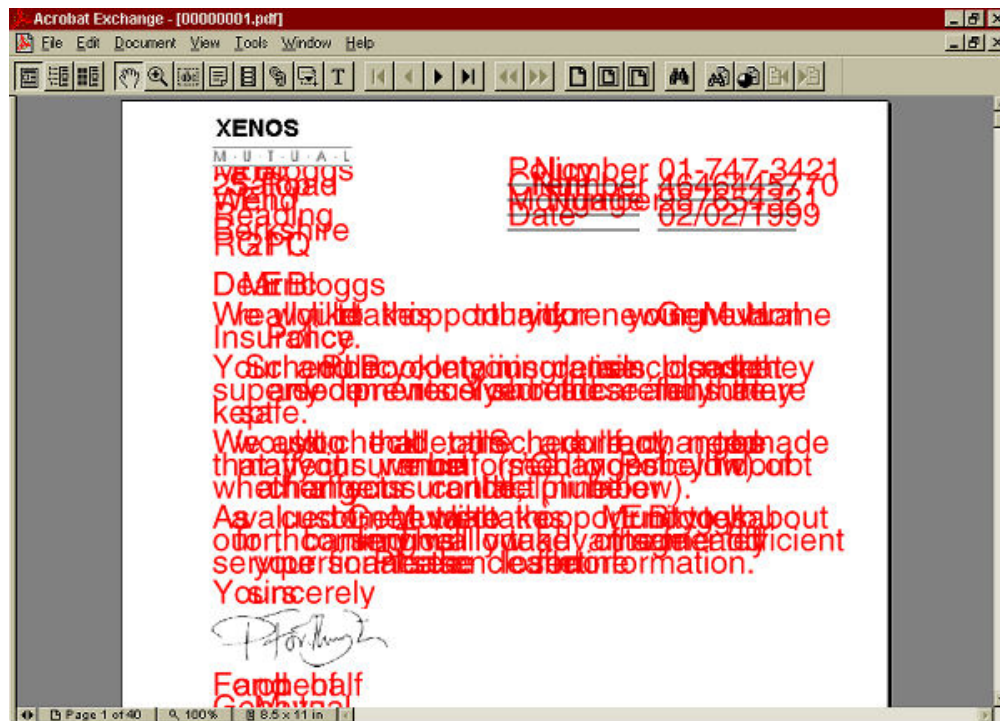


Figure 11-1: PDF Output with default font

11.7.3 Output Text is Unreadable

Your output file may contain text that looks like Greek or symbols.

^,CE <|| <<T|||-. |||CE _ ><CE OUECT />
 </- %CE fi + t /> CE !> CE ' fi « ± CE / fi <
 -,CE " (CE - < t || fi / %
 -,CE " (CE ¶ ?) <-
 -,CE (/ -> < > /> • CE -

 ^? CE < ? CE > < / • CE «) —

 „? « fi CE /> • t — < /> CE > < ?> - ? CE CE
 AE
 " «> < ? _ fi CE < fi - • AE ~~~~~
 » ,CE < -< — ± (" " /> , • ?
 /- % + % — ± (" " /> , • ?

Figure 11-2: Example of unreadable output text

There are a couple of possible reasons for this.

- You may be attempting to use too many font files. The list of font files is built from the wildcards in the FD fonts parameter. The program performs a directory search using the wildcards in the FD parameter and shows all the files which meet the criteria. This includes files that may not apply for that particular Project. The redundant font files should be removed. However, the only time you can remove an individual file is if you have explicitly added that file.

If you want to remove a specific file when you have a wildcard specified you can either:

- Remove that file from the source directory and leave the FD parameter unchanged.
- Add an explicit reference to each file which you do require, excluding the file which you want to be removed.
- Your font correlation table may be mapped incorrectly. Check the font mapping, correct any errors, and re-run the project.

Note: With 4.0, changes have been made to the way the engine uses TrueType symbolic font encoding. If you find that a project that was mapped prior to version 4.0 is no longer displaying text characters correctly in version 4.0 or later, you may need to rebuild your MFCT.

11.7.4 Windows Not Visible

If you are unable to see the parameters window, the messages window, or the window:

1. Click **View > Options**.
2. Select the **Reset Window Layout and Menus** check box. (For more information about View Options, see [“Select Open View Options” on page 214](#).)
3. Click **OK**.

All windows reset to their default configuration.

You may be unable to see some of the properties windows because you have other windows open on top of them. To view any hidden windows, click **Window > Cascade** to display all open windows in a cascade format. See Window Menu.

11.7.5 Windows Display Problems

If you encounter any additional problems with the display of your windows or menus, and resetting the windows layout doesn't help, locate a file named `d2elayout.wlf` in the folder named

```
\Program Files\xenos\d2e-vision\DeveloperStudio
```

Deleting this file causes a new one to be created the next time is launched. The new file is created with a set of 'clean' defaults and usually fixes any display problems.

11.7.6 No Project Window

If you double-click the green Projects icon or project name in the tree, it collapses all the branches of the tree, making it appear as if the tree has no sub-elements, as pictured below.

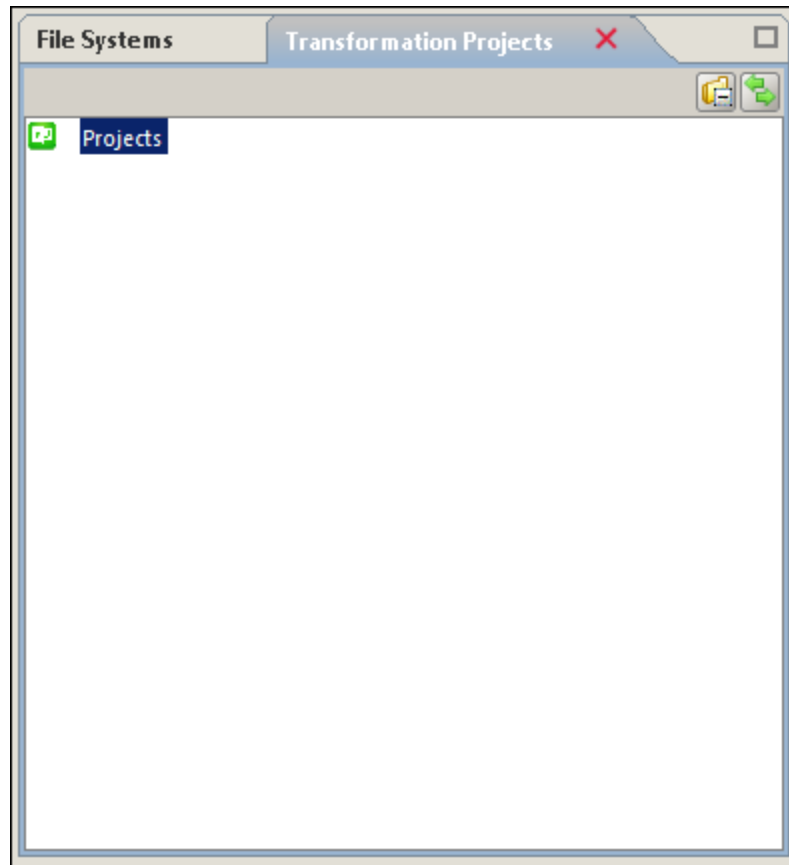


Figure 11-3: Project window with project tree collapsed

Double-click the green Projects icon or name again to re-display all of the project sub-elements.

11.7.7 Error Occurred During Open

This error message most often occurs when:

- Output Transformation Designer cannot find or access the file called in the path. For example, your output file may not have been written because of an error in the output path or file name.
- You are viewing a PDF document, and attempt to run a Output Transformation Engine project that writes to the same file name. Output Transformation Designer will not be able to open that document for writing as it is already in use by Adobe Acrobat.
- The file does not exist.

11.7.8 Data not Picked Up by Fields

If the data extraction operation from the **Run** tab does not pick up the values for your fields, it may be because the field you defined is too small. **Resize** your field to ensure the program can fully read the contents.

11.8 Frequently Asked Questions

The following are some frequently asked questions and answers.

On the Select Resource Path pane, what is the purpose of Jump to Relative Path?

On the **View > Options** dialog, logical names can be defined for directory locations, such as `CommonAfpResources`. An `FDef` can be based upon this logical location. The parameter `fdAfpFonts` could be `{CommonAfpResources}fonts/*.fnt` and the parameter `fdPagedefs` could be `{CommonAfpResources}pagedefs/*.pde`. This allows projects to be relocated based upon the definitions of these logical locations. The **Jump** lets you select one of the defined logical locations from which to select a path.

On the Select Resource Path pane, what is the purpose of Cache?

This defines Job, Application, or System. The first time a resource, such as an AFP Font, is loaded into memory, it is placed in the given cache. For Job, it is shared by any process during that job. For Application, the resource is cached in memory and available the next time a project from that application is run; the resource is not loaded from disk a second time. For System, the resource is available for any job and any application.

What is the purpose of the Tuning parameter?

The parsers and generators enable you to tune the performance by adjusting the number of threads used for the component.

I am getting several messages about unmapped fonts. Why do I need to map fonts for TIFF output?

You can have outline fonts that cannot be translated into raster fonts. In this case you need to map them to True Type fonts. You can also force a True Type font to be used in place of the raster font. The MFCT must also be able to translate the font's encoding into Unicode for XFT processing.

My output contains elements that appear behind other elements on the page when they should be on top, or some elements are covering areas of the page but should be placed in the background. What options do I have to try to correct this?

Typically when a parser processes an input file, a `XifPage` object is created, which represents a single page in the XIF Intermediate format, before being processed by a generator. `XifOptions` can then be applied and if the overlays are not configured properly, then you may end up with elements incorrectly being placed either below or on top of other elements.

Under **XifDrawingOptions**, the value of the **DrawingOrder** parameter manages the drawing order of XIF graphical elements. When set to the default **Optimized** value, elements of the same type are grouped together in the output document structure and in most cases, the Optimized setting is appropriate. However, sometimes elements get incorrectly placed above or below other elements as Optimized does not consider an input document's original drawing order. To fix this issue, set the value to **Original**, which ensures that subsequent generators in your project preserve the original drawing order of the input document.

