

## OpenText™ Intelligent Capture

### **Designer Guide**

This guide describes how to use Intelligent Capture Designer to create, configure, deploy, and perform end-to-end testing of a capture system.

ECPCORE220300-CPD-EN-01

---

**OpenText™ Intelligent Capture  
Designer Guide**

ECPCORE220300-CPD-EN-01

Rev.: 2022-July-20

**This documentation has been created for OpenText™ Intelligent Capture CE 22.3.**

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>

**Copyright © 2022 Open Text. All Rights Reserved.**

Trademarks owned by Open Text.

Adobe and Adobe PDF Library are trademarks or registered trademarks of Adobe Systems Inc. in the U.S. and other countries.

One or more patents may cover this product. For more information, please visit <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

<b>1</b>	<b>Getting started .....</b>	<b>13</b>
1.1	Features .....	14
1.2	High-level steps .....	17
1.3	User interface .....	20
1.3.1	Top banner .....	20
1.3.2	Area navigation .....	21
1.3.3	Problems tab and status bar .....	22
1.4	Navigating Designer .....	23
1.5	Understanding SnapMatch .....	24
<b>2</b>	<b>Configuring the design environment .....</b>	<b>25</b>
2.1	Configuring the working directory .....	25
2.2	Adding custom modules to the palette .....	26
<b>3</b>	<b>Preparing a capture system .....</b>	<b>29</b>
3.1	Defining capture systems .....	29
3.1.1	Adding a new capture system .....	30
3.1.2	Adding samples .....	30
3.1.3	Renaming a capture system .....	31
3.1.4	Deleting a capture system .....	31
3.2	Defining capture servers .....	32
3.2.1	Adding a capture server .....	32
3.2.2	Editing a capture server .....	33
3.2.3	Deleting a capture server .....	33
3.2.4	Connecting to a capture server .....	34
3.3	Defining connections .....	34
3.3.1	Creating an ODBC connection .....	36
3.3.2	Creating a Documentum connection .....	38
3.3.3	Creating a file system connection .....	41
3.3.4	Creating an email connection .....	42
3.3.5	Creating a CMIS repository connection .....	45
3.3.6	Creating an ApplicationXtender connection .....	47
3.3.7	Creating a Content Server connection .....	48
3.3.8	Creating a SnapMatch connection .....	49
3.3.8.1	Automatically Refreshing a SnapMatch Connection .....	50
3.3.9	Copying connections .....	51
3.4	Defining styles .....	52
3.4.1	Customizing system styles .....	52
3.4.2	Adding custom styles .....	53
3.5	Defining shortcuts .....	55
3.5.1	Customizable module shortcuts .....	55

3.5.2	Reserved module shortcuts .....	62
3.6	Defining global options .....	63
3.6.1	Batch level names .....	63
3.6.2	Annotation colors .....	64
3.6.3	Image zone highlight colors .....	64
3.6.4	Panel background color .....	65
3.6.5	Thumbnail size .....	65
3.6.6	CTRL key behavior in image view .....	66
3.6.7	Non-system files for deployment .....	66
3.6.8	Recognition project path .....	67
3.7	Defining queries .....	68
3.7.1	Adding a query .....	68
3.7.1.1	ODBC query statement .....	71
3.7.1.2	DQL query statement .....	71
3.7.1.3	Text file query statement .....	72
3.7.2	Renaming a query .....	73
3.7.3	Testing queries .....	73
3.7.4	Query text file format .....	74
<b>4</b>	<b>Designing a CaptureFlow .....</b>	<b>75</b>
4.1	Getting started .....	75
4.1.1	What is CaptureFlow Designer? .....	75
4.1.1.1	Features .....	75
4.1.1.2	Opening CaptureFlow Designer .....	76
4.1.1.3	User interface .....	76
4.1.1.4	CaptureFlow Designer versus Process Developer .....	82
4.1.1.5	CaptureFlow Designer versus Intelligent Capture Administrator .....	84
4.1.1.6	Multi-language support .....	85
4.1.2	What is an XPP file? .....	85
4.1.3	High-level steps to a production CaptureFlow .....	87
4.1.4	Quick start: creating a simple process .....	89
4.2	Managing CaptureFlows .....	91
4.2.1	Creating a CaptureFlow .....	92
4.2.2	Opening a CaptureFlow .....	92
4.2.3	Opening a non-versioned CaptureFlow .....	93
4.2.4	Deleting a CaptureFlow .....	95
4.3	Designing a CaptureFlow chart .....	96
4.3.1	What is a CaptureFlow chart? .....	96
4.3.2	High-level steps for creating a CaptureFlow chart .....	98
4.3.3	Clipboard Operations .....	100
4.3.4	Adding batch creating steps .....	102
4.3.5	Adding batch processing steps .....	104

---

4.3.6	Renaming a step .....	107
4.3.7	Assigning departments to a step .....	107
4.3.8	Adding a decision block .....	109
4.3.9	Prioritizing tasks for a step .....	110
4.3.10	Adding a jump .....	111
4.3.11	Adding synchronization .....	112
4.3.12	Configuring data transfer between steps .....	112
4.3.12.1	Adding value mappings on the chart .....	113
4.3.12.2	Automatic value mappings .....	115
4.3.12.3	Special use cases .....	116
4.3.12.4	Transferring document data .....	116
4.3.12.5	Copying document fields to IA values .....	117
4.3.12.6	Copying IA values to document fields .....	119
4.3.13	Configuring the end of a CaptureFlow .....	120
4.3.14	Saving a CaptureFlow .....	121
4.3.15	Saving a CaptureFlow with a different name .....	121
4.3.16	Printing a CaptureFlow chart .....	122
4.4	Defining a custom value for a CaptureFlow .....	122
4.4.1	Creating custom values .....	122
4.4.2	Copying values between CaptureFlows .....	124
4.4.3	Using Clipboard operations .....	125
4.5	Developing scripting .....	126
4.5.1	Understanding CaptureFlow scripting .....	126
4.5.1.1	Supported scripting languages .....	127
4.5.1.2	Event handlers .....	127
4.5.1.3	Scope of operations .....	129
4.5.1.4	Code execution order for step initialization .....	129
4.5.1.5	Code execution order for a step in error .....	130
4.5.2	Script editor .....	131
4.5.2.1	User interface .....	131
4.5.2.2	Clipboard operations .....	135
4.5.3	Adding scripting .....	136
4.5.3.1	Adding scripting in the script editor .....	136
4.5.3.2	Adding step scripting in the chart .....	137
4.5.4	Deleting scripting .....	138
4.5.4.1	Deleting scripting in the script editor .....	138
4.5.4.2	Deleting step scripting in the chart .....	139
4.5.5	Opening scripting .....	139
4.5.6	Editing scripting .....	140
4.5.6.1	Inserting script methods .....	140
4.5.6.2	Shortcuts for editing code .....	140
4.5.7	Saving scripting .....	141

4.5.8	Building scripting .....	142
4.5.9	Debugging scripting .....	142
4.6	Monitoring warnings and errors .....	144
4.7	Searching for text .....	144
<b>5</b>	<b>Designing file and email import .....</b>	<b>147</b>
5.1	High-level steps .....	147
5.2	Email import profile .....	148
5.2.1	Creating an email import profile .....	148
5.2.1.1	Email profile options .....	149
5.2.1.2	Polling rule variables .....	160
5.3	File system import profile .....	161
5.3.1	Creating a file system profile .....	161
5.3.1.1	File system profile options .....	162
5.3.1.2	Variables for polling rules .....	172
5.3.1.3	Variables for batch name schema .....	173
5.4	Deploying email and file system profiles to standard import module instances .....	174
<b>6</b>	<b>Designing a profile for the Web Client .....</b>	<b>175</b>
6.1	Creating a Distributed Capture import profile .....	177
6.2	Setting up batch processing .....	187
<b>7</b>	<b>Designing image processing .....</b>	<b>189</b>
7.1	Managing Image Processing profiles .....	190
7.2	Designing an image processing profile .....	191
7.2.1	Adding filters .....	195
7.2.2	Reordering filters .....	195
7.2.3	Removing a filter .....	196
7.3	Annotating images .....	196
7.3.1	Adding IA values to the annotations .....	198
7.4	Evaluating image processing quality with test images .....	199
7.5	Image processing filters usage example .....	200
7.6	Image filters reference .....	200
7.6.1	Detection filters .....	201
7.6.1.1	Detect barcodes .....	201
7.6.1.1.1	Supported barcode types .....	202
7.6.1.2	Detect patch codes .....	205
7.6.1.3	Detect blank pages .....	206
7.6.1.4	Detect color marks .....	207
7.6.1.5	Detect colorfulness .....	207
7.6.2	Removal filters .....	208
7.6.2.1	Remove background .....	208

7.6.2.2	Remove black bars .....	209
7.6.2.3	Remove holes .....	209
7.6.2.4	Remove lines .....	210
7.6.3	Color adjustment filters .....	211
7.6.3.1	Adjust overall color .....	212
7.6.3.2	Convert specific color .....	213
7.6.3.3	Convert to black white .....	213
7.6.3.4	Convert to black white advanced .....	214
7.6.3.5	Invert black white .....	215
7.6.4	Image quality filters .....	216
7.6.4.1	Adjust lighting .....	216
7.6.4.2	Adjust thickness .....	216
7.6.4.3	Remove specks .....	217
7.6.4.4	Smooth edges .....	217
7.6.5	Page correction filters .....	218
7.6.5.1	Crop .....	218
7.6.5.2	Deskew .....	221
7.6.5.3	Rotate .....	223
7.6.5.4	Scale .....	224
7.6.6	Third-party filters .....	225
7.6.6.1	Convert to black and white (adaptive) .....	225
7.6.6.2	Line removal .....	225
7.6.6.3	Set DPI .....	227
7.6.6.4	Check Preprocessor - U.S. ....	228
<b>8</b>	<b>Designing image conversion .....</b>	<b>229</b>
8.1	Image Conversion profiles .....	229
8.1.1	What is image conversion? .....	229
8.1.2	How does image conversion work? .....	230
8.1.3	High-level steps .....	230
8.2	Image Conversion profiles editor .....	232
8.3	Managing Image Conversion profiles .....	246
8.3.1	Image Conversion profiles list .....	246
8.3.2	Adding a new profile .....	248
8.3.3	Testing profiles .....	248
8.4	Image Conversion profiles usage examples .....	250
8.4.1	Converting Microsoft Office multi-page documents into images .....	250
8.4.2	Scanned images merged into a multi-page PDF document .....	251
8.4.3	Converting Microsoft Office multi-page documents into single-page PDF documents .....	252
8.4.4	Converting Microsoft Office multi-page documents into multi-page PDF documents .....	253
8.4.5	Converting emails into multi-page PDF documents .....	255

8.4.6	Processing PDF attachments .....	256
<b>9</b>	<b>Designing a profile for optical character recognition processing .....</b>	<b>259</b>
9.1	What is Standard OCR? .....	259
9.2	How does Standard OCR functionality work? .....	260
9.3	High-level steps .....	260
9.4	Managing Standard OCR profiles .....	262
9.4.1	Standard OCR profiles list .....	262
9.4.2	Adding a new profile .....	263
9.5	Standard OCR profile editor .....	264
9.5.1	Testing a Standard OCR profile .....	275
<b>10</b>	<b>Designing a profile for exporting images and data .....</b>	<b>277</b>
10.1	High-level procedure .....	277
10.2	Listing all, creating, opening, and deleting export profiles .....	277
10.3	Defining an export profile .....	278
10.3.1	Using autocompletion .....	279
10.3.2	Using the embedded content editor .....	281
10.4	Setting profile properties .....	283
10.5	Defining filters .....	283
10.6	Defining export commands .....	285
10.6.1	Defining ApplicationXtender content .....	287
10.6.1.1	AX Export configuration: ApplicationXtender mapping options .....	288
10.6.1.2	Setting file image options .....	289
10.6.1.3	Setting Advanced options .....	290
10.6.2	Configuring CMIS content .....	292
10.6.2.1	Export options .....	293
10.6.2.2	Setting Advanced options .....	294
10.6.3	Defining Content Server content .....	296
10.6.3.1	Creating an export list .....	297
10.6.3.2	Configuring item definitions .....	298
10.6.3.2.1	Specifying an item location .....	298
10.6.3.2.2	Specifying categories and attributes .....	300
10.6.3.2.3	Specifying file options .....	301
10.6.3.2.4	Specifying permissions .....	302
10.6.3.2.5	Specifying a Workflow Map and Attributes .....	303
10.6.3.3	Setting Advanced options .....	304
10.6.4	Defining CSV content .....	306
10.6.5	Defining xECM content .....	308
10.6.6	Defining email content .....	312
10.6.6.1	Defining the email header .....	312
10.6.6.2	Defining attachments .....	314

10.6.6.3	Defining the email body .....	315
10.6.6.4	Setting advanced options .....	316
10.6.7	Defining file export .....	317
10.6.8	Defining ODBC content .....	319
10.6.8.1	Related Topics .....	322
10.6.9	Defining text content .....	322
10.6.9.1	Defining text file structure .....	322
10.6.9.2	Setting advanced options .....	324
10.6.10	Defining XML content .....	325
10.6.10.1	Defining the XML file structure .....	326
10.6.10.2	Defining XSL transformation rules .....	327
10.6.10.3	Setting advanced options .....	327
<b>11</b>	<b>Designing a document classification or data extraction project .....</b>	<b>329</b>
11.1	Information extraction projects .....	329
11.1.1	High-level procedure—Information extraction projects .....	329
11.1.2	Creating information extraction profiles .....	330
11.1.3	Converting Advanced Recognition document types to Information Extraction document types .....	333
11.2	Recognition projects .....	334
11.2.1	What is a recognition project? .....	334
11.2.2	High-level steps to use a project in production .....	335
11.2.3	Managing recognition projects .....	336
11.2.3.1	Recognition projects list .....	336
11.2.3.2	Adding a new recognition project .....	337
11.2.3.3	Opening a recognition project .....	338
11.2.3.4	Importing a recognition project .....	338
11.2.3.5	Deleting a recognition project .....	340
<b>12</b>	<b>Designing a document type .....</b>	<b>341</b>
12.1	Document types and Intelligent Capture runtime components .....	341
12.1.1	Which Intelligent Capture runtime components use document types? .....	341
12.1.2	Identification restrictions .....	343
12.1.3	Web Client restrictions .....	343
12.2	High-level procedure .....	345
12.3	Defining fields .....	347
12.4	Managing fields, controls, segments, and tables .....	348
12.5	Designing the layout of the form .....	350
12.6	Creating population and validation rules .....	353
12.6.1	Identification: order of triggering .....	354
12.6.2	Extraction: order of triggering .....	354

12.6.3	Completion: order of triggering .....	355
12.6.4	Creating population rules .....	356
12.6.4.1	Creating SnapMatch definitions .....	360
12.6.5	Creating validation rules .....	361
12.6.6	Using expressions in population and validation rules .....	364
12.6.6.1	Details .....	365
12.6.6.2	ArraySum function .....	366
12.6.7	Creating queries .....	367
12.6.8	Example: Using population rules .....	369
12.6.9	Example: Using validation rules .....	374
12.7	Maintaining integrity between advanced recognition projects and document types .....	380
12.8	Document properties reference .....	381
12.9	Field, control, segment, and table reference .....	385
12.9.1	Text Box (field) .....	385
12.9.1.1	Using masking .....	399
12.9.1.2	Standard date/time format conversions for Web Client .....	401
12.9.2	Drop-down List Box (field) .....	402
12.9.3	List Box (field) .....	409
12.9.4	Combo Box (field) .....	414
12.9.5	Check Box (field) .....	421
12.9.6	Text Block (control) .....	425
12.9.7	Group Box (control) .....	426
12.9.8	Button (control) .....	428
12.9.9	Image (control) .....	430
12.9.10	Table .....	431
<b>13</b>	<b>Deploying a component .....</b>	<b>433</b>
13.1	How to use deployment .....	433
13.2	Understanding Deployment tab elements .....	434
13.3	Uploading or deploying service components including profiles and CaptureFlows .....	439
13.4	Filtering service components list .....	440
13.5	Deleting service components .....	441
<b>14</b>	<b>Testing and debugging a CaptureFlow .....</b>	<b>443</b>
14.1	Compiling a CaptureFlow .....	443
14.2	Installing a CaptureFlow .....	443
14.3	Setting up CaptureFlow steps .....	448
14.3.1	Setting up a step on a different machine .....	450
14.4	Editing an installed CaptureFlow .....	450
14.5	Deleting a CaptureFlow instance .....	452
14.6	Uploading a CaptureFlow .....	453

---

14.7	Downloading components from the capture server .....	455
<b>15</b>	<b>Programming Reference—Expressions .....</b>	<b>457</b>
15.1	Data paths .....	459
15.2	Operators .....	459
15.3	Data types .....	460
15.4	Functions .....	463
15.4.1	Date functions .....	463
15.4.1.1	DateAdd .....	463
15.4.1.2	DateDiff .....	463
15.4.1.3	FormatDate .....	464
15.4.1.4	GetDatePart .....	466
15.4.1.5	Now .....	466
15.4.1.6	ResetDate .....	466
15.4.1.7	ResetTime .....	467
15.4.2	Get/Set functions .....	467
15.4.2.1	GetDataProperty .....	467
15.4.3	Numeric functions .....	467
15.4.3.1	Abs .....	467
15.4.3.2	Ceiling .....	468
15.4.3.3	Floor .....	468
15.4.3.4	FormatNumber .....	468
15.4.3.5	Random .....	469
15.4.3.6	ToNumber .....	470
15.4.4	String functions .....	470
15.4.4.1	Capitalize .....	470
15.4.4.2	ApproximateMatch .....	470
15.4.4.3	CompareStrings .....	471
15.4.4.4	CreateGUID .....	471
15.4.4.5	IndexOf .....	472
15.4.4.6	Length .....	472
15.4.4.7	MatchAny .....	472
15.4.4.8	RegexMatch .....	473
15.4.4.9	RegexReplace .....	473
15.4.4.10	Basic string manipulations with RegexReplace .....	473
15.4.4.11	String manipulations with RegexReplace .....	475
15.4.4.12	RemoveCharacters .....	476
15.4.4.13	RemoveWhiteSpace .....	476
15.4.4.14	Replace .....	477
15.4.4.15	ReplaceCharacters .....	477
15.4.4.16	Select .....	477
15.4.4.17	Substring .....	478

15.4.4.18	ToLower .....	478
15.4.4.19	ToUpper .....	478
15.4.4.20	Trim .....	479
15.5	Data formatting .....	479
15.5.1	Format string syntax .....	479
15.5.2	Custom number format examples .....	480
15.5.3	Custom date and time format examples .....	480
<b>16</b>	<b>Reference .....</b>	<b>481</b>
16.1	Naming Conventions .....	481
16.1.1	Reserved IA values .....	487
16.2	IA Values .....	490
16.2.1	_Batch IA Values .....	490
16.2.2	_Node IA Values .....	491
16.2.3	Custom values .....	492
16.2.4	Dynamic IA Values (\$Runtime) .....	492
16.2.5	ErrorCode IA Value .....	493
16.2.6	Step IA Values .....	494
16.2.7	Tree IA Values .....	494
16.2.8	UimData IA Value .....	495
16.2.9	PreUimDataImportMode IA Value .....	495
16.3	Capture System Files and Folders .....	496
16.4	Sample XPP Files .....	498
16.4.1	AdvancedMultiDocCapture Process .....	498
16.4.2	AutoIndexing Process .....	500
16.4.3	BasicMultiDocCapture Process .....	502
16.4.4	BasicRescan Process .....	504
16.4.5	eDocumentAndImageCapture Process .....	506
16.4.6	eDocumentCapture Process .....	511
16.4.7	ExpenseCollate Process .....	513
16.4.8	WSInputRescan Process .....	516
16.4.9	WSOutputScan Process .....	518
16.5	Deployment Utility .....	520
<b>GLS</b>	<b>Glossary .....</b>	<b>525</b>

## Chapter 1

# Getting started

Intelligent Capture Designer is a centralized design tool for creating, configuring, deploying, and end-to-end testing of a capture system.

---

### Design tools

Intelligent Capture Designer (referred to as *Designer* in this documentation) provides a single point of setup for process design tasks and enables access to various design tools:

- Reusable configuration profiles that you can apply across capture processes and assign dynamically per task.
- Configurable image processing filters for images quality enhancement and preparation for recognition step.
- Export to XML, text, and email with flexible expressions-based data assembly mechanism to filter and organize data into the required format.
- Deployment environment for isolating the capture server design and customer projects from environmental factors.
- Integrated scripting and debugging environment letting you concentrate on your profiles design tasks without switching to other Intelligent Capture modules.

---

### Design areas

To provide features mentioned above, Designer unites a number of design areas, namely:

- **Image Processing:** create profiles intended for improving images quality and optionally adding annotations (text or visuals) to images. For details, see section [“Designing image processing” on page 189](#).
- **Image Conversion:** create profiles for transforming files from one type to another, for example, converting non-image files to images, creating output files with specific formats and merging and splitting documents and annotations added to *TIFF* images by other modules. For details, see section [“Image Conversion profiles” on page 229](#).
- **Standard OCR:** create profiles for extracting data from electronic documents and images, converting output files to *PDF* or text format, and producing OCR data cache as a result of processing. For details, see section [“Designing a profile for optical character recognition processing” on page 259](#).
- **Recognition:** allows you to edit *DPP* projects to be later used with Completion, Extraction, and other modules. For details, see section [“Recognition projects” on page 334](#).

- **Document Types:** design data entry forms for indexing and validation performed by the Completion module and Intelligent Capture Web Client operators. For details, see section [“Designing a document type” on page 341](#).



**Note:** Hereafter, Intelligent Capture Web Client is referred to as Web Client (unless more clarity is required).

- **Import:** imports image files from directories, email and attachments from an email server, and files and batch node values from the Web Client as batches. For details, see [“Designing file and email import” on page 147](#).
- **Export:** create profiles intended for extracting data from batch nodes and converting to a number of formats. For details, see section [“Designing a profile for exporting images and data” on page 277](#).
- **CaptureFlow Process Designer:** use a graphical environment intended for capture process design letting you create a model of the process and access to various process design tools.

---

#### Efficiency for distributed work

In addition to having a single place for setting up and managing everything related to capture system configuration, Designer allows multiple developers to work in the same environment and profiles simultaneously as it enables work with different parts of the system in isolation.

---

## 1.1 Features

After you install Intelligent Capture components and launch Designer, the **Overview** tab opens. This tab provides easy access to let you perform the major tasks of capture system design:

1. Set up the working capture system including capture servers for development, test, and production purposes.
2. Create and test profiles. A profile is a configuration file that corresponds to a client module.
3. Set up CaptureFlows, deploy them to the capture server, and debug your scripts if required.

[“Designer features” on page 15](#) describes the most common tasks and Designer features to use.

**Table 1-1: Designer features**

Task	Designer element to use
1. Set up a capture system and capture server	<p>The <b>Systems &gt; Overview</b> tab provides a description of the capture system setup and quick navigation to various workflow setup steps.</p> <p>The <b>System Configuration &gt; Capture System</b> allows you to create a workspace for storing a set of CaptureFlows, profiles, document types, and other entities, which are generally called service components that make up an end-to-end working system.</p> <p>The <b>System Configuration &gt; Capture Server</b> enables setup of servers to be used during different stages of lifecycle, such as development, test, and production.</p> <p>The <b>Systems &gt; Deployment</b> tab is used to push any service components created or modified using Designer to a previously created server. For details, see <a href="#">Capture System</a>.</p>

Task	Designer element to use
<p>2. Prepare capture profiles</p>	<p>Profiles in Designer represent configurations that are not specific to any process. You design profiles using available elements in the <b>Area Navigation</b>:</p> <ul style="list-style-type: none"> <li>• Use <b>Import</b> for importing image files from directories, email and attachments from an email server, and files and batch node values from the Web Client as batches. For more information, see <a href="#">“Designing file and email import” on page 147</a>.</li> <li>• Use <b>Distributed Capture</b> for scanning paper documents and importing document image files. For more information, see <a href="#">“Designing a profile for the Web Client” on page 175</a>.</li> <li>• Use <b>Image Processing</b> for enhancing images, detecting their properties, and adding annotations to <i>TIFF</i> images. For more information, see <a href="#">“Designing image processing” on page 189</a>.</li> <li>• Use <b>Image Conversion</b> to create image files from different input formats and convert them to a number of output formats. For more information, see <a href="#">“Designing image conversion” on page 229</a>.</li> <li>• Use <b>Standard OCR</b> for extracting data from different types of electronic documents and images and generating OCR data cache. For more information, see <a href="#">“Designing a profile for optical character recognition processing” on page 259</a>.</li> <li>• Create <b>Information extraction and Recognition</b> projects for setting up classification and data extraction properties of information extraction and recognition projects. For more information, see <a href="#">“Designing a document classification or data extraction project” on page 329</a>.</li> <li>• Use <b>Document Types</b> for defining indexing fields and tables. For more information, see <a href="#">“Designing a document type” on page 341</a>.</li> <li>• Use <b>Export</b> to set up data export rules for exporting data into standard formats or emails. For more information, see</li> </ul>

Task	Designer element to use
	<a href="#">“Designing a profile for exporting images and data” on page 277.</a>
3. Define CaptureFlows	Use <b>CaptureFlow</b> to specify how the batches are created and tasks are performed using attended and unattended modules. For more information, see <a href="#">“Designing a CaptureFlow” on page 75.</a>
4. Upload service components to capture server	In Designer, deployment is a method allowing to push all locally modified service components to a previously setup capture server.  Use <b>System &gt; Deployment</b> tab to upload the profiles you have created in Designer to the capture server once you have created a test batch. For more information, see <a href="#">“Deploying a component” on page 433.</a>

### Related Topics

[“Preparing a capture system” on page 29](#)

[“Designing a CaptureFlow” on page 75](#)

[“Deploying a component” on page 433](#)

## 1.2 High-level steps







Designer has several major usage examples:

- Create and deploy content to a new Intelligent Capture installation.
- Create and deploy content to existing Intelligent Capture installation on a new developer's machine.
- Move the design processes from the test machine to a production environment.
- Import externally created content.

For more information, see [“Transferring document data” on page 116.](#)

### To set up high-level steps in Designer:


1. Set up a capture system environment, including system-wide configuration options.
2. To create profiles that specify processing options for images and define the user interface that operators view, click the corresponding toolbar button and click **New** on the tab that lists the profiles. The following table displays the corresponding toolbar button.

button	Module	For information, see
 <b>Import</b>	Standard Import	In Designer, see <a href="#">Designing file and email import</a> . For documentation on Standard Import, see <i>OpenText Intelligent Capture - General Import/Export Modules Guide (ECPCORE-CIO)</i> .
 <b>Image Processing</b>	Distributed Capture	In Designer, see <a href="#">Designing a profile for the Web Client</a> .
 <b>Image Conversion</b>	Image Processor	In Designer, see <a href="#">Designing image processing</a> . For documentation on Image Processor, see <i>OpenText Intelligent Capture - Image Handling Modules Guide (ECPCORE-CIH)</i> .
 <b>Standard OCR</b>	Image Converter	In Designer, see <a href="#">Designing image conversion</a> . For documentation on Image Converter, see <i>OpenText Intelligent Capture - Image Handling Modules Guide (ECPCORE-CIH)</i> .
 <b>Export</b>	Standard OCR	<a href="#">Designing a profile for optical character recognition processing</a> For documentation on Standard OCR, see <i>OpenText Intelligent Capture - Recognition and Advanced Recognition Modules Guide (ECPCORE-CMR)</i> .
 <b>Export</b>	Standard Export	In Designer, see <a href="#">Designing a profile for exporting images and data</a> . For documentation on Standard Export, see <i>OpenText Intelligent Capture - General Import/Export Modules Guide (ECPCORE-CIO)</i> .

When designing a capture system, you can create multiple profiles of the same type intended for different business needs. Once uploaded on the server, a profile can be applied to an appropriate CaptureFlow step in one of the following ways:

- A profile can be assigned to a step during setup.
- A profile can be assigned in the module's IA value at design time, in which case this assigned value has a priority over the setup setting.
- A profile can be assigned to the module's IA value dynamically at runtime. This can be done in scripting. As in the previous case, a profile assigned to the IA value overrides the setup setting.

The same profile can be applied to multiple steps in the same CaptureFlow as well as shared by multiple CaptureFlows.

3.  To perform document classification and data extraction, click **Recognition** and then select either **Advanced Recognition** or **Information Extraction** for **Project Type**:
  - For **Advanced Recognition**, you use Recognition Designer. In Recognition Designer, the document type is displayed as an index family in the recognition project that you have associated with the document type. Therefore, the document type fields correspond to the index fields in the index family (which are mapped to templates). To complete the design of the classification templates, you map the index fields to the individual pieces of data by visually positioning the fields on the classification templates and setting classification properties. For more information, see [“Recognition projects” on page 334](#).
  - For **Information Extraction**, you use the **Information Extraction Project Editor**. For more information, see [“Information extraction projects” on page 329](#).
4. Create one or more CaptureFlows that specify how batches are created and processed using attended and unattended steps.
5. Install an instance of the process on the test capture server using the designed CaptureFlow.
6. Set up steps of the installed process.
7. Create test batches based on the process installed on the server and use the production modules to process data in the batches.
8. Deploy service components, including the profiles, to the production capture servers.

## Related Topics

[“Deploying a component” on page 433](#)

## 1.3 User interface

This section describes the Designer user interface available when you launch it for the first time to create your first capture system and optionally copy samples.

### 1.3.1 Top banner


The **Top banner** has several commands common for all Designer components and provides information on the **Connection** status and the **main toolbar**.


The **main toolbar** lets you perform a number of actions on the current screen in the main work area, namely:

- Connect or disconnect from a selected capture server and upload locally modified profiles or settings.
- Undo and redo previously used commands, perform standard edits.
- Save any modifications to profiles or capture system settings.
- Access context help for Designer.

“[Top Banner Elements](#)” on page 20 provides a detailed description of the **Top banner** elements.

**Table 1-2: Top Banner Elements**

Element	Description
Target Capture System link	<p>The status of the currently selected capture system and connection to the capture server you are currently using. The following connection states are available:</p> <ul style="list-style-type: none"> <li>• <b>Connected</b> (the link text displays green)</li> <li>• <b>Disconnected</b> (the link text displays red)</li> </ul> <p> <b>Note:</b> If a connection to the capture server is not established, you cannot deploy any modified service components.</p> <p>For more information about capture systems, see “<a href="#">Preparing a capture system</a>” on page 29.</p>
Connect toggle button	<p>Connects or disconnects to the currently selected capture server.</p> <p>For more information about capture systems, see “<a href="#">Preparing a capture system</a>” on page 29.</p>

Element	Description
<b>Upload Local Changed</b> button	<p>Enables uploading all modified service components from the local machine to the selected capture server.</p> <p> <b>Note:</b> The button is disabled if there is no connection to the capture server.</p> <p>For more information about deploying service components, see <a href="#">“Deploying a component” on page 433</a>.</p>
<b>Edit</b> drop-down menu	<p>Lets you use the drop-down menu items to edit the currently selected profile or service component:</p> <ul style="list-style-type: none"> <li>• <b>Undo (CTRL+Z)</b></li> <li>• <b>Redo (CTRL+Y)</b></li> <li>• <b>Cut (CTRL+X)</b></li> <li>• <b>Copy (CTRL+C)</b></li> <li>• <b>Paste (CTRL+V)</b></li> <li>• <b>Delete (DELETE)</b></li> </ul>
<b>Save</b> button	<p>Click the main <b>Save</b> button (or press <b>CTRL+S</b>) to save all recently made changes or select one of the following commands from the drop-down menu:</p> <ul style="list-style-type: none"> <li>• <b>Save:</b> Saves changes in the currently opened profile.</li> <li>• <b>Save As:</b> Opens a <b>Save As</b> dialog for the currently opened profile or CaptureFlow.</li> <li>• <b>Save All:</b> Saves modified profiles in all profile editors and capture system settings.</li> </ul>
<b>Help</b> button	<p>Click the main <b>Help</b> button to view the context help and other documentation resources.</p>

### 1.3.2 Area navigation

The **Area Navigation**, as its name suggests, helps you navigate to different design tools, usually showing a first page/tab or previously selected page of the selected design tool:

- **System:** A landing page letting you create a capture system that contains all of the service components and settings required for correct deployment to the test and production servers.
- **Import:** Specify how to import image files from directories, email and attachments from an email server, and files and batch node values from the Web Client as batches.

- **Distributed Capture:** Specify options for scanning paper documents and importing document image files.
- **Image Processing:** Analyze, enhance, and transform images, and add annotations.
- **Image Conversion:** Create files in the required format.
- **Standard OCR** Specify options for extracting data from different types of electronic documents and images and generating OCR data cache.
- **Recognition:** Create recognition projects to define how the images can be classified automatically and the data is extracted.
- **Document Types:** Create data entry forms to define the indexing fields to be used by capture users.
- **Export:** Specify how data and files in a particular batch can be sent to the back-end systems.
- **CaptureFlow:** Specify and test CaptureFlows, which define how the batches are created and how attended and unattended modules' steps are run.

When you select the component in the **Area Navigation** for the first time, you navigate to the component **Overview** page. If you have already worked with the selected component, the last page you opened when working with it opens automatically. Consider the following navigational options when using the **Area Navigation**:

- To navigate and switch between the items in the **Area Navigation**, use **CTRL+TAB** or **CTRL+SHIFT+TAB** to navigate in reverse direction.
- To navigate between the elements in the same panel, use the **TAB** key.
- To navigate between the controls on the selected panel, use the **arrow** keys.

### 1.3.3 Problems tab and status bar

The **Problems** tab and status bar provide information on the current state of the capture system, the profiles, and the deployment and testing status. When you switch between capture systems, the panel contents adjusts the current state of the selected system.

By default, the **Problems** tab is hidden and displays as soon as an error or warning appears.

[“Problems tab elements description” on page 23](#) provides a description of **Problems** tab elements.

**Table 1-3: Problems tab elements description**

Name	Description
Errors toggle button	Displays the number of errors in Designer.  Toggle the <b>Errors</b> icon to see/hide the errors from the <b>Problems Panel</b> list.
Warnings toggle button	Displays the number of warnings in Designer.  Toggle the <b>Warnings</b> icon to see/hide the warnings from the <b>Problems Panel</b> list.
Description column	Provides error or warning summary.
Area	Lets you navigate to the specific screen, panel, or field of the component where the warning/error has occurred.
Component	Displays the name of the component where the issue occurred.
Found In	Displays the name of the profile, specific connection, query, or other service component where an issue occurred.



**Note:** You can copy the information in the **Problems > Description**. To do so, select the desired message, click the right mouse button and select **Copy**. Then navigate to the desired text editor and paste the information.

The **Problems** status bar is an information area located at the bottom of Designer window. It shows the count as well as information on errors and warnings discovered and includes several sections, namely a list of issues found. To investigate the issue found, click the discovered issue link. To hide the **Problems** tab, click **Hide** in the **Problems** status bar. To display the **Problems** tab, click the issue link in the **Problems** status bar.

## 1.4 Navigating Designer

Each **Designer** component works independently except for navigation and **Top banner** commands. The following navigation tools are available:

- **Top Banner** with **Connections** area shows a list of capture servers (for example, development, test, and production) and connection information and a toolbar
- Left-hand **Area Navigation** with a list of available designers including the capture system designer
- Component-specific toolbar
- **Problems** panel

## 1.5 Understanding SnapMatch

SnapMatch allows you to extract information that is not presented on the pages (or might not be recognized on the page) by finding it in the prepared text file or from the database (from the generated SnapMatch data file). To do this, each document is matched against the file or database using the text existing on the page. When the correct record is found, the *UIM* document's fields are populated according to configured SnapMatch population rule mappings.

SnapMatch requires OCR data on input. It reads text and tries to find matches in rows using a local index. When successful, SnapMatch returns the requested value. As a result, the Extraction module populates UIM document's fields with values from the SnapMatch data file. If a match cannot be found, the UIM document's fields are not filled and stay unchanged.

In previous versions, users had to manually refresh the SnapMatch data connection. Starting in 22.1, SnapMatch data from the database can be automatically refreshed and synchronized with the database or source file. SnapMatch data is fetched only during setup; SnapMatch data will have up to date synchronization with the SnapMatch database or source file.

For information about using SnapMatch, see [“Creating a SnapMatch connection” on page 49](#).

## Chapter 2

# Configuring the design environment

## 2.1 Configuring the working directory

By default, Designer stores all designer created content in the local file system at `C:\Users\<username>\My Documents\<product_name_and_version>\Default\<Capture System name>\. Designer uses this working directory for capture system lookup.`

Designer can be configured to use a different working directory, either in the local file system or in the network. You can do it by updating the `Intelligent CaptureDesigner.exe.config` configuration file. A modified working directory will indicate the path where Designer will save new capture systems. Capture systems located in a previous working directory become invisible to Designer.

Choosing a network directory may be useful when multiple developers are working in the same capture system. However, Designer does not allow multiple developers work in the same capture system simultaneously.

### To change the default working directory:

1. On the Designer host machine, browse to `c:\<Program Files>\InputAccel\Client\binnt`.
2. Open the `CaptivaDesigner.exe.config` file using Notepad or any other text editor.
3. Scroll to the section `<applicationSettings>`. This section is commented out by default.
4. Set the `<value>` of the `DataPath` setting to one of the following:
  - A different location in the local file system . For example, `c:\system\`.
  - A network file share. For example, `\\<machine name>\system`.

If the `DataPath` setting is not specified or its value is empty, the capture system is created in the default location.

5. (Optional) Repeat the above steps on each Designer host from which the working directory will be accessed.
6. (Optional) If the network location is specified, ensure that all involved developers have been granted read-write access to the folder specified by the `DataPath` value.

## 2.2 Adding custom modules to the palette

The **Steps** panel of CaptureFlow Designer can be extended with custom modules. Once added to the **Steps** panel, a custom module can be dragged to the CaptureFlow chart and inserted as a step.

You also need to add your custom modules to CaptureFlow Designer manually after upgrade to Intelligent Capture 7 from Intelligent Capture 6.5 or 6.5 SP1/SP2.



### Caution

Before adding a module to CaptureFlow Designer, make sure that the module's MDF file has UTF-8 encoding and does not include IA values with non-English characters in their names.

#### To include a new custom module in CaptureFlow Designer:

1. Install the module files on the Designer host machine in the following folder:  
c:\Program Files\InputAccel\Client\binnt.
2. Open the UserModules.xml file for editing. This file is located on the Designer host machine in the following folder: c:\Program Files\InputAccel\Client\src\MDF\User.
3. For each new module, you need to insert the *Item XML* element inside the *Palette* element.

Example:

```
<Item Label="New Module" InternalName="newmodul" BatchCreator="true"
DefaultLevel="7" assemblytype="1" assembly="newmodul" DefaultStepName="NewModule"
icon="newmodule.ico" >
  <files>
    <file>NewModule.mdf</file>
    <file>NewModuleObjects.mdf</file>
  </files>
</Item>
```

The *Item XML* element has the attributes and nested elements provided in the following table.

**Table 2-1: UserModules.xml file: Item XML element structure**

XML item	Mandatory	Description
Label	Yes	This attribute specifies the name of the module that will be displayed on the <b>Steps</b> panel.
InternalName	Yes	This attribute specifies the internal name of the module. The internal name must not exceed 8 characters.

XML item	Mandatory	Description
assembly	Yes	This attribute specifies the assembly name (without extension) of a custom module.
assemblytype	Yes	This attribute specifies the type of module for use in setup mode. It takes the following values: <ul style="list-style-type: none"> <li>• 1: A pre-6.0 module that uses the pre-6.0 architecture</li> <li>• 2: A module that uses the QuickModule architecture</li> <li>• 3: A module that uses the DCC architecture</li> </ul>
BatchCreator	No	This attribute specifies whether the module creates a batch. It takes the following values: <ul style="list-style-type: none"> <li>• <b>False</b>: the module does not create a batch. It will be added to the <b>Steps</b> panel inside the <b>Custom</b> section.</li> <li>• <b>True</b>: the module creates a batch. You will be able to add it to your CaptureFlow chart by right-clicking on the <b>Start</b> element and selecting this module from the context menu.</li> </ul>
SupportedLevels	No	This attribute specifies a list of supported levels for the step. Add levels without spaces and separate them with a comma. Example: "0,1,2,3".  If this attribute is skipped, the step is considered to support all levels.
DefaultLevel	Yes	This attribute specifies the default level with which the step is added to the CaptureFlow chart.

XML item	Mandatory	Description
DefaultStepName	No	This attribute specifies the default name with which the step is added to the CaptureFlow chart.
icon	No	This attribute specifies the icon (<filename>.ico file) with which the module is displayed on <b>Steps</b> panel.  This attribute is optional if the module's internal name (InternalName) matches the ICO file name without extension. Otherwise, this attribute is required.
files	Yes	This element includes a collection of file XML elements.
file	Yes	This element specifies the name of the MDF file that declares IA values of the module.  If the module has several MDF files, you need to list all of them. The MDF file with IA values must be included at the first position. The MDF files that define objects, global values, and variants must be included after the main MDF.

4. Copy all of the module's MDF files to the same folder that stores the UserModules.xml file.



**Note:** If CaptureFlow Designer cannot find the module's MDF files in the same location as the UserModules.xml file, the module will not appear on the **Steps** panel.

5. (Optional) If you want your module be displayed on the **Steps** panel with a custom icon, add an ICO file to the same folder that stores the UserModules.xml file. The icon size must be 16 x 16 pixels or 32 x 32 pixels. You can use any name for the ICO file. To point the custom icon, use the icon XML attribute (above).

## Chapter 3

# Preparing a capture system

### 3.1 Defining capture systems

When designing a new capture process in Designer, you start with creating a *capture system*. A capture system is a solution that will encapsulate all components and configuration settings of your future capture process at design time.

#### Lifetime

A capture system serves for design purposes only. Once created, a capture system includes no purpose-based content, except for sample components and configuration settings (that are added optionally). In the file system, a capture system is presented by a predefined structure of empty **folders** that are intended for future content. When designing a capture process, you fill the capture system with various objects and configuration settings that will constitute the behavior of the capture process and its execution environment. At the end, you deploy the capture system components on the target server, after which the role of the capture system is done. All deployed components are stored on the server without any indication of a capture system to which they used to belong. The capture system can remain on the designer's machine for further updates, or it can be archived in the file storage and reused later.

#### Isolated design environment

Designer allows you to create multiple capture systems on the same machine and switch between them when developing several capture processes simultaneously. A capture system safely isolates its content from other capture systems. Any object created in a capture system can be viewed and edited in this capture system only, and deployed from within this capture system only. When developing solutions for independent sites or separate customers, capture system mechanism helps you avoid any potential intersections.

#### Multi-process design

When working on the task, you can design several capture processes within one capture system in parallel. All of these processes share the same components and settings, except for those strictly particular to a process. This approach allows you to try different implementations for the same process and choose the best candidate to be deployed with other components in the execution environment.

A capture system can define multiple execution environments for testing, demonstration, and production purposes. This option gives you flexibility to deploy a process in different purpose-specific environments and run it immediately, without updating its configuration settings.

## Teamwork in a capture system

You can place the capture system files in a shared location and this way support teamwork within the same capture system. However, multiple users cannot edit capture system components simultaneously. If any file in the capture system directory is opened for editing, the capture system is locked for other users.

### 3.1.1 Adding a new capture system

**To add a capture system:**

1. Select **System** and the **System Configuration** tab.
2. In the **Capture System** drop-down list, click **New Capture System**. The **New Capture System** dialog box opens.



**Note:** You can use the **Default** capture system as a template for creating a system. To do this, select the **Default** system in the **Capture System** drop-down list, click the **Edit** button, and make the changes you need.

3. In the **Capture System Name** field, specify the name for the capture system following the **naming convention**.
4. Select **Copy Samples** to create a capture system with **sample content**. The samples will include several CaptureFlows, profiles, queries, and connections.



**Note:** You can add samples only when creating a capture system.

5. Click the **Save** button.

The new capture system appears in the **Capture System** list selected as a current system.

### 3.1.2 Adding samples

Designer is installed with a collection of **sample XPP files**, profiles, recognition projects, document types, queries, and other configuration files. Sample files are installed on your desktop in the following directory by default:

`<drive>:\Program Files\InputAccelerator\Client\src\Sample Capture System`

To make the samples available for use in Designer, you need to create a capture system with the **Copy Samples** option enabled as described in **“Adding a new capture system” on page 30**. All samples are copied to the folder created specifically for that capture system: `<drive>:\Users\<User>\Documents\<product_name_ and_ version>\<capture system name>\GlobalData\`



**Note:** The subfolders included in the GlobalData folder are described in **“Capture System Files and Folders” on page 496**.

The copied samples are meant for demonstration purposes and are available for editing in Designer's embedded editors. For example, you can open an XPP sample in CaptureFlow Designer and use it as a model for your own CaptureFlow.

Sample files are installed with a collection of "read me" files (TXT). The `Readme First.txt` file contains an introduction to samples and an overview of all folders created for each capture system. Other TXT files contain the descriptions of the provided sample processes located in the `\GlobalData\XPP` folder. The "read me" files are located on your desktop in the following directory:

```
<drive>:\Users\<User>\Documents\<product_name_and_version>\<capture system name>\Readme\
```

### Related Topics:

["Adding a new capture system" on page 30](#)

["Capture System Files and Folders" on page 496](#)

["Sample XPP Files" on page 498](#)

## 3.1.3 Renaming a capture system

### To rename a capture system:

1. In Designer, select **System** and the **System Configuration** tab.
2. In the **Capture System** drop-down list, select the capture system you want to edit, and click the **Edit** button near the **Capture System** field.
3. In the **Edit Capture System** dialog box, specify the new capture system name in the **Capture System Name** field.
4. Click the **Save** button. The changes are applied to the capture system.



**Note:** After a capture system is renamed, it may get disconnected from the capture server.

## 3.1.4 Deleting a capture system

A capture system is deleted with all of its components, including CaptureFlows, profiles, queries, and connections. The capture system folder is deleted from the file system on the local machine.



### Caution

You cannot undo the deletion of a capture system.

### To delete a capture system:

1. Select **System** and then the **System Configuration** tab.

2. In the **Capture System** drop-down list, select the capture system you want to delete, and click the **Edit** button near the **Capture System** field.
3. In the **Edit Capture System** dialog box, click **Delete**. Confirm the deletion in the prompt dialog.

You cannot delete the **Default** capture system if it is the only system in the **Capture System** list. If you delete the **Default** capture system and then delete all remaining capture systems, the **Default** capture system will be automatically created.

## 3.2 Defining capture servers

A *capture server* is a required capture system component that points the real Intelligent Capture Server or a ScaleServer group where a process will be deployed and executed. A capture system is created with the **Default** capture server that you can rename and re-configure to meet your requirements. You can add multiple capture servers to the capture system, for instance, to use different execution environments when testing, debugging, and running the process in production.

### 3.2.1 Adding a capture server


**To add a capture server:**


1. In Designer, select **System** and select the **System Configuration** tab.
2. In the **Capture Server** drop-down list, select the **New Capture Server** link. The **New Capture Server** dialog opens.



**Note:** To use the **Default** capture server as a template for a new server, select **Default** the **Capture Server** drop-down list, click the **Edit** button, and make the necessary changes.

3. In the dialog fields, specify information for the new server. The following table describes the properties that must be specified when creating a capture server.

Property	Description
Alias	Specify the capture server name following the <b>naming convention</b> . This property is required.
Server	Specify the machine name of the server.   <b>Note:</b> The IP address may not always work correctly. It is recommended to use machine name to connect to the server.

Property	Description
<b>User</b>	Specify the user name registered on the server. The specified user must be granted permissions to access the capture server database.   <b>Note:</b> To debug a batch in CaptureFlow Designer, use the Windows user account credentials. To connect as a Windows user, specify * (asterisk) in the <b>User</b> field and leave the <b>Password</b> field blank.
<b>Password</b>	Specify the user password.
<b>Connect after save</b>	Optional. Select this check box if you want to establish the connection to this server immediately after saving the settings.

- Click the **Save** button.

The new capture server is added to the **Capture Server** list and is selected as a current server. The connection to the server is established if **Connect after save** was selected.

### 3.2.2 Editing a capture server

**To edit a capture server:**

- In Designer, select **System** and select the **System Configuration** tab.
- In the **Capture Server** drop-down list, select the capture server you want to edit, and click the **Edit** button near the **Capture Server** field.
- In the **Edit Capture Server** dialog, make the necessary changes.
- Click the **Save** button.

### 3.2.3 Deleting a capture server

A capture system must include at least one capture server. If you delete the only capture server from the capture system, Designer will create the **Default** capture server with default settings automatically.

Designer will not allow you to delete the **Default** capture server if the capture system defined no other servers.



#### **Caution**

The deletion of a capture server cannot be reverted using the undo command.

**To delete a capture server:**

1. In Designer, select **System** and select the **System Configuration** tab.
2. In the **Capture Server** drop-down list, select the capture server you want to delete, and click the **Edit** button near the **Capture Server** field.
3. In the **Edit Capture Server** dialog, click **Delete**.

The deletion of the server will be performed without any notifications.

### 3.2.4 Connecting to a capture server

To connect or disconnect to the selected capture server, use the **Connect** button located near the **Capture Server** drop-down list. As the connection status changes, the button changes its name between **Disconnected** and **Connected**.



**Note:** When connecting to a capture server, the dialog on updating a license can appear. To continue, update the license for the current server. For detailed information on managing licenses, see *OpenText Intelligent Capture - Administration Guide (ECPCORE-AON)*.

## 3.3 Defining connections

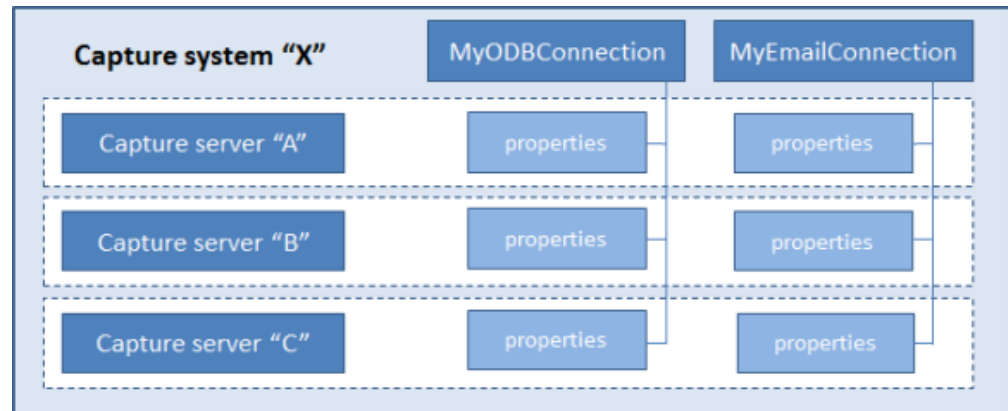
A connection is a capture system component that encapsulates all necessary information to let a capture server connect to the external data source. Designer allows you to create the following types of connections:

- **Database connection:** used to establish an *ODBC* connection to a particular database. Also, you can connect to a file database (such as Access and Excel files) to use it for testing purposes.
- **Documentum connection:** used to establish a connection to a particular OpenText Documentum Server repository (a doabase).
- **File system connection:** used to establish a connection to a particular text file located in the local file system or in the network.
- **Email connection:** used to establish a network connection to a mail server.
- **CMIS repository connection:** used to establish a connection to a CMIS-compliant repository.
- **ApplicationXtender connection:** used to establish a connection to an ApplicationXtender repository.
- **Content Server connection:** used to establish a connection to the Content Server.
- **SnapMatch connection** used to generate a SnapMatch data file from a text file or SQL Server database. The SnapMatch data file is stored locally and is required for a SnapMatch population rule.

Connections to databases, files, Documentum Server, CMIS-compliant and ApplicationXtender repositories are used by the **named queries** that are executed in

population and validation rules during validation. Connections to the Content Server and mail servers are used by **export**. The same connection can be reused in multiple queries or export profiles.

A connection can be deployed on any capture server available in the capture system. A connection is identified by a unique name and the connection type, and defines connection properties for each capture server available in the capture system.



**Figure 3-1: ODBC connection with servers**

When you create a connection and specify its properties, initially they apply to all capture servers available in the capture system. If necessary, these initial properties can be modified for each capture server individually. When you deploy a connection on the selected capture server, this connection is installed with the unique name and properties specific for that server (initial or modified ones).

### Automatic server to client synchronization


The same physical Intelligent Capture Server can be added as a component in several capture systems (on the same Designer host machine or in different ones). At design time, every connection installed on that server from any capture system is automatically synchronized with all other capture systems once they get connected to the server in Designer:

- If an installed connection is missing in the capture system, it is downloaded automatically and appears in the capture system with the properties specific for that server. If the capture system defines other servers, the downloaded connection properties apply to all these servers by default but can be edited for each server individually.
- If an installed connection is found in the capture system, it is not downloaded, even if the locally stored connection properties are not identical to the server-side properties.

Automatic synchronization works in the "from server to client" direction only. Connections are never uploaded to a capture server automatically.


### 3.3.1 Creating an ODBC connection

A database connection describes an ODBC connection of the capture system to a particular data source. Database connections can be used to access ODBC-compliant database systems as well as Access and Excel files and structured text files.


 **Note:** Connecting to a file database (such as Access and Excel files) can be used for testing purposes only.

A database connection is a required component in an ODBC query. If your CaptureFlow populates document fields from a database or validates extracted values using data from a database, you need to create a database connection to access this external data. Then you can reuse this connection in as many ODBC queries as needed.


You can create and edit database connections in the **System** tool of Designer. The **Database Connection** list (**System** > **System Configuration** tab > **Configuration Settings** list > **Database Connection** option) displays existing ODBC connections.

 **Note:** The **Database Connection** list may also include sample connections for testing purposes. Sample connections are only available if you create a capture system with the **Copy Samples** option enabled. For more detail, see topic [“Adding a new capture system” on page 30](#).

#### To add a new database connection:

1. In the **System** tool of Designer, open the **Database Connection** list:  
Click **System** > **System Configuration** tab > **Configuration Settings** list > **Database Connection** option.  
The existing ODBC connections are displayed.  
 **Note:** The IE Repository ODBC connection is mandatory for information extraction and cannot be deleted. The values are set by the installer, but you can change the settings by updating the connection.
2. Select a capture server and connect to it. When disconnected, you still can create connections, but you cannot specify connection parameters and test connections.
3. Click **<New Connection>** in the **Database Connection** list and specify the name of the new connection following the [naming convention on page 484](#). The database connection cannot be renamed later.
4. Define the database connection parameters as stated in the following table.

**Table 3-1: Database connection parameters**

Parameter	Description
<b>Server</b>	Specify the name or the <i>IP</i> address of the database server. The property is required.  To connect to a file database (Access or Excel file), enter a wildcard (*) character.
<b>Database</b>	Specify the database name. The property is required.  To connect to a file database, enter a wildcard (*).
<b>Driver</b>	Specify the name for the ODBC driver that must be installed on the capture server. The property is required. The value in this property is not validated.  For example, specify "SQL Server Native Client 10.0" to connect to a database or {Microsoft Excel Driver (*.xls)} to connect to an Excel spreadsheet file.  The ODBC driver must also be installed on the client machines where the Completion, Identification, and Extraction modules run the corresponding ODBC query.   <b>Note:</b> For the full list of supported ODBC drivers, see <i>Intelligent Capture Release Notes</i> (available in My Support ( <a href="https://support.opentext.com/">https://support.opentext.com/</a> )). For drivers that are not officially supported, find instructions for the .NET ODBC framework on the website of the <i>ODBC</i> driver manufacturer.
<b>Port</b>	(Optional) Specify the server port. Range: 1 to 65535.  Leave 0 to use the default port. You can use 1433 for SQL Server and 1521 for Oracle.
<b>AdditionalParameters</b>	(Optional) Specify the ODBC connection string in the format required by the ODBC driver. The value in this property is not validated.  For example, Microsoft Excel Driver requires a connection string formatted as <code>DBQ=&lt;file_path&gt;\&lt;file_name&gt;.xls</code>  For detailed information, see the ODBC driver documentation.

Parameter	Description
<b>ConnectionRetryCount</b>	Specify the number of connection retries to perform if the database connection fails.
<b>ConnectionTimeoutSec</b>	Specify the database connection time-out (in seconds).
<b>User</b>	Specify the user name registered in the database and granted access permissions. The property is required.  To connect to a file database with the current Windows user credentials, enter a wildcard (*) character.
<b>Password</b>	Specify the user password. To connect to a file database, leave this property blank.

5. Test the connection by clicking the **Test** button.  
Designer attempts to connect to the database using the specified connection parameters. If successful, Designer marks the connection with the green check icon in the **Database Connection** list. A failed connection is marked with a red cross icon, and the issue description appears in the **Problem** panel.
6. (Optional) Specify the above connection properties for other capture servers available in the capture system. By default, the connection holds the same set of properties for all capture servers and, when installed on any of these servers, points the same data source.

### 3.3.2 Creating a Documentum connection

A Documentum connection encapsulates all necessary information to let a capture server connect to a particular OpenText Documentum Server repository (a doabase). In production, this communication typically takes place when a CaptureFlow validates extracted data by sending a *DQL* query to the specified Documentum Server repository.

A physical connection is established between the capture server that executes a CaptureFlow and a REST service that communicates with one or more Documentum Servers. The service address, the doabase name, and authentication parameters are specified in the Documentum connection (capture system component) that is included as a parameter in a DQL query. Encrypted authentication (Kerberos) is supported as an option.

You can create and edit Documentum connections in the **System** tool of Designer. The **Documentum Connection** list (**System** > **System Configuration** tab > **Configuration Settings** list > **Documentum Connection** option) displays existing connections to Documentum Server repositories.



**Note:** The **Documentum Connection** list may also include sample connections for testing purposes. Sample connections are only available if you create a


capture system with the **Copy Samples** option enabled. For more detail, see topic [“Adding a new capture system” on page 30](#).



If your CaptureFlow validates extraction results using data from one or several Documentum Server repositories, you need to create a Documentum connection to each repository. Then you can reuse these connections in as many DQL queries as needed.

**To create a Documentum connection:**

1. Open the **Documentum Connection** list as described above.
2. Select a capture server and connect to it. When disconnected, you still can create components, including Documentum connections, but you cannot specify connection parameters and test connections.
3. Click **<New Connection>** in the **Documentum Connection** list and specify the name of the new connection following the **naming convention**. The Documentum connection cannot be renamed.
4. Define the new connection parameters as stated in the following table.

**Table 3-2: Documentum connection parameters**

Parameter	Description
Address	<p>The network address of the Documentum REST services configured to connect to the Documentum Server(s). See the following supported formats:</p> <ul style="list-style-type: none"> <li>• http://&lt;machine_name&gt;:&lt;port&gt;/dctm-rest</li> <li>• http://&lt;IP_address&gt;:&lt;port&gt;/dctm-rest</li> </ul> <p>If Kerberos authentication is selected, show the REST services address in format http://&lt;machine_name&gt;:&lt;port&gt;/dctm-rest. The <i>IP</i> address cannot be used.</p> <p> <b>Note:</b> Documentum REST services are not Intelligent Capture component. These services must be installed and configured to be able to communicate with Documentum Servers whose repositories (docbases) will be accessed by your CaptureFlow in production. Contact your administrator to get the IP addresses of the required REST services.</p>
Docbase	The name of the Documentum Server repository to be used for the lookup.

Parameter	Description
<b>Method</b>	<p>The <i>HTTP</i> method for sending client DQL requests. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>GET</b>: Default. Use this method to get a response with better cache performance.</li> </ul> <p> <b>Note:</b> The client library may fail to handle the HTTP GET request properly when the DQL query is very long (for example, over 2,000 characters).</p> <ul style="list-style-type: none"> <li>• <b>POST</b>: Use this method to perform a query with a long DQL statement.</li> </ul>
<b>AuthenticationType</b>	<p>An HTTP authentication type used by the Documentum REST services. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Basic</b>: Default. The Documentum Server user credentials are sent in BASE64 encoded plain text.</li> <li>• <b>Kerberos</b>: The Documentum Server user credentials are encrypted using Kerberos tickets.</li> </ul> <p> <b>Note:</b> The Kerberos authentication and deployment process is described in <i>OpenText Documentum REST Services - Development Guide (EDCPKRST-PGD)</i>.</p>
<b>User</b>	<p>The Documentum Server user name. If Kerberos authentication is selected, the wildcard (*) character in this field indicates that the current Windows user credentials will be used.</p>
<b>Password</b>	<p>The Documentum Server user password. Can be empty only in Kerberos mode if the <b>User</b> parameter shows the wildcard (*) character.</p>

5. Test the connection by clicking the **Test** button.

Designer attempts to connect to the Documentum Server repository using the specified connection parameters. If successful, the connection is marked with a green check icon in the **Documentum Connection** list. A failed connection is marked with a red cross icon, and the issue description appears in the **Problem** panel.

6. (Optional) Specify the above connection properties for other capture servers available in the capture system. By default, the connection holds the same set of properties for all capture servers and, when installed on any of these servers, points the same data source.

### 3.3.3 Creating a file system connection

A file system connection describes a capture system connection to a particular text file located in the local file system or in the network. A file system connection is a required parameter in a text file query. If your CaptureFlow populates document fields from a text file or validates extracted values using text file data, you need to create a file system connection to access this file. Then you can reuse this connection in as many text file queries as needed.

#### To create a file system connection:

1. Go to **System > System Configuration tab > Configuration Settings list > File System Connection**.
2. Select a capture server and connect to it. When disconnected, you still can create a file system connection, but you cannot edit connection properties and test connections.
3. Click **<New Connection>** in the **File System Connection** list. Specify the name of the new connection following the **naming convention**. The file system connection cannot be renamed.
4. Define the new connection parameters as stated in the following table.

**Table 3-3: File system connection parameters**

Parameter	Description
Protocol	FileSystem or FTP.
<b>FileSystem</b>	
FileDirectory	The full path to the <b>formatted text file</b> to be used for the lookup. The path can optionally include the file name and extension.  You can specify the path in the local file system, or point the file in the network using a path with the mapped drive letter or a <b>UNC</b> path. Make sure that the specified network directory can be accessed without authentication.
FileName	Optional. If the file name is not included in the <b>FileDirectory</b> parameter, specify it here.
<b>FTP</b>	

Parameter	Description
URL	URL to the FTP or SFTP (SSH File Transfer Protocol) site.
SSL	Whether or not to use FTP over SSL/TLS.
UserName	Name of the user to log in with.
Password	Password for the user.
Description	A short description to document the purpose of this FTP or SFTP site.

5. Test the connection by clicking the **Test** button. If the file is found at the specified path, the connection is marked with a green check icon in the **File System Connection** list. A failed connection is marked with a red cross icon, and the issue description appears in the **Problem** panel.
6. (Optional) Specify the above connection properties for other capture servers available in the capture system. By default, the connection holds the same set of properties for all capture servers and, when installed on any of these servers, points the same data source.

### 3.3.4 Creating an email connection

An email connection describes a network connection of the capture system to a particular mail server. Email connections are used in export and import profiles.

**To create an email connection:**



1. Open the **Email Connection** list from **System > System Configuration > Configuration Settings > Email Connection**.
2. Select a capture server and connect to it.  
When disconnected, you still can create an email connection, but you cannot edit connection properties and test connections.
3. Click **<New Connection>** in the **Email Connection** list and specify the name of the new connection following the **naming convention**.







**Note:** Email connections cannot be renamed.

4. Define the email connection parameters as stated in the following table.

Table 3-4: New Email Connection Parameters

Parameter	Description
Protocol	<p>Select a protocol to use to communicate with the mail server.</p> <p> <b>Notes</b></p> <ul style="list-style-type: none"> <li>• When using POP3 SSL, configure the mail server certificate to be trusted by the client machine with the Designer and Standard Import modules installed. A self-signed certificate or a certification authority (CA) certificate should be added to the list of trusted root certification authorities on the client machine.</li> <li>• When using <b>Microsoft Graph</b>, the <b>Server</b>, <b>SSL</b>, and <b>Port</b> parameters are hidden because they are not required.</li> </ul>
Server	<p>Define a name or <i>IP</i> address of the mail server. The property is required.</p> <p> <b>Note:</b> When using POP3 SSL, mail server must be contacted by its DNS name, not IP address.</p>
SSL	<p>If necessary, improve security by using Secure Sockets Layer (<i>SSL</i>) or Transport Layer Security (<i>TLS</i>).</p> <p>Before using TLS, an appropriate security certificate must be installed. Specify the port number exactly in accordance with your server configuration.</p>
Port	<p>(Optional) Define the server connection port number. Range: 1 to 65535.</p> <p>Leave the default 0 to use the default port. The default port value for SSL is 465. If SSL is disabled, the default port is 25.</p>

Parameter	Description
<b>Timeout</b>	Timeout in seconds for mail operations. Default value is 0 (zero), which means that a time-out that works for the most common situations is used. This default value differs depending on the protocol. We recommend that you use the default; however, if you are going to process an email box with a large number of emails (typically more than 1000), then to help avoid a premature time-out, you could specify a larger time-out, such as 200 seconds or more.
<b>InboxFolder</b>	For <b>IMAP4</b> , <b>ExchangeWebDAV</b> , <b>ExchangeWebServices</b> and <b>Microsoft Graph</b> only, the folder from which to read messages.
<b>Authentication</b>	Select an authentication method for your mail server. If no authentication is required, then select <b>None</b> .   <b>Note:</b> NTLM is valid for Exchange Web Services only.  <b>OAuth</b> is valid for Microsoft Office 365 email (through Exchange Web Services, required for Microsoft Graph) and Standard Import only.
<b>OAuthTokenUrl</b>	OAuth 2 token URL supplied by the token provider. For Microsoft Office 365 email (through Exchange Web Services), specify the following (where <i>&lt;tenant&gt;</i> is supplied by the token provider):  https://login.microsoftonline.com/ <i>&lt;tenant&gt;</i> /oauth2/token   <b>Note:</b> Required for OAuth authentication.
<b>ClientId</b>	Application ID registered for authentication.   <b>Note:</b> Required for OAuth authentication.
<b>ClientSecret</b>	Your application's client secret, which was generated when you registered your client.   <b>Note:</b> Required for OAuth authentication.

Parameter	Description
User	Define the user name specified in the email account. This property is required.  For certain email protocols, the connection test might fail if another email client is connected using the same user account. It is recommended that all other email client connections for this account be closed before testing this email connection or using this email connection in production.
Password	Define the user password.

- Test the connection by clicking the **Test** button. If the connection to the mail server is successful, then the connection is marked with the green check icon in the **Email Connection** list. A failed connection is marked with a red cross icon, and the issue description appears in the **Problem** panel.
- (Optional) Specify the above connection properties for other capture servers available in the capture system. By default, the connection holds the same set of properties for all capture servers and, when installed on any of these servers, points to the same mail server.

### 3.3.5 Creating a CMIS repository connection

Starting in 22.2, Intelligent Capture supports browser binding in addition to the AtomPub binding type.



**Note:** The browser binding configuration has been tested only with Core Content .


- In **System > System Configuration** tab > **Configuration Settings** list > **CMIS Repository Connection**, click **<New Connection>** and type the name of the new connection following the **naming convention**.  
  
You cannot rename a connection.
- Define the new connection parameters as stated in **“CMIS-compliant repository connection parameters”** on page 45.



**Note:** To obtain the CMIS-compliant server's connection information, see your administrator .

**Table 3-5: CMIS-compliant repository connection parameters**


BindingType	Select one of the following binding types: <ul style="list-style-type: none"> <li>AtomPub</li> <li>Browser</li> </ul>
-------------	---

<b>URL</b>	The URL to the AtomPub Service Document (also sometimes referred to as the <i>service URL</i> ) or to the browser target.
<b>Required</b>	An asterisk (*) indicates that the attribute is required; otherwise, the attribute is optional.
<b>RepositoryId</b>	<p>The CMIS repository to connect to. For example, the URL to a Documentum CMIS-compliant repository might be the following:</p> <pre>https://15.45.92.234:8080/cmis/resources</pre> <p> <b>Note:</b> For more information about Documentum Content Management Interoperability Services, see <i>OpenText Documentum Content Management Interoperability Services - Reference Guide (EDCPKCO-ARE)</i>.</p>
<b>AuthenticationType</b>	<p>Select one of the following types of authentication:</p> <ul style="list-style-type: none"> <li>• <b>Basic:</b> Specify a valid user name (<b>User</b>) and password (<b>Password</b>).</li> <li>• <b>NTLM:</b> The Microsoft Windows NTLM security protocol. Specify a valid user name (<b>User</b>) and password (<b>Password</b>).</li> <li>• <b>OAuth:</b> The OAuth 2.0 standard for access delegation. Specify the following in addition to a valid user name (<b>User</b>) and password (<b>Password</b>): <ul style="list-style-type: none"> <li>– <b>OAuth 2 Token Url:</b> The URL to the OAuth service.</li> <li>– <b>Client ID:</b> Your OAuth client ID that is provided by the OAuth service administrator.</li> <li>– <b>Client secret:</b> Your OAuth client secret that is provided by the OAuth service administrator.</li> </ul> </li> </ul>
<b>User</b>	A valid user name.
<b>Password</b>	A valid password.
<b>Description</b>	(Optional) A description of the connection.

3. To test the connection, click **Test**.


If the connection is successful, the connection is indicated with a green check mark in the **CMIS Repository Connection** list. Failed connections are indicated with a red cross in the **CMIS Repository Connection** list, and the issue description is displayed in the **Problem** panel.

### 3.3.6 Creating an ApplicationXtender connection

 **Note:** For brevity, ApplicationXtender is referred to as AX.

#### To create an ApplicationXtender connection:

1. In **System > System Configuration** tab > **Configuration Settings** list > **ApplicationXtender Connection**, click <New Connection> and specify the name of the new connection following the **naming convention**.  
The connection cannot be renamed.
2. Define the new connection parameters as stated in “**AX connection parameters**” on page 47.


 **Note:** See your administrator for the AX server's connection information.

**Table 3-6: AX connection parameters**

Parameter	Description
URL	The URL to ApplicationXtender REST Services website.
DataSource	The AX data source name.
User	The name of the user account with which to connect to the AXserver.
Password	The password for the user account.

3. Test the connection by clicking the **Test** button.

If the connection is successful, then the connection is marked with a green check mark in the **ApplicationXtender Connection** list. If the connection fails, then the connection is marked with a red cross in the **ApplicationXtender Connection** list and the issue description is displayed in the **Problem** panel.

 **Note:** If the user account has been changed on the AX server, then you must first restart Designer and then specify another valid user account. However, if only the password has been changed, then you only need to correct the **password** parameter. Furthermore, if the IP address and/or host name of the AX server has been changed, then you must also first restart Designer and then correct the **URL** parameter.

### 3.3.7 Creating a Content Server connection

Content Server connection is required to setup and process the export content using the Content Server Export or xECM Export.

**To create a Content Server connection:**

1. In **System > System Configuration** tab > **Configuration Settings** list > **Content Server Connection**, click <New Connection> and specify the name of the new connection following the **naming convention**.

The connection cannot be renamed.

2. Define the new connection parameters as stated in the following table.

**Table 3-7: Content Server connection parameters**

Parameter	Description
URL	The Content Server URL; for example, <code>https://12.34.567.89/OTCS/cs.exe</code> .
User	The Content Server account name.
Password	The Content Server account password.
Domain	(Content Server only) The account domain.
OAuthAuthentication	Select to require OAuth authentication.
OAuthTokenUrl	OAuth 2 token URL.
ClientId	Application ID required for authentication.
ClientSecret	Your application's client secret, which was generated when you registered your client.

3. Test the connection by clicking the **Test** button:
  - If the connection is successful, it is marked with a green check mark in the **Content Server Connection** list;
  - If the connection fails, it is marked with a red cross in the **Content Server Connection** list and the issue description is displayed in the Problem panel.

### 3.3.8 Creating a SnapMatch connection

A SnapMatch connection generates a SnapMatch data file from a text file or SQL Server database. The SnapMatch data file is stored locally and is required for a SnapMatch population rule.

**To create a SnapMatch connection:**

1. In **System > System Configuration** tab > **Configuration Settings** list > **SnapMatch Connection**, click <New Connection> and specify the name of the new connection following the **naming convention**.

The connection cannot be renamed.

2. Define the new connection and data file parameters as stated in the following table.

**Table 3-8: SnapMatch connection parameters**

Parameter	Description
<b>SnapMatch Shared Directory</b>	The directory in which to store the SnapMatch data file. The default is <code>C:\Users\<i>&lt;username&gt;</i>\Documents\<i>&lt;product_name_and_version&gt;</i>\Default\GlobalData\SnapMatch.</code>
<b>Data Definition</b>	Specify the data format (for example, columns) of the SnapMatch import type ( <b>File System</b> or <b>SQL Server Database</b> ).
<b>Status</b>	Information about the import, such as whether the connection and import was successful or not and, if so, the name of the connection and a record count.

3. Specify the properties outlined in the following table for the data file and then click **Create** to create the data file.

Property	Description
<b>Import tab</b>	
<b>Import Type</b>	Specify the source for the import, that is, a text file from the <b>File System</b> or directly from an <b>SQL Server Database</b> .
<b>File System</b>	
<b>File System Connection</b>	Select a connection. These connections are populated from <b>System &gt; System Configuration</b> tab > <b>Configuration Settings</b> list > <b>File System Connection</b> , where <b>Protocol</b> is <b>FileSystem</b> .

Property	Description
Encoding	Select an IANA encoding. Default is 65001 Unicode (UTF-8).
Text Delimiter	Select a text delimiter.
Separator	Select a column separator.
<b>SQL Server Database</b>	
Connection	Select a connection. These connections are populated from <b>System &gt; System Configuration tab &gt; Configuration Settings list &gt; Database Connection</b> , where <b>Driver</b> is a Microsoft SQL Server driver.
Statement	Specify a SQL statement to select the relevant rows from the SQL Server database table.
Test Statement	Click to test the SQL statement.
Columns to Match tab	In the <b>To Match</b> column, select the columns (in the <b>Name</b> column) in the text file or SQL Server database table that you want to match fields in the document type. By default, all the words for a column value must occur as a continuous string in the document.  Checking <b>Match per Word</b> for a column indicates that the words in the column value do not have to occur as a continuous string in the document to be a valid match.
Replace Text tab	Specify replacements to be applied to the text on the document and to the column values of the SQL Server database table or text file columns prior to data matching.

- To verify that the data file exists in the specified SnapMatch shared directory, click **Test**.

### 3.3.8.1 Automatically Refreshing a SnapMatch Connection

In the Designer interface, you can manually refresh the SnapMatch data file. To automate this task, you can create a Windows Task Scheduler task that updates the SnapMatch data file regularly at user-specified intervals.

We cannot make recommendations about how frequently you should refresh the SnapMatch connection. If you have a lot of data, then you will need to give the application ample time to run.

**To automatically refresh a SnapMatch connection:**

1. From the Start menu, select **Administrative Tools > Task Scheduler**.
2. Right-click Task Scheduler Library and select **Create Task**.
3. On the General tab, give your task a name and description.
4. On the Triggers tab, click **New** and set the schedule for running the task by setting the start date and recurrence.
5. Click **OK**.
6. On the Actions tab, click **New** and do the following:
  1. In the Action field, select **Start a program**.
  2. In the Program/script field, navigate to your UpdateSnapMatchData.exe file.  
For example, C:/"Program Files (x86)"/InputAccel/Client/binnt/UpdateSnapMatchData.exe
  3. In the Add arguments field, add your arguments. For example:  
`-profiles:* -solution:default`
  4. Click **OK**.
7. Click **OK**.
8. To close the Task Scheduler, click **File > Exit**.

**3.3.9 Copying connections**

A quick way to create an already configured connection is to save a copy of the existing connection with a new name.

**To create a connection based on an existing one:**

1. Select **System** and open the **System Configuration** tab. Select the capture system and the capture server.
2. Expand the **Configuration Settings** drop-down list and select the type of connection. In the list of available connections, select the connection that you want to copy.  
The connection will be copied with the connection parameters specific for the currently selected capture server.
3. Click the **Save** button on the toolbar and select the **Save As** option.
4. In the **Configuration – Save As** window, specify the name of the new connection following the **naming convention**.  
The new connection with the defined parameters appears in the list of connections.

## 3.4 Defining styles

A style is a collection of visual settings that apply to data entry fields and controls in the Identification and Completion modules. A style has a unique descriptive name and specifies the font settings, the foreground and background color, the color and style of the field borders, and the text padding.

Select **System** and expand the **System Configuration** tab. To view the styles already defined in the capture system, expand the **Configuration Settings** drop-down list and select the following options:

- **System Styles:** This list includes the predefined styles that ship with Designer. When installed on the capture server, these styles apply to the fields and controls of all Identification and Completion modules triggered by the processes running on this server. Each system style applies to a particular type of control or to a data entry field with particular status.

The list of system styles cannot be extended, but the style settings can be **customized**.

- **Custom Styles:** This list is empty by default. If necessary, you can **create custom styles** and assign each to particular fields in a document type. When installed on the capture server, a custom style applies to these specific fields and controls in Identification and Completion if the currently processed document belongs to the document type in question.

A custom style assigned to a document type does not override the field/control visual settings where they are set by the system style.

### 3.4.1 Customizing system styles

**To customize a system style:**

1. Select **System** and the **System Configuration** tab. Select the **System Styles** category in the **Configuration Settings** drop-down list.

The system styles are available as stated in **“System styles” on page 52**.

**Table 3-9: System styles**

System style	Description	File name
<b>AnnotationText</b>	Applies to the annotation text.	AnnotationText.config
<b>FieldWithError</b>	Applies to the data entry fields set in error.	FieldWithError.config
<b>FieldWithFlag</b>	Applies to the flagged data entry fields.	FieldWithFlag.config
<b>MessageBar</b>	Applies to the text in the message bar.	MessageBar.config

System style	Description	File name
ReadonlyField	Applies to the read-only data entry fields.	ReadonlyField.config
SelectedField	Applies to the selected data entry fields.	SelectedField.config

- In the **System Styles** list, select the style you need to modify.
- In the style's property sheet, double-click the **Value** you want to change, and provide a new value by selecting the corresponding check boxes or a value from a drop-down list.



**Note:** The system styles with the text attribute imply fonts that are available on the Designer host machine. These include system default fonts, Microsoft Office fonts, and any installed third-party fonts registered in the system.

- Save the changes.

The updated system styles are saved to their local `.config` files. The updates will be available on the server after deployment.

### Related Topics:

[“Defining styles” on page 52](#)

[“Adding custom styles” on page 53](#)

## 3.4.2 Adding custom styles

A custom style can be linked to particular fields, controls, and labels in a document type and this way define how the Identification or Completion operator will view these fields, controls, and field/control labels in the data entry form. A custom style does not override the field/control visual settings where they are set by the system style.

### To create a custom style:

- Select **System** and select the **System Configuration** tab. Select the **Custom Styles** category in the **Configuration Settings** drop-down list.
- In the **Custom Styles** list, click **<NewStyle>**. Specify a unique name for the new style following the **naming convention**.



**Note:** Custom styles cannot be renamed. However, you can create a copy of the existing custom style with the same properties and a different name using the **Save As** menu command.

3. Specify the style properties by selecting the corresponding check boxes or selecting a value from a drop-down list.
4. Save the changes. The <custom style name>.config file is created in the \<capture system name>\GlobalData\CustomStyle folder in the Designer working directory. The styles will be available on the server after deployment.

“Style settings for fields and controls” on page 54 describes how custom style settings affect fields and controls when they are displayed in the data entry form.

**Table 3-10: Style settings for fields and controls**

Properties	Text Box / List Box	Drop-down List Box / Combo Box	Check Box	Text Block / Label <sup>[a]</sup>	Rectangle / Group Box	Button <sup>[b]</sup>	Image
Font (including name, size, weight, underline, strikethrough, italic)	Yes (value) except that strikethrough and underline properties do not apply to text box or list box fields.	Yes (value) except that strikethrough and underline properties do not apply to drop-down list box or combo box fields.	Yes (button text)	Yes (text)	Yes (group box name)	Yes (button label)	No
Background Color	Yes (control rectangle)	No	Yes (control rectangle)	Yes (control rectangle)	Yes (control rectangle)	Yes	No. Picture is stretched, so not used.
Foreground Color	Yes (value)	No	Yes (button text)	Yes (text)	Yes (group box name and border)	Yes (button label and border)	Yes (border)
Line Style	No	No	No	No	Yes (border)	No	Yes (border)
Line Thickness	No	No	Yes	No	Yes	Yes	Yes
Corner Style	No	No	No	No	Yes	No	Yes

Properties	Text Box / List Box	Drop-down List Box / Combo Box	Check Box	Text Block / Label <sup>[a]</sup>	Rectangle / Group Box	Button <sup>[b]</sup>	Image
Text Padding	Yes	Yes	Yes	Yes	Yes	Yes	No

<sup>[a]</sup> Includes labels for Text Box, List Box, and Drop-down List Box fields.

<sup>[b]</sup> Includes label/button text for Check Box fields.

### Related Topics:

[“Defining styles” on page 52](#)

[“Customizing system styles” on page 52](#)

## 3.5 Defining shortcuts

You can specify the system-wide shortcuts for the Identification and Completion modules. When deployed on the Intelligent Capture Server, these shortcuts will be passed in to the operator modules triggered by any capture process on that server.

### 3.5.1 Customizable module shortcuts

Identification and Completion operators can use keyboard shortcuts to quickly perform common tasks and actions in the application. The shortcut keys provided to the operators are defined in the `<DesktopShortcutKeys>.config` file that is located on the server where the batch resides. The shortcuts in this file can be customized as described in this topic. Once the customized `<DesktopShortcutKeys>.config` file is uploaded to the server, all Identification and Completion modules receiving tasks from this server share the same list of shortcuts.

To customize the operator module shortcuts, go to the **System Configuration** tab, expand the **Configuration Settings** drop-down list, and select the **Intelligent Capture Desktop Shortcuts** option. The list of actual shortcuts appears. You can click any shortcut and enter a key combination.

The following key combinations cannot be used:


- Reserved shortcuts (see [“Reserved module shortcuts” on page 62](#))



**Note:** Windows shortcuts not listed as a reserved shortcut can be overridden.

- Any single alphanumeric, punctuation (except for ~ (tilde)), or symbol key by itself

[“Identification and Completion default keyboard shortcuts” on page 56](#) provides the list of Identification and Completion default shortcuts that can be customized.

 **Note:** All shortcuts need to be uppercase.

**Table 3-11: Identification and Completion default keyboard shortcuts**

Property	Default shortcut key	Action
Section <b>Task Management:</b> Actions with tasks.		
TaskDone	F10	Finishes work on the current task.
TaskStop	F11	Cancels/abandons current task, and continues on to the next task.
TaskCancel	F12	Stops working on tasks.
Section <b>Navigation:</b> Navigation through documents, pages, or segments.		
SetFocusToForm	F7	Moves the focus to the data entry form.
PageMoveNext	CTRL+SHIFT+K	Sets the focus to the next page.
PageMovePrevious	CTRL+SHIFT+J	Sets the focus to the previous page.
DocumentMoveNext	CTRL+M	Sets the focus to the first field of the next document.
DocumentMovePrevious	CTRL+U	Sets the focus to the first field of the previous document.
SegmentMoveNext	CTRL+K	Sets the focus to the first field in the next segment.
SegmentMovePrevious	CTRL+J	Sets focus to the first field in the previous segment.
Section <b>Document Actions:</b> Actions with documents, document folders, and document fields.		
DocumentTypeSet	CTRL+D	Changes the document type for the current document.
MergeDocument	F3	Merges the selected document with the previous one. Duplicates the <b>Document &gt; Merge Document</b> menu option.
MergeFolder	(not defined)	Merges the selected folder with the previous one. Duplicates the <b>Document &gt; Merge Folder</b> menu option.

Property	Default shortcut key	Action
MergeStack	(not defined)	Merges the selected stack with the previous one. Duplicates the <b>Document &gt; Merge Stack</b> menu option.
MergeLevel4	(not defined)	Merges the selected “level 4” node with the previous one. Duplicates the <b>Document &gt; Merge Level4</b> menu option.
MergeLevel5	(not defined)	Merges the selected “level 5” node with the previous one. Duplicates the <b>Document &gt; Merge Level5</b> menu option.
MergeLevel6	(not defined)	Merges the selected “level 6” node with the previous one. Duplicates the <b>Document &gt; Merge Level6</b> menu option.
SplitFolder	<b>F4</b>	Splits the folder by the selected document. Duplicates the <b>Document &gt; Split Folder</b> menu option.
SplitFolderUpToStack	(not defined)	Splits the stack by the selected document. Duplicates the <b>Document &gt; Split Folder up to Stack</b> menu option.
SplitFolderUpToLevel4	(not defined)	Splits the “level 4” node by the selected document. Duplicates the <b>Document &gt; Split Folder up to Level4</b> menu option.
SplitFolderUpToLevel5	(not defined)	Splits the “level 5” node by the selected document. Duplicates the <b>Document &gt; Split Folder up to Level5</b> menu option.
SplitFolderUpToLevel6	(not defined)	Splits the “level 6” node by the selected document. Duplicates the <b>Document &gt; Split Folder up to Level6</b> menu option.
NewDocument	<b>CTRL+N</b>	Splits the document by the selected page. Duplicates the <b>Document &gt; New Document</b> menu option.

Property	Default shortcut key	Action
NewEmptyDocument	(not defined)	Adds a new empty document to the task tree structure. Duplicates the <b>Document &gt; New Empty Document</b> menu option.
FlagItem	CTRL+G	Flags the current document. Duplicates the <b>Document &gt; Flag</b> menu option.
Section <b>Template Actions (Identification Only)</b> : Actions with page templates. Available in Identification only.		
TemplateAssignLastUsed	CTRL+SHIFT+Z	Assigns the last used template to the selected page(s) in the <b>Documents</b> panel. Duplicates the <b>Template &gt; Assign Last Used to Selected Page(s)</b> menu option.
TemplateAssignLastToRest	CTRL+SHIFT+X	Assigns the last used template to the rest of the document pages. Duplicates the <b>Template &gt; Assign Last Used to the Rest of This Document</b> menu option.
Section <b>Table Actions (Completion Only)</b> : Actions with tables. Available in Completion only.		
TableInsertRow	CTRL+INSERT	Inserts a table row. Duplicates the <b>Table &gt; Insert Row</b> menu option.
TableDeleteRow	CTRL+DELETE	Deletes the selected table row. Duplicates the <b>Table &gt; Delete Row</b> menu option.
TableClear	CTRL+SHIFT+DELETE	Clears the table if the selection is a table field. Duplicates the <b>Table &gt; Clear</b> menu option.
TableRowMoveUp	(not defined)	Moves the selected table row up. Duplicates the <b>Table &gt; Row Move Up</b> menu option.
TableRowMoveDown	(not defined)	Moves the selected table row down. Duplicates the <b>Table &gt; Row Move Down</b> menu option.

Property	Default shortcut key	Action
TableRowMerge	(not defined)	Merges the selected table row with the previous one. Duplicates the <b>Table &gt; Row Merge</b> menu option.
TableExtractionMode	(not defined)	Enters the table extraction mode. Duplicates the <b>Table &gt; Extraction Mode</b> menu option.
Section <b>Page Actions</b> : Actions with pages.		
ToggleDefaultZoom	(not defined)	Toggles on/off the default zoom.
PageZoomIn	<b>CTRL++</b> <b>CTRL+MOUSE WHEEL UP</b>	Zooms in the current page. Duplicates the <b>Page &gt; Zoom In</b> menu option.
PageZoomOut	<b>CTRL+-</b> <b>CTRL+MOUSE WHEEL DOWN</b>	Zooms out the current page. Duplicates the <b>Page &gt; Zoom Out</b> menu option.
PageZoomReset	<b>CTRL+*</b> (asterisk)	Resets zoom to the default value of page width for the current page. Duplicates the <b>Page &gt; Zoom Reset</b> menu option.
PageZoomFitToWidth	(not defined)	Zooms the page to fit to width.
PageZoomFitToHeight	(not defined)	Zooms the page to fit to height.
PageZoomFitToWindow	(not defined)	Zooms the page to fit to window.
PageRotateLeft	<b>CTRL+1</b>	Rotates the current page left. Duplicates the <b>Page &gt; Rotate Left</b> menu option.
PageRotateRight	<b>CTRL+3</b>	Rotates the current page right. Duplicates the <b>Page &gt; Rotate Right</b> menu option.
PageRotate180	<b>CTRL+0</b> (zero)	Rotates the current page by 180 degrees. Duplicates the <b>Page &gt; Rotate 180</b> menu option.
PagePanUp	<b>CTRL+8</b>	Pans up on the current page. Duplicates the <b>Page &gt; Pan Up</b> menu option.

Property	Default shortcut key	Action
PagePanDown	CTRL+2	Pans down on the current page. Duplicates the <b>Page &gt; Pan Down</b> menu option.
PagePanRight	CTRL+6	Pans right on the current page. Duplicates the <b>Page &gt; Pan Right</b> menu option.
PagePanLeft	CTRL+4	Pans left on the current page. Duplicates the <b>Page &gt; Pan Left</b> menu option.
PageAutoPan	CTRL+5	Enters the automatic panning mode, which automatically shows the image zone for the currently focused entry field. If already in that mode, exits the automatic panning mode. Duplicates the <b>Page &gt; Auto Pan</b> menu option.
PageAnnotationMode	CTRL+T	Enters the annotation mode for the current page. If already in that mode, exits the annotation mode. Duplicates the <b>Page &gt; Annotations</b> menu option.
ToggleClickToExtract	CTRL+E	Toggles <b>Page &gt; Click to Extract</b> menu.
Section <b>View Actions</b> : Toggling between elements actions.		
TogglePageBrowserView	F9	Expands and collapses task nodes in the <b>Documents</b> panel.
ToggleFieldsSummary	(not defined)	Displays or hides the <b>Summary</b> panel in Completion. Duplicates the <b>View &gt; Summary</b> menu option.
TogglePromptAtEnd	(not defined)	Toggles <b>View &gt; Prompt at &gt;End of Document</b> .
TogglePromptAtEndofTask	(not defined)	Toggles <b>View &gt; Prompt at &gt; End of Task</b> .
TogglePromptAtDeletePage	(not defined)	Toggles <b>View &gt; Prompt at &gt; Delete Page</b> .
TogglePromptAtDeleteDocument	(not defined)	Toggles <b>View &gt; Prompt at &gt; Delete Document</b> .
TogglePromptAtDeleteFolder	(not defined)	Toggles <b>View &gt; Prompt at &gt; Delete Folder</b> .

Property	Default shortcut key	Action
ToggleResizableMessageBar	(not defined)	(Completion only) Enables or disables the ability to resize the height of the message bar. If resizing is disabled, then message bar height is automatically adjusted to fit the content. If resizing is enabled, then the user could manually adjust the message bar height. If only some of the content can be displayed, then a scroll bar is displayed. Also, the height is saved and restored for the user's next session. Duplicates the <b>View &gt; Resizable Message Bar</b> menu item.
ToggleTableView	(not defined)	Switches between the grid view and record view modes in the <b>Form</b> panel of Completion. Duplicates the <b>Table &gt; Grid View/Record View</b> menu option.
ToggleImageSnippets	(not defined)	Enables or suppresses the display of image snippets near the selected field in the data entry form. Duplicates the <b>View &gt; Image Snippets</b> menu option.
ToggleImageSnippetsTemporarily	(not defined)	Hides/shows the image snippet on the focused field. Works one time on a particular field. This functionality is not duplicated in any menu option.
ToggleFormPosition	(not defined)	Duplicates the <b>View &gt; Form on Right/Form on Bottom</b> menu option.
ToggleImagePosition	(not defined)	Displays the <b>Page View</b> panel in a separate window or docks it into the main window. Duplicates the <b>View &gt; Dock/Float Page View</b> menu option.

## Related Topics

*“Reserved module shortcuts” on page 62*

### 3.5.2 Reserved module shortcuts

This topic provides the full list of reserved shortcuts that cannot be used as **custom module shortcuts**. They are reserved for use by the Identification and Completion controls.

“**Identification and Completion: Reserved Shortcut Keys**” on page 62 provides a full list of reserved shortcuts.

**Table 3-12: Identification and Completion: Reserved Shortcut Keys**

Property	Default shortcut key	Action
DocHelp	F1	Displays help.
PanelFocus	CTRL+TAB	Moves the focus to the next panel.
PanelFocusBackward	CTRL+SHIFT+TAB	Moves the focus to the previous panel.
ShowToolTip	SHIFT+F1	Displays the tool tip.
Print	CTRL+P	Launches the <b>Print</b> dialog to print the current page or document.
Cut	CTRL+X	Cuts the current selection.
Copy	CTRL+C	Copies the current selection.
Delete	DELETE	Deletes the current selection.
Edit	F2	Edit
EscapeCurrent	ESC	Closes the current drop-down list.
FieldMovePreviousVisited	CTRL+BACKSPACE	Sets the focus on the previously-focused field.
FieldFocusOnSearch	CTRL+F	Sets the focus on the <b>Find</b> field.
FieldMoveNextNeedingAttention	ENTER / RETURN	Sets the focus on the next work field that needs validation.
Paste	CTRL+V	Paste
Undo	CTRL+Z	Undo

## 3.6 Defining global options

You can configure the behavior and appearance of the Identification and Completion modules (also referred to as desktop modules) running in the network of a particular company. These settings are installed on the Intelligent Capture Server of that company as global options. All capture processes of the company running on this server share the same set of global settings, including the client module settings. When a process triggers the Identification or Completion module on the operator machine, the client module settings are passed in to that module with a task.

The client module settings are described in this section.

### 3.6.1 Batch level names

In production, Identification and Completion both display the task in the tree view panel. The task nodes above page level display the node name, for example, **Document**, **Folder**, and **Other Options**.

**To customize the displayed node names:**

1. Select **System** and the **System Configuration** tab.
2. Select the **Other Options** category and **Global Options** in the **Configuration Settings** drop-down list.
3. In the **Batch Level Names** section, click in the **Value** field against each of the listed node levels and enter the preferred node name as outlined in the following table.



**Note:** If you do not specify the custom name for any level, the default name will be displayed. The following node names are used by default:

Displayed name	Node level	Note
Page	Level 0 (Page level)	Pages are displayed in the tree view without node name.
Document	Level 1	
Folder	Level 2	
Stack	Level 3	
Level4	Level 4	
Level5	Level 5	
Level6	Level 6	
Level7	Level 7 (Batch level)	The batch node is never displayed in the tree view.

The updated node names are saved to the local `GlobalOptions.config` file automatically. The updates will be available on the server after deployment.

### 3.6.2 Annotation colors

The Identification and Completion modules both allow the operator to annotate pages with several types of annotations, such as stamps, rectangles, text, and others. Annotation line and background have a default color which can be configured and installed on the Intelligent Capture Server as global options. Also, you can specify a palette of eight colors to be available to the operator if the default annotation color does not suit.

#### To customize default annotation colors:

1. Select **System** and the **System Configuration** tab.
2. Select the **Other Options** category and **Global Options** in the **Configuration Settings** drop-down list.
3. In the **Annotations** section, customize the color palette in properties **Color1** to **Color8**. To customize the color, click in the **Value** field of a property and select the color from the color picker.
4. Customize other properties in the **Annotations** section by selecting the default color from the drop-down list:
  - **AnnotationArrowLineColor**: Default color used by the “arrow” annotation type.
  - **AnnotationHighlightColor**: Default color used by the “highlight” annotation type.
  - **AnnotationRedactColor**: Default color used by the “redact” annotation type.

This list shows the colors defined in the color palette.

The updated colors are saved to the local `GlobalOptions.config` file automatically. The updates will be available on the server after deployment.

### 3.6.3 Image zone highlight colors

Both the Identification and Completion modules can receive documents with recognized image zones that can be highlighted when the operator set the focus in the field bound to this zone. Also, Identification and Completion can highlight particular characters with low confidence rate (displayed in the field as question marks). The highlight color for both cases can be customized.

#### To customize image zone highlight colors:

1. Select **System** and the **System Configuration** tab.
2. Select the **Other Options** category and **Global Options** in the **Configuration Settings** drop-down list.

3. In the **Intelligent Capture Desktop** section, customize the highlight color for the following properties:
  - **FieldZoneColor**: Default color used to highlight an image zone for the field in focus.
  - **CharZoneColor**: Default color used to highlight uncertain characters on an image.

To customize the color, click in the **Value** field of a property and select the color from the color picker.

The updated colors are saved to the local `GlobalOptions.config` file automatically. The updates will be available on the server after deployment.

### 3.6.4 Panel background color

You can customize the background color for the tree view and image view panels in the Completion and Identification modules.

**To customize the background color for Tree View and Page View:**

1. Select **System** and the **System Configuration** tab.
2. Select the **Other Options** category and **Global Options** in the **Configuration Settings** drop-down list.
3. In the **Intelligent Capture Desktop** section, click in the **Value** field of the **PageBackgroundColor** property and select the color from the color picker.

The updated color is saved to the local `GlobalOptions.config` file automatically. The update will be available on the server after deployment.

### 3.6.5 Thumbnail size

You can customize the size of large thumbnails that are used to display pages in the tree view panel of the Completion and Identification modules. If the thumbnail size options are defined by the batch creating module, such as ScanPlus, then the client modules use that size when generating thumbnails. Otherwise, they use the specified thumbnail size.

**To customize the thumbnail size in page browser:**

1. Select **System** and the **System Configuration** tab.
2. Select the **Other Options** category and **Global Options** in the **Configuration Settings** drop-down list.
3. In the **Intelligent Capture Desktop** section, customize the highlight color for the following properties:
  - **ThumbnailHeight**: Default height of the large thumbnails set in pixels.

- **ThumbnailWidth:** Default width of the large thumbnails set in pixels.

To customize the property, click in the **Value** field and enter the custom value.

The updated thumbnail properties are saved to the local `GlobalOptions.config` file automatically. The updates will be available on the server after deployment.

### 3.6.6 CTRL key behavior in image view

**To change the CTRL key behavior in image view:**

1. Select **System > System Configuration tab > Other Options > Global Options > Intelligent Capture Desktop > CtrlClickBehavior**.
2. In the **Value** field, select one of the following:
  - **Click to Extract:** (Default as of 16.6) The operator can select individual characters using **CTRL+click**. The Click to Extract feature must be enabled; otherwise, **CTRL+click** does not do anything.
  - **Image Zoom:** Use **CTRL+left-click/right-click** to zoom in/out on the image


### 3.6.7 Non-system files for deployment

You can deploy on the selected server a list of files that are not part of the capture system. In particular, you can deploy document type scripting, code for the .Net Code module, and other kinds of .NET script code created for a capture process.

Every assembly with the unique name exists on the selected server as a single instance.

**To configure file deployment on the capture server:**

1. Select **System > System Configuration > Configuration Settings > Other Options > DeploymentFiles**.
2. Click **Add** to add files to be deployed and **Remove** to remove them.

 **Note:** If the file was not already in `C:\Users\<username>\Documents\<Product name>\<Capture System name>\bin`, then it is copied to that directory. To be uploaded to the capture server, the file must reside in this directory.

3. To deploy the listed files, connect to the server, select **System > Deployment > Configuration Settings – Deployment Files**, check each file, and click **Upload**.

#### **Notes**

For more information, see the following:

- Files larger than **4 MB** are not uploaded using the deployment mechanism and must be manually copied on the client machines where the respective modules are installed.

- Commas are not allowed in file names.
- In production, when the process triggers Completion, Extraction, or the .Net Code module, the deployed files are pulled to the %localappdata%\Emc\InputAccel\Custom\Bin directory on the client machine automatically. If you need to replace the files in the %localappdata%\Emc\InputAccel\Custom\Bin directory with a new version, copy the new files to the bin folder of the current capture system again, deploy them to the server, and restart the module.
- When debugging .NET scripting, it is recommended to create the %localappdata%\Emc\InputAccel\Custom\DebugBin directory on the designer's machine and copy the *DLL* and *PDB* files into it. The previously deployed script files must be removed from the capture server.
- For pre-7.6 **DeploymentFiles** fields, the format in which files are comma-separated is preserved.

### 3.6.8 Recognition project path

Advanced recognition modules (Classification, Identification, Extraction, and Collector) read part of their configuration settings from the recognition project file (DPP) that must be indicated for each advanced recognition CaptureFlow step during setup. Classification and Collector are set up with a full path to the project file. Identification and Extraction only know the project name (the root file name without extension). To find the recognition project in the network, these modules read the project path from the global options (GlobalOptions.config) uploaded on the Intelligent Capture Server.



**Note:** Recognition projects are not deployed using the Designer and not stored on the capture server. All recognition projects used by all capture processes installed on a particular Intelligent Capture Server must be located in one shared folder. Creating this folder and granting access rights is the responsibility of the capture process designer.

#### To specify the recognition project path:

1. Select **System** and the **System Configuration** tab.
2. Select the **Other Options** category and **Global Options** in the **Configuration Settings** drop-down list.
3. In the **File Management** section, specify the *UNC* path to the shared folder in the **RecognitionProjectSharedDirectory** property.

The updated path is saved to the local GlobalOptions.config file automatically. The update will be available on the server after deployment.

## 3.7 Defining queries

A capture system can optionally include a collection of queries. Each query includes all necessary information for retrieving data from an external data source such as a database, or a Documentum repository, or a structured text file. A query is intended for use in a population or a validation rule created within a document type. An existing query can be reused in as many rules and document types as needed.

A query is identified by a unique name in the given capture system. It includes a connection (capture system component) to an external data source and specifies the data lookup criteria in the query statement. A query statement can also include input parameters which serve as placeholders to be substituted by external values at runtime. For example, you can create a query-based rule such that a dependent field maps an input parameter. In production, whenever the field is updated, the field's value is assigned to the input parameter, and the triggered rule sends a request with that parameter to an external data source.

The following types of queries are supported:

- **ODBC query:** Serves for requesting data from an external database through *ODBC* using a *database connection*. The query statement uses the *SQL syntax*.
- **DQL query:** Serves for requesting data from a Documentum repository using a *Documentum connection*. The query statement uses the *DQL syntax*.
- **Text file query:** Serves for requesting data from a formatted text file using a *file system connection*. The statement of the text file query must comply with the *text file query syntax*.

### 3.7.1 Adding a query

To create a query, navigate to **System** and select **Queries**. The **Queries** tab serves for creating queries and managing queries that already exist in the selected capture system.

When created or modified, queries are automatically saved to the GlobalData\NamedQuery\ folder of the capture system.

#### To create a query:


1. In the **Query Name** column, click the <**New Query**> link. The dialog box opens.
  1. Specify the query name following the *naming convention*.
  2. Expand the **Query Type** drop-down list and select the type of query you want to create: **ODBC**, **DQL**, or **TextFile**.
  3. Click **Save**.

The new query appears in the list of queries with the **Query Name** and **Type** parameters already set.


2. Specify the query parameters as described in the table to follow.

**Table 3-13: Query parameters**

Parameter	Description
Connection Name	Expand the <b>Connection Name</b> drop-down list and select the name of the connection. If necessary, <b>create a connection</b> and then select it in the list. The <b>Connection Name</b> list is filtered to show connections that match the type of query.
Description	Enter a description of the query. When creating a query-based rule, this description will be displayed in the dialog box to help you select the right query.
SQL Query Documentum Query TextFile Query	Enter the query statement. The required syntax depends on the type query: <ul style="list-style-type: none"> <li>• For <b>ODBC queries</b>, enter an SQL statement</li> <li>• For <b>DQL queries</b>, enter a DQL statement</li> <li>• For <b>text file queries</b>, use special expression syntax</li> </ul>
Add parameter for docbase name	(Optional) If you are creating a <b>DQL</b> query, select the <b>Add parameter for docbase name</b> option to add the <b>DocbaseName</b> input parameter in the <b>Input Parameters</b> list.  When designing a document type, you can map this parameter to a data entry field and thus let the operator choose a different Documentum repository for data validation.

Parameter	Description
<p><b>Input Parameters</b></p>	<p>Click the <b>Update</b> button to refresh the <b>Input Parameters</b> list. All input parameters defined in the query statement are automatically populated or updated in the <b>Input Parameters</b> list with the following properties (grid columns):</p> <ul style="list-style-type: none"> <li>• <b>Input Name:</b> The name of the input parameter. Read-only, copied from a query statement.</li> <li>• <b>Data Type:</b> Expand this list and select the data type of the input parameter: <b>String</b> (default), <b>Number</b>, <b>DateTime</b>.</li> <li>• <b>Test Value:</b> Click in this field and initialize the input parameter with the value for testing. The test value will be inserted in the query and passed to the external data source. Test values are remembered and remain when you reopen the query.</li> </ul> <p> <b>Note:</b> If you remove an input parameter from the query statement and click <b>Update</b>, the parameter disappears from the <b>Input Parameters</b> list and the query is saved in this state automatically. You cannot undo this change. If you add this input parameter in the query statement again and press <b>Update</b>, you need to reenter the data type and the test value manually.</p>

3. Click the **Test** button to **test** your query. The result of the query execution appears in the message area near the **Test** button. For instance, you may view the following message: Query test successful. Returned rows: 2

 **Note:** If the **Test** button is dimmed, verify the following:

- You are connected to the capture server.
- The **Connection Name** parameter is defined for the query.
- The query statement is specified (not blank).

**Related Topics:**

[“DQL query statement” on page 71](#)

[“Testing queries” on page 73](#)

### 3.7.1.1 ODBC query statement

The syntax of the ODBC query statement is defined by the requirements and SQL extensions specific to the database management system to be requested. In a general case, keep to the *ANSI* standard *SQL* language. For more detail, see the query syntax documentation of the requested database management system.

When adding input parameters in the query statements, the following rules apply:

- Use the following syntax:  
: <parameter\_name>
- An input parameter name can be a maximum of 64 characters and must contain ASCII alphanumeric characters only.
- *ODBC* queries can have input parameters in the conditional part of the statement only. Input parameters added in the *SELECT* part of the query cannot be populated to the **Input Parameters** list and will be ignored.

### 3.7.1.2 DQL query statement

*DQL* queries are served by the OpenText Documentum Server that uses the DQL (Documentum Query Language). This language is based on *ANSI SQL* extended with Documentum objects.

For more information about DQL syntax, see *OpenText Documentum Foundation Classes - Development Guide (EDCPKCL-DGD)*.

For instance, the following DQL query fetches all document records that have the word "Invoice" in their title:

```
SELECT * FROM dm_document WHERE title LIKE '%Invoice%'
```

When adding input parameters in the query statements, the following rules apply:

- Use the following syntax:  
: <parameter\_name>
- An input parameter name can be a maximum of 64 characters and must contain ASCII alphanumeric characters only.
- DQL queries can have input parameters in the conditional part of the statement only. Input parameters added in the *SELECT* part of the query cannot be populated to the **Input Parameters** list and will be ignored. For example, the following DQL query demonstrates how to use the P2 parameter to fetch all document records with the specific word in the title:

```
SELECT * FROM dm_document WHERE title LIKE :P2
```

### 3.7.1.3 Text file query statement

To fetch data from a formatted text file, use syntax based on the **expressions**. The text file query statement must be formatted as follows:

```
<field_name> = :<input_parameter1> [AND <field_name> = :<input_parameter2> [AND ...]]
```

- When adding input parameters in the query statements, the following rules apply:
  - Use the following syntax:
    - : <parameter\_name>
  - An input parameter name can be a maximum of 64 characters and must contain ASCII alphanumeric characters only.
  - Each <input\_parameter> must match the corresponding column name in the text file.
- Each included <field\_name> must be assigned an <input\_parameter>. Constant values cannot be assigned. Furthermore, each included <field\_name> must match the corresponding <input\_parameter> name.
- If the formatted text file defines multiple field names in the header, include only those field names in which you do the search.
- Use the AND operator to join the statement parameters (file field names). Other operators are not supported.

The returned data set includes all lines that matches the search criteria. Each returned line includes all the fields that are defined in the file header. For example, the following query statement will fetch all lines from the text file where ContractID and HireDate match the values passed through the input parameters at runtime:

```
ContractID = :ContractID AND HireDate = :HireDate
```

If you need to return all file lines, add a special field in the text file and populate it with the same value in each line. Then use this special field in a query as a search parameter.

For instance, the file can include the FetchAllRows field populated with “1” in each row. The text file query fetching all lines may look as follows:

```
FetchAllRows = :FetchAllRows
```

The FetchAllRows input parameter must be set to “1” in the definition of the rule using this query.

### 3.7.2 Renaming a query

You cannot rename a query since it may be already in use. To save your query with a different name, use the **Save As** button.

### 3.7.3 Testing queries

Before you start testing queries, ensure the corresponding test button is not dimmed. If the **Test** or **Test All** button is disabled, verify that you are connected to a capture server, that the **Connection Name** parameter is defined for each query you are going to test, and that the query statements are not blank.

You can test each query in the capture system individually or run a test for all queries.

#### To test a query:

1. Open the **Queries** tab and select a query to test.
2. Verify that input parameters specified in the query statement are synchronized with the **Input Parameters** list. Click **Update** to update the **Input Parameters** list if necessary.
3. (Optional) If you are testing a query with input parameters, manually assign a value for each parameter in the **Input Parameters** list.
4. Click the **Test** button to start the test.

A query test checks the validity of the connection, the existence of all query parameters, and the query syntax. Then the query is executed and the number of returned rows appears in the result message area. If any part of the test fails, the result message area displays the error message.



**Note:** If no rows are returned, try to test the query again using different test values assigned in input parameters.

#### To test several queries simultaneously:

1. Open the **Queries** tab.
2. Check that the **Test All** button is available.
3. Verify that input parameters specified in each query statement are synchronized with the **Input Parameters** list. Click **Update** to update the **Input Parameters** list where necessary.
4. (Optional) If you are testing queries with input parameters, manually assign a value for each parameter in the **Input Parameters** list.
5. Click the **Test All** button to start testing all queries in the capture system.

The test results appear in the result message area. If the query execution is successful, the number of rows or zero output is displayed.

The last entered values for each input parameter are saved with the query. You can reopen the query and test it with the values saved in the last session.

### 3.7.4 Query text file format

A text file query can fetch data from a formatted text file whose structure meets the following requirements:

- **File header:** The first line in the file defines field names delimited by a comma. A field name must be one word, maximum of 64 characters, and must contain *ASCII* alphanumeric characters only. Spaces between field names are not used.
- **File body:** Each line next to the file header is considered a data record in which field values are delimited by a comma.
  - The number of delimited values in the line is equal to the number of fields in the file header.
  - Empty (NULL) values are denoted with two commas in a row with no data between them.
  - Strings in a delimited text can be enclosed in double quotation marks (“”).
  - No blanks can occur before or after delimited values. Spaces (if any) are considered part of the field's value.
  - Each data record starts with a new line.

“Formatted text file: example” on page 74 provides examples for formatted text files.

**Table 3-14: Formatted text file: example**

Header	EmployeeID,ContractID,LoginID,ManagerID,Title,BirthDate,Gender,HireDate,SalariedFlag,SickLeaveHours,ModifiedDate
Line 1	1,14417807,adventure-works \guy1,16,Production Technician - WC60,5/15/1972 0:00,M,7/31/2006 0:00,FALSE,30,7/31/2004 0:00
Line 2	2,253022876,adventure-works \kevin0,6,Marketing Assistant,6/3/1977 0:00,M,2/26/1997 0:00,FALSE,41,7/31/2004 0:00
Line 3	3,509647174,adventure-works \roberto0,12,Engineering Manager,12/13/1964 0:00,M,12/12/1997 0:00,TRUE,21,7/31/2004 0:00

## Chapter 4

# Designing a CaptureFlow

## 4.1 Getting started

This section introduces the basics of CaptureFlow design in graphics mode and the role of the XPP file in this process. Also, this section presents the CaptureFlow Designer tool for process design and provides comparative analysis of this tool with other Intelligent Capture tools used for the development and management of capture processes

### 4.1.1 What is CaptureFlow Designer?

The CaptureFlow Designer tool is used to create capture processes in graphics mode, managing process files in development, and deploying capture processes in the production environment.

#### 4.1.1.1 Features

CaptureFlow Designer is a graphical development environment for designing capture processes (also referred to as *CaptureFlows*). The list of key features includes:

- **Visualized process design.** The user interface helps you design CaptureFlows in graphics mode. You can create a CaptureFlow chart by simply dragging *steps* and *routing elements* from the tools palette to the chart. The layout of the chart is formed automatically, the design issues are immediately reported. Right on the chart, you can perform the following actions:
  - configure steps
  - define conditional routing between the branches
  - define department-based routing of the step's tasks
  - define the transfer of data between steps
  - add scripting to steps
- **Extendable set of modules.** CaptureFlow Designer ships with a collection of client modules developed by OpenText. In addition, you can create *custom modules* using the provided *API* and **add them as steps on the palette** of CaptureFlow Designer.
- **Custom values support.** In addition to IA values of the process, you can define *custom values* of standard types in CaptureFlow Designer. New custom values are saved in *MDF* files automatically.
- **Expressions support.** When designing a CaptureFlow, you can use IA value data paths and expressions to specify the step parameters and routing conditions.

CaptureFlow Designer provides the expressions editor in which you can build an expression correctly using autocompletion.

- **Integrated script editor.** When adding a script to a step, you can create the script code (C# or VB.NET) in the script editor that is called from CaptureFlow Designer. Scripting may be required for complex data processing and data transfer to other steps.
- **Upload to the server.** You can install the designed CaptureFlow on the server from CaptureFlow Designer. The same CaptureFlow chart can be used for installing multiple CaptureFlow instances under unique names.
- **Step setup.** If a CaptureFlow is designed on your machine and installed on the server, you can set up steps of that CaptureFlow in CaptureFlow Designer. Setup requires that the module of that step is installed on your machine.
- **Debugging script code.** If you create a batch from your CaptureFlow in production, you can debug the script code of the CaptureFlow. Debugging is started from CaptureFlow Designer and performed in the script editor. You can trace the execution of all scripts using breakpoints and watchers in the code.

### Related Topics:

[“Adding custom modules to the palette” on page 26](#)

#### 4.1.1.2 Opening CaptureFlow Designer

To open CaptureFlow Designer, click the **CaptureFlow** icon on the toolbar. **Open** an existing CaptureFlow or **create** a new one. The CaptureFlow opens in CaptureFlow Designer ([Figure 4-1](#)).

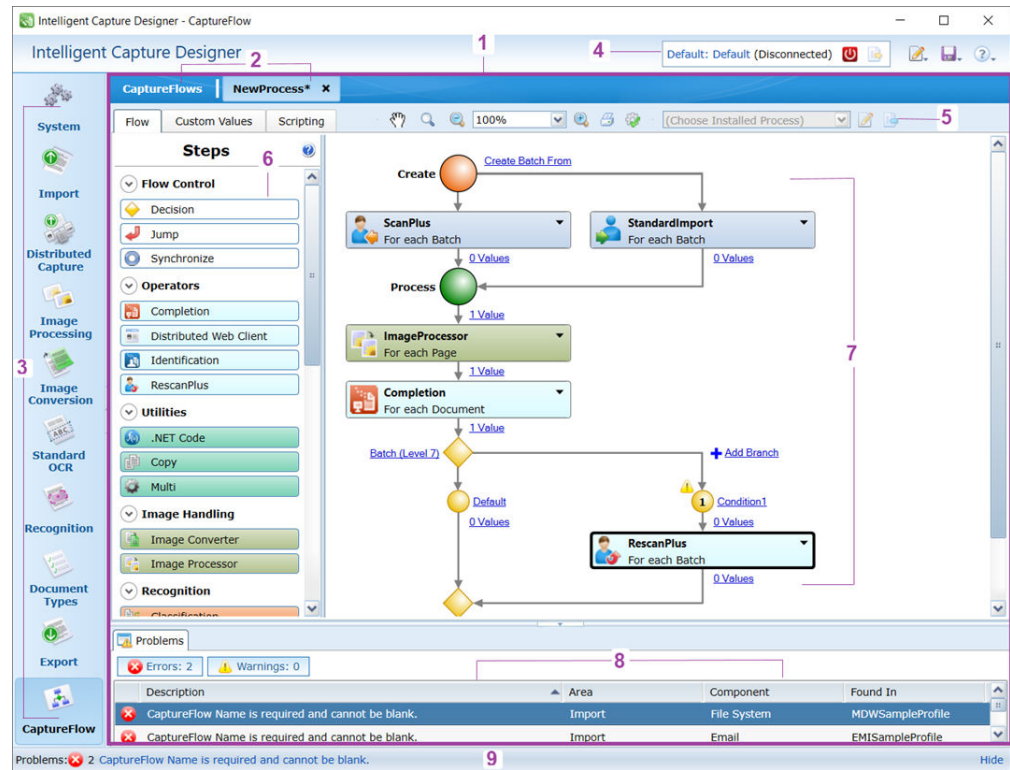
### Related Topics:

[“Creating a CaptureFlow” on page 92](#)

[“Opening a CaptureFlow” on page 92](#)

#### 4.1.1.3 User interface

The user interface of CaptureFlow Designer includes the following areas and elements:



**Figure 4-1: CaptureFlow Designer user interface**

“CaptureFlow Designer: user interface elements” on page 77 provides a description of the CaptureFlow Designer user interface.

**Table 4-1: CaptureFlow Designer: user interface elements**

ID	Area	Description and included elements
1	CaptureFlow Designer area	Includes the following areas: <ul style="list-style-type: none"> <li>CaptureFlow tabs: see (2).</li> <li>CaptureFlow Designer toolbar: see (5).</li> <li><b>Steps</b> panel, or palette: see (6).</li> <li><b>Design</b> panel, or canvas: see (7).</li> <li><b>Problems</b> panel: see (8).</li> </ul>

ID	Area	Description and included elements
2	CaptureFlow tabs	<p>The following tabs are displayed:</p> <ul style="list-style-type: none"> <li>• <b>CaptureFlow List:</b> displays the list of CaptureFlows (XPP files) available on your machine. Every item in this list opens the relevant CaptureFlow chart in the <b>&lt;CaptureFlow_name&gt;</b> tab.</li> <li>• <b>&lt;CaptureFlow_name&gt;:</b> displays a CaptureFlow chart already opened for editing.</li> </ul>
3	Toolbar	<p>Displays the tool buttons. To open CaptureFlow Designer, click the <b>CaptureFlow</b> button.</p>
4	Command bar	<p>Displays the following elements (from left to right):</p> <ul style="list-style-type: none"> <li>• <b>Server connection panel:</b> displays the server name, the connection status (<b>Connected/Disconnected</b>), the <b>Connect</b> button, and the <b>Upload</b> button.</li> <li>• <b>Edit</b> menu button: displays shortcuts of the Clipboard operations.</li> <li>• <b>Save</b> menu button: displays shortcuts of the save operations.</li> <li>• <b>Help</b> button: launches the help system.</li> </ul>

ID	Area	Description and included elements
5	CaptureFlow Designer toolbar	<p>Displays the shortcuts of the following CaptureFlow Designer tools and commands (from left to right):</p> <ul style="list-style-type: none"> <li>• <b>Flow:</b> opens the canvas with the currently selected CaptureFlow chart.</li> <li>• <b>Custom Values:</b> opens the <b>Custom Values</b> panel in which you can add custom values.</li> <li>• <b>Scripting:</b> opens the <b>Scripting</b> panel with the embedded script editor in which you can design scripting for the currently opened CaptureFlow chart.</li> <li>• <b>Pan</b> button: enables or disables the panning mode for the currently selected CaptureFlow chart. If you toggle this button, the chart moves right/left and up/down after the mouse cursor. To disable panning, toggle off the button or press <b>ESC</b>.</li> <li>• <b>Search</b> button: displays the <b>Search</b> panel next to the <b>Problems</b> panel (8).</li> <li>• <b>Zoom Out/Zoom In</b> buttons: diminish/enlarge the chart in the design area. You can also do it with a mouse wheel, keeping the <b>CTRL</b> button pressed.</li> <li>• <b>&lt;100&gt;%</b> combo box: Sets the size of the chart within 25% and 200%, or scales it to fit in the design area (7) by width or entirely. Entering a custom value within the allowed range and</li> </ul>

ID	Area	Description and included elements
		<p>pressing <b>Enter</b> scales the chart accordingly.</p> <ul style="list-style-type: none"> <li>• <b>Print</b>: prints the CaptureFlow; you could set up print options such as paper size and whether to print page numbers and preview the printout.</li> <li>• <b>Compile</b>: compiles the CaptureFlow and its scripting.</li> <li>• <b>(Install New Process)</b> combo box: displays the <b>(Install New Process)</b> button and the list of installed processes that were created from the currently opened CaptureFlow. Select the item to do one of the following: <ul style="list-style-type: none"> <li>– <b>(Install New Process)</b> button: compiles the currently opened CaptureFlow and installs a new capture process with the specified name on the server.</li> <li>– <i>&lt;Process instance name&gt;</i>: select the process before you install it, set up its steps, debug its scripting, or uninstall the process with the <b>Edit Process</b> button.</li> </ul> </li> <li>• <b>Edit Process</b> button: provides options to install a copied process, edit setup settings of an installed process, or uninstall a process from the server.</li> <li>• <b>Download settings</b> button: downloads from the server the step setup configuration made on a different machine.</li> </ul>

ID	Area	Description and included elements
6	Steps panel	<p>Displays the primitives that can be dragged to the CaptureFlow chart. The following sections can be expanded or collapsed:</p> <ul style="list-style-type: none"> <li>• <b>Flow Control:</b> displays the primitives that affect the execution order of steps in the process.</li> <li>• <b>Operators:</b> displays the operator managed tools.</li> <li>• <b>Utilities:</b> displays the Intelligent Capture enabled utilities.</li> <li>• <b>Image Handling:</b> displays the modules for image processing.</li> <li>• <b>Recognition:</b> displays the modules for image recognition.</li> <li>• <b>Input Output:</b> displays the modules for data capture and data export.</li> <li>• <b>Export (Enterprise):</b> displays the modules for data export to external data management systems.</li> <li>• <b>Custom:</b> displays the customer added modules. This section is hidden by default. It appears when a custom module is added.</li> </ul> <p>The modules deprecated in the current Intelligent Capture release are marked as <b>(Deprecated)</b>.</p>
7	Design area, or canvas	<p>Displays a CaptureFlow chart opened for editing. The <b>elements</b> can be added to the chart from the <b>Steps</b> panel (6).</p>

ID	Area	Description and included elements
8	<b>Problems panel</b>	<p>Appears when the <b>More</b> link is clicked on the <b>Problems</b> bar. Displays error messages generated for the currently used capture system. The messages are displayed in two filters:</p> <ul style="list-style-type: none"> <li>• <b>Critical:</b> displays critical errors, such as system configuration errors and CaptureFlow design and compilation errors. A CaptureFlow chart with critical errors cannot be compiled.</li> <li>• <b>Warnings:</b> displays non-critical issues.</li> </ul>
9	<b>Problems bar</b>	<p>Appears if any configuration, design, or compilation error happens to occur in the currently used capture system. Displays the counter of errors and the text of the first error in the list. The <b>More/Hide</b> link expands and collapses the <b>Problems</b> panel.</p>

#### 4.1.1.4 CaptureFlow Designer versus Process Developer

Intelligent Capture installs with two process design tools on board – CaptureFlow Designer and Process Developer. The following table compares their features and functionality.

**Table 4-2: CaptureFlow Designer vs Process Developer**

Functionality	CaptureFlow Designer	Process Developer
Visualization	<p>You design the CaptureFlow chart. You can drag and drop graphical primitives, such as steps, conditional blocks, jumps, and synchronization elements, to the chart. The added elements and data transfer between steps are configured right on the chart.</p>	<p>Graphics design mode is not provided. The process is coded.</p>

Functionality	CaptureFlow Designer	Process Developer
Multiple batch-forming steps	Supported. You can add several batch-forming steps, of which only one will be executed. You cannot add the same batch-forming module to the chart twice.	Supported. You can add several batch-forming steps, of which only one will be executed. The same module can be added more than once.
Capture process files	<p>The CaptureFlow chart, scripting, information about the installed CaptureFlow instances, and the setup settings of steps (modules) of the installed CaptureFlows are packed in the XPP file.</p> <p>IPP and IAP files are generated when you install a CaptureFlow on the server. They do not require manual handling.</p>	<p>The process flow is stored as an IPP file.</p> <p>A compiled IPP file results in the IAP (executable) file.</p>
Custom values	<p>You add custom values to a CaptureFlow in CaptureFlow Designer. You do not need to edit MDF files.</p> <p>Custom values are restricted to standard value types and cannot use user-defined object types.</p>	
MDF files	At design time, if CaptureFlow Designer finds a discrepancy between an MDF file installed on the designer's machine and the same MDF stored in the XPP file, it uses the installed MDF and updates the XPP file accordingly.	At design time, you have the option to use MDF files that are embedded within an IPP file or MDF files installed on your machine.
Task execution	Parallel task execution for a batch node is not supported. Tasks are serialized. By choosing a task level lower than 7, you can perform parallel execution among different nodes.	Supports parallel task execution for a given batch node.
Value assignments	Value assignments have intelligent iterations and do not require additional "For loop" logic to assign values at a lower level.	Cycle coded manually.

Functionality	CaptureFlow Designer	Process Developer
Value access	Value access is restricted to batch data values. Trigger values are handled automatically by the graphical flow logic. Setup values are handled by running the module in setup mode and can also be accessed by the .NET Code module.	Access to values, trigger values, and setup values is handled directly in code.
Conditions and expressions	Uses a generic set of operators and functions to define conditions and expressions.	Uses VBA operators and functions.

#### 4.1.1.5 CaptureFlow Designer versus Intelligent Capture Administrator

XPP files can be managed using one of the following tools – CaptureFlow Designer or Intelligent Capture Administrator.

Using CaptureFlow Designer is preferable because this tool supports the integrity of the XPP file during the CaptureFlow lifetime. In particular, CaptureFlow Designer updates the XPP file accordingly:

- when installing a CaptureFlow on the server,
- when uninstalling a CaptureFlow from the server,
- when setting up a CaptureFlow step.

Intelligent Capture Administrator does not update the XPP file when these events happen.



#### Caution

Avoid using both CaptureFlow Designer and Intelligent Capture Administrator when managing the same XPP file.

### 4.1.1.6 Multi-language support

The designer's machine that hosts CaptureFlow Designer must use the same code page as the server to which the CaptureFlows will be installed. Most textual elements in a CaptureFlow chart are restricted to the language for the system code page. If multiple languages must be used during the CaptureFlow design then the following rules apply:

- CaptureFlow, step, IA values (including custom values), and department names must conform to the language for the system code page and contain alphanumeric or underscore ( `_` ) characters, and the first character has to be alphabetic. The first character of a step name must be an ASCII character from the a-z or A-Z range. The allowed field length depends on the feature. An error message displays **Problems** status bar when the maximum number of characters is exceeded.
- Expression and condition text boxes in the **Assign Values**, **Condition**, and **Step Departments** dialog boxes cannot contain multilingual literals.
- When multilingual literals are needed, use the **Custom Values** panel to define a literal as a custom value. For example, define a custom value with the name "JapaneseChar" as a string data type. In the **Initial Value** field, add a character in the Japanese language within double quotes. Then reference `<CustomValues:<level>.JapaneseChar>` in the expression and condition text boxes.
- Comment text boxes located in the **Step Properties** and **Condition** dialog boxes do not have any restrictions and can contain any combination of characters.

### 4.1.2 What is an XPP file?

CaptureFlow Designer saves each CaptureFlow chart in a separate *XPP* file. The root name of this file matches the CaptureFlow name specified in CaptureFlow Designer.

The *XPP* file is stored on the designer's machine in the **working directory**. By default, the path is `%USERPROFILE%\Documents\<product_name_and_version number>\GlobalData\XPP`.

An XPP file encapsulates information about the CaptureFlow structure, IA values, all installed CaptureFlow instances, and all step setup settings. The XPP file collects this information during the CaptureFlow lifetime.

The following CaptureFlow Designer events update the XPP file.

**Table 4-3: XPP file updates**

Information	When added to an XPP file	Comments
CaptureFlow chart structure (all steps and routing elements)	At design time: when the chart is created and saved; when the chart is modified and saved.	<p>When a chart is saved for the first time, CaptureFlow Designer creates an XPP file and assigns a process version ID to it. Unless you install a process, each saving only updates the XPP version ID locally on your designer's machine in the <b>working directory</b>.</p> <p>To identify a CaptureFlow current version ID, open the <code>&lt;CaptureFlow_name&gt;</code> tab, then while holding <b>ALT+V</b>, switch to the <b>CaptureFlows</b> list tab. The new <b>Version ID</b> column appears on the <b>CaptureFlows</b> tab.</p>
IA value assignments between steps	At design time: when IA value assignments are added and saved.	
Source code files (CS or VB) with step scripting	At design time: when scripting is added to a step or modified, and the CaptureFlow is saved.	
Custom IA values	At design time: when custom values are added and saved.	
Module IA values (MDF file)	At design time: when a step is added to the chart and the CaptureFlow is saved.	
A CaptureFlow installation on the server	Every time a capture process is installed on the server from the CaptureFlow chart (XPP).	Every time the current version of XPP is used to <b>install a process instance</b> , the copy of this particular version of XPP file is saved in a separate folder for each installed process on the server in the <code>\IAS\process</code> directory.
The step setup information for each installed CaptureFlow instance	Every time a step in any installed process is set up.	

Due to its encapsulated nature, an XPP file can be transferred to other client machines at any stage of the CaptureFlow development or testing. However, the content of the XPP file cannot be opened by clicking the file in the system folder. A CaptureFlow chart must be opened in the context of its capture system, and this can only be done in Designer.

When you install an instance of a CaptureFlow on a test server and start testing, it is important to keep the local CaptureFlow (the chart, IA values, and scripting) unchanged. This is important for the following reasons:

- When you run script code debugging, the batch is created from the installed CaptureFlow, while the assembly is compiled from the local XPP file and scripting is executed on the designer's machine. The testing results are considered reliable only when the local and installed CaptureFlows are fully identical.
- When you upload the tested capture processes on the production server, you actually reinstall the CaptureFlows and set their steps from the local XPP file. Keeping the local CaptureFlow unchanged guarantees that you have tested and deployed the same capture process.

### 4.1.3 High-level steps to a production CaptureFlow

“High-level steps for creating a CaptureFlow” on page 87 includes the high-level steps for creating a capture process in CaptureFlow Designer and deploying it in the production environment.

**Table 4-4: High-level steps for creating a CaptureFlow**

Stage	Job	Related topics
1	Creating a CaptureFlow chart.	“Designing a CaptureFlow chart” on page 96
2	Compiling a CaptureFlow and resolving design issues.	“Compiling a CaptureFlow” on page 443
3	Installing an instance of the CaptureFlow on the test server.	“Installing a CaptureFlow” on page 443

Stage	Job	Related topics
4	(Optional) Creating profiles, document types, and recognition projects for the following steps (if included in the CaptureFlow): <ul style="list-style-type: none"> <li>• <b>Standard Import</b></li> <li>• <b>Distributed Capture</b></li> <li>• <b>Image Processor</b></li> <li>• <b>Image Converter</b></li> <li>• <b>Classification</b></li> <li>• <b>Identification</b></li> <li>• <b>Collector</b></li> <li>• <b>Extraction</b></li> <li>• <b>Completion</b></li> <li>• <b>Standard Export</b></li> <li>• <b>Standard OCR</b></li> </ul>	For more information, see the following: <ul style="list-style-type: none"> <li>• “Designing file and email import” on page 147.</li> <li>• “Designing a profile for the Web Client” on page 175.</li> <li>• “Designing image processing” on page 189.</li> <li>• “Designing image conversion” on page 229.</li> <li>• “Designing a profile for optical character recognition processing” on page 259.</li> <li>• “Designing a document classification or data extraction project” on page 329.</li> <li>• “Designing a document type” on page 341.</li> <li>• “Designing a profile for exporting images and data” on page 277.</li> </ul>
5	Setting up the CaptureFlow steps.	“Setting up CaptureFlow steps” on page 448
6	(Optional) Creating a batch and debugging scripting in production mode.	“Debugging scripting” on page 142
7	Deploying the tested CaptureFlow in the production environment.	“Uploading a CaptureFlow” on page 453
8	(Optional) Repeating setup for the CaptureFlow steps in production. This refers to the modules that keep their global setup settings on the test server. The modules include: ScanPlus, Web Services Input, Standard Import, and eIndex.	“Setting up CaptureFlow steps” on page 448

### 4.1.4 Quick start: creating a simple process

This topic shows you how to create a simple process named `MyProcess.xpp`. In this process, pages are passed from one step to another. Conditional routing is not used.

#### To create a process:

1. Click **CaptureFlow**.
2. In the **CaptureFlow List** tab, click **New** to create a CaptureFlow.
3. In the **Process – New** dialog box, enter `MyProcess` in the **New Process Name** edit box and click **Save**.

CaptureFlow Designer creates the `MyProcess.xpp` file and opens an empty chart. By default, the CaptureFlow chart includes the **ScanPlus** step that will create a batch. At a minimum, a process requires one batch-creating step.

4. Add steps. From the **Steps** panel, drag and drop the following modules to your CaptureFlow chart below the **Process** point:
  - **Image Processor**: This module is located in the **Image Handling** section of the **Steps** panel.
  - **Completion**: This module is located in the **Operators** section of the **Steps** panel.
  - **Standard Export**: This module is located in the **Input Output** section of the **Steps** panel.

To drag a module, select it in the **Steps** panel and drag towards the chart, keeping the left mouse button pressed. As you drag, the green arrows on the chart indicate the available locations for the step. When selected as a destination, the arc on the CaptureFlow chart turns green.

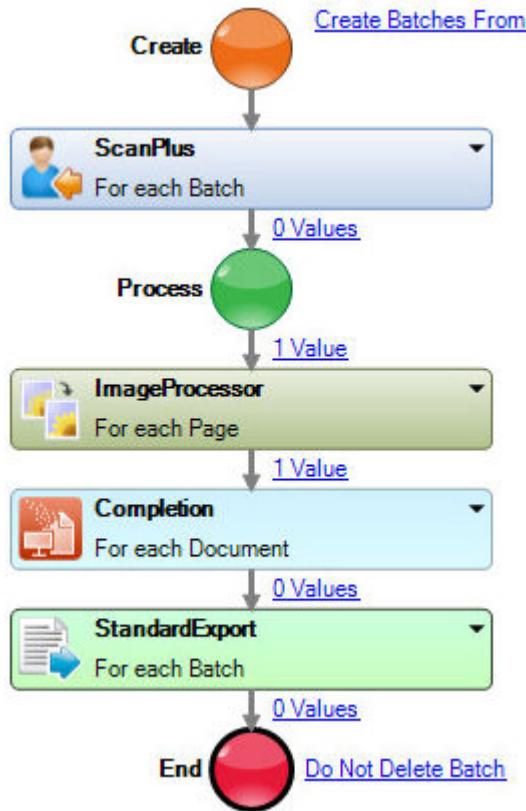
5. When you add steps, the IA value assignments are added automatically. Click each **1 Value** link above the step and view the generated IA value assignments in the **Assign Values** dialog as shown in [“Assign values fields” on page 89](#).

**Table 4-5: Assign values fields**

Step	Assign to	Value
<b>ImageProcessor</b>	ImageProcessor:0.InputImage	ScanPlus:0.OutputImage
<b>Completion</b>	Completion:0.Image	ImageProcessor:0.OutputImage

6. Define the end of the process. Click **Delete Batch** and select **Do Not Delete Batch**.

The structure of the process is complete and looks as follows:



**Figure 4-2: Process chart with steps**

7. Install an instance of your process to the server. To do this, expand the list of installed processes and select the **(Install New Process)** option.
8. In the **New Installed Process** dialog, enter the unique name of the process instance in the **Process Name** edit box. Leave the **Apply module settings** option set to **Default**. Click **Save**.

This action saves all changes to the XPP file, starts the compilation of a new process, and uploads the compiled process to the connected server. If the new process has been successfully installed, you can see its name in the list of installed processes.

The **Problems** panel displays a warning that step setup may be incomplete.

9. Before you proceed to setting up process steps, you need to create and upload on the server the profiles for the following steps:
  - **ImageProcessor:** Create an image processing profile as described in *“Designing image processing”* on page 189. You can also use a sample image processing profile from `... \<capture system name>\GlobalData\ ImageProcessing`.

- **StandardExport:** Create an export profile as described in [“Defining an export profile” on page 278](#). You can also use a sample export profile from `... \<capture system name>\GlobalData\Export`.
10. Set up steps in the new process. Right-click a step on the CaptureFlow chart and select **Module Settings** from the context menu.

This menu command runs the module in setup mode. Specify the required setup settings in the module’s setup window:

- ScanPlus: Leave the default settings.
- ImageProcessor: Select the image processing profile and save the settings.
- Completion: Specify the setup settings as described in *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*.
- StandardExport: Select the export profile and save the settings.

The module setup data is saved on the server and in the XPP file from which the process has been installed.

The process is created and ready for testing.

## Related Topics

[“Designing a CaptureFlow chart” on page 96](#)

## 4.2 Managing CaptureFlows

Click the **CaptureFlow** icon on the toolbar. The **CaptureFlows** list displays the CaptureFlows that are available in the currently selected *capture system*. The table in this section describe the operations on CaptureFlows that you can perform from the **CaptureFlows** list.

**Table 4-6: CaptureFlow list tab: tabs, columns, buttons**

Tabs, columns, buttons	Description
Tabs	Displays the <b>CaptureFlows</b> list and a specific tab for each opened CaptureFlow chart. Clicking a tab opens the list of CaptureFlows or a particular CaptureFlow chart. Clicking <b>X</b> on a CaptureFlow tab closes the chart.
<b>CaptureFlow</b> column	Displays the CaptureFlow names sorted alphabetically.
<b>Version ID</b> column	Displays CaptureFlow version ID. By default, the column is hidden. To show the <b>Version ID</b> column in the list, open any <code>&lt;CaptureFlow_name&gt;</code> tab, then while holding <b>ALT+V</b> , switch to the CaptureFlows list tab.

Tabs, columns, buttons	Description
Description column	Displays the description of the CaptureFlow. Click in this field to edit the description.
File Time column	Displays the time and date of the last change based on the local machine settings.
Open button	Opens the selected CaptureFlow chart.
New button	Creates a new CaptureFlow and opens a chart with one default step ( <b>ScanPlus</b> ) in a separate tab.
Delete button	Deletes the selected CaptureFlow from the capture system and from the <b>CaptureFlows</b> list. The XPP file with the deleted CaptureFlow is removed from the disk.

## 4.2.1 Creating a CaptureFlow

Adding an new CaptureFlow chart to the **CaptureFlow List** tab is the first step in creating a CaptureFlow.

### To add a CaptureFlow:

1. Open the **CaptureFlow List** tab and click **New**.
2. Enter the CaptureFlow name in the **New CaptureFlow Name** edit box of the **Process - New** window. When entering the name, follow the [naming convention on page 481](#) for CaptureFlows.
3. Press **Save**. CaptureFlow Designer opens a new CaptureFlow chart for editing.

## 4.2.2 Opening a CaptureFlow

To open a CaptureFlow for editing, use CaptureFlow Designer. You cannot open a CaptureFlow by double-clicking an XPP file in the system folder because the operating system cannot associate the XPP contents with any capture system.



**Note:** If the IA values for a module have changed, a warning is displayed in the CaptureFlow steps. When you save the CaptureFlow, it is upgraded after which you can use the new IA values.

### To open a CaptureFlow for editing:

1. Open the **CaptureFlow List** tab and select the required CaptureFlow.



**Note:** A yellow triangle icon near a CaptureFlow indicates the XPP file does not have a **process version ID**. This CaptureFlow is valid and can be opened for editing after a conversion. For details, see [“Opening a non-versioned CaptureFlow” on page 93](#).

Conversely, you should avoid opening an XPP with a process version ID in the version of CaptureFlow Designer that does not support the process versioning functionality. Even if the XPP file can be opened and used for installing a process from it, in future it might lead to unexpected behavior during batch processing.

2. Double-click the selected CaptureFlow or click the **Open** button.

The CaptureFlow chart opens for editing in CaptureFlow Designer in a separate <CaptureFlow\_name> tab.

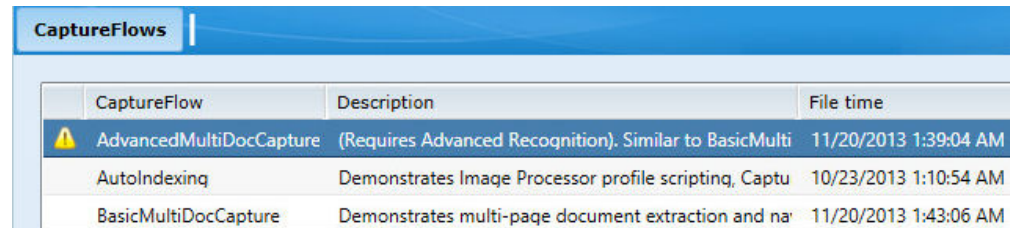
### Related Topics:

[“What is an XPP file?” on page 85](#)

[“Opening a non-versioned CaptureFlow” on page 93](#)

## 4.2.3 Opening a non-versioned CaptureFlow

Sometimes you need to open CaptureFlows created in earlier versions of Designer. In cases when an old version of Designer does not support the process versioning functionality, the CaptureFlows created in it do not have a **process version ID**. In the **CaptureFlow List** tab, such CaptureFlows are displayed with a yellow triangle icon.



CaptureFlow	Description	File time
⚠ AdvancedMultiDocCapture	(Requires Advanced Recognition). Similar to BasicMulti	11/20/2013 1:39:04 AM
AutoIndexing	Demonstrates Image Processor profile scripting, Captu	10/23/2013 1:10:54 AM
BasicMultiDocCapture	Demonstrates multi-page document extraction and na	11/20/2013 1:43:06 AM

**Figure 4-3: Capture flows without process version ID**


### Converting a CaptureFlow to the versioned format

Prior to editing a non-versioned CaptureFlow in CaptureFlow Designer, you need to convert the XPP file to the versioned format. When opening a non-versioned CaptureFlow, the prompt dialog displays for you to select and confirm a conversion option.



**Note:** Ensure the CaptureFlow you are going to convert has no running batches on the server. After the conversion is complete, non-versioned batches cannot be processed based on the new versioned XPP file.

Depending on the version of the CaptureFlow Designer in which the original XPP file has been created, the prompt dialog might contain one option or a set of options from the following table.

Options	Description
<p>Creating a copy of this CaptureFlow in a new format</p>	<p>A new CaptureFlow with a unique name is created and assigned a process version ID. The copy has the original CaptureFlow's structure and includes its IA values, step setup settings, and all installed CaptureFlow instances.</p> <p> <b>Note:</b> After converting, new batches are created versioned based on this process. However, the copied process can still have old non-versioned batches. Such process cannot be <b>deleted</b> until these batches exist.</p> <p>The original CaptureFlow is not modified and remains in the <b>CaptureFlow List</b> tab marked with a triangle icon. It can be deleted as appropriate.</p>
<p>Upgrading this CaptureFlow</p>	<p>The original non-versioned CaptureFlow is converted to the new format and assigned a process version ID. The upgraded CaptureFlow inherits the structure, IA values, step setup settings, and all installed CaptureFlow instances from the original CaptureFlow.</p>



**Note:** When you need to convert a non-versioned CaptureFlow that implements scripting, take the following into consideration:

- The AssemblyInfo file is not converted and in case it includes your custom code, this code will be lost. You are responsible for moving custom code from AssemblyInfo to the appropriate script file created for the CaptureFlow step.
- The original event handlers are copied to the new CaptureFlow as is, without changes. You can **open the converted code** in the embedded script editor to verify the conversion results. See the list of available **event handlers** and **insert** the new methods into the converted code, if necessary.

## Opening a non-versioned CaptureFlow for editing

**To open a non-versioned CaptureFlow for editing:**

1. Open the **CaptureFlow List** tab and select a CaptureFlow marked with a yellow triangle.
2. Double-click the selected CaptureFlow, or click the **Open** button.
3. In the prompt dialog, select whether you want to copy or upgrade this CaptureFlow. Click **OK**.

4. In the **Save As** dialog box, specify the name of the new/upgraded CaptureFlow following the [naming convention on page 481](#). Click **Save** to save the new/upgraded CaptureFlow.



**Note:** In some cases, a message might display to inform you about external script files which are in a conflict with the CaptureFlow Designer scripting. The reason is that the new XPP file encapsulates scripting rather than keeps it in the external files. You can confirm the deletion of the unnecessary external files from the disk. If you click **Cancel**, this message will display every time you open the XPP file.

5. The converted CaptureFlow chart opens for editing in CaptureFlow Designer in a separate `<CaptureFlow_name>` tab.

#### **Related Topics:**

[“What is an XPP file?” on page 85](#)

[“Deleting a CaptureFlow” on page 95](#)

[“Developing scripting” on page 126](#)

[“Inserting script methods” on page 140](#)

### **4.2.4 Deleting a CaptureFlow**

You can use CaptureFlow Designer to delete a corrupted or unnecessary CaptureFlows.

#### **To delete a CaptureFlow:**

1. Close each CaptureFlow before you delete it.
2. Open the **CaptureFlow List** tab and select one CaptureFlow or several CaptureFlows you want to delete.
3. Click the **Delete** button and confirm the deletion in the prompt dialog box.

The deleted CaptureFlow disappears from the **CaptureFlow List** tab, gets deleted from the capture system and physically removed from the disk. The confirmed deletion of a CaptureFlow cannot be reverted by the **Undo** command.

#### **Related Topics:**

[“Deleting a CaptureFlow instance” on page 452](#)





## 4.3 Designing a CaptureFlow chart

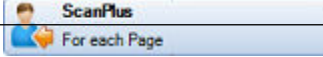


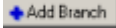
The initial step in creating a CaptureFlow is designing its visualized model – a *CaptureFlow chart*. The topics in this section describe how to design a CaptureFlow chart.





### 4.3.1 What is a CaptureFlow chart?

A CaptureFlow chart is a visualized model of a capture process. It includes the tools and icons provided in “[CaptureFlow chart: tools and icons](#)” on page 96.

**Table 4-7: CaptureFlow chart: tools and icons**

Name	 con	Description
<b>Create</b>		<p>The <b>Create</b> point starts the CaptureFlow chart. This element cannot be moved or deleted from the chart.</p> <p>The section between the <b>Create</b> and the <b>Process</b> points contains the steps that create or modify the batch structure. The first step located right below the <b>Create</b> point is the batch creating step.</p> <p>A chart can include multiple batch creating steps. In this case, the <b>Create</b> point has several branches, one per a batch creating step. All of these branches end at the <b>Process</b> point.</p>
<b>Create Batches From</b>	N/A 	Click this link to display a list of steps that create a batch.
<b>Process</b>		<p>The <b>Process</b> point starts the batch processing area of the CaptureFlow chart. This element cannot be moved or deleted from the chart.</p> <p>The section between the <b>Process</b> and the <b>End</b> points includes the steps that handle the batch data.</p>

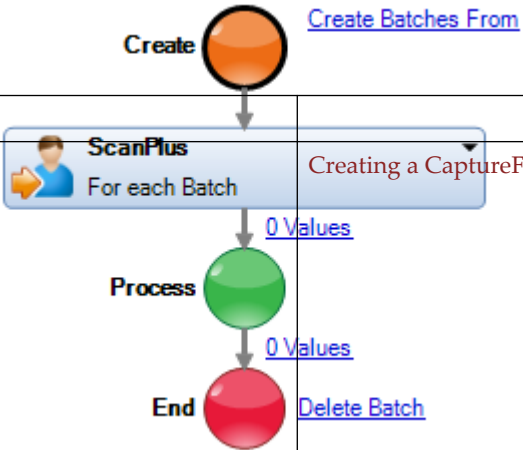
Name	Tool/Icon	Description
Arc	N/A	Arcs represent transitions between elements on a CaptureFlow chart. The elements of a process are executed in the order they are connected with arcs.
Step icon		A <b>Step</b> element represents a particular module indicated by the step name. When you add a step on your CaptureFlow chart, you specify the module that will be triggered at a given point of the process.
<b>Decision</b>		<p>A <b>Decision</b> block is a routing element that includes at least two branches, a default one and a conditional one. You can add more conditions using the <b>Add Branch</b> link.</p> <p>Branches include steps and other elements.</p> <p>Conditional branches are ordered. A conditional branch is executed if its condition evaluates to <b>True</b>. If all of the conditional branches evaluated to <b>False</b>, the default branch is executed.</p>
<b>Condition</b>		A condition starts a branch in the <b>Decision</b> block. Conditions evaluate to either <b>True</b> or <b>False</b> . The capture process uses a conditional expression to decide whether a branch is selected or not.
<b>Add Branch</b>		The <b>Add Branch</b> link adds a branch to a <b>Decision</b> block for additional conditional routing. The <b>Condition&lt;N&gt;</b> label increments as you add branches.

Name	Tool/Icon	Description
<b>Jump</b>		The <b>Jump</b> element used in a <b>Decision</b> block redirects the execution of a process back to an earlier point on the CaptureFlow chart.
<b>Synchronize</b>		The <b>Synchronize</b> element helps you synchronize the batch processing at a given point on the CaptureFlow chart. The level of this element indicates that all preceding tasks below this level must be completed at the synchronization point before the next task is taken for processing.
<b>Delete Batch</b>	N/A  	This link opens the menu for process end options: <ul style="list-style-type: none"> <li>• <b>Delete Batch:</b> used on the chart by default.</li> <li>• <b>Do Not Delete Batch:</b> if selected, this option replaces the default one on the CaptureFlow chart.</li> </ul>
<b>End</b>		The <b>End</b> point indicates the end of the process. This element cannot be moved or deleted from the chart.

### 4.3.2 High-level steps for creating a CaptureFlow chart

Designing a CaptureFlow chart includes several stages as provided in “[High-level steps for creating a CaptureFlow chart](#)” on page 99.

Table 4-8: High-level steps

Step		
1	 <p>The diagram illustrates the high-level steps of a CaptureFlow chart. It starts with a 'Create' step (orange circle) labeled 'Create Batches From'. This leads to a 'ScanPlus' step (blue box with a person icon) labeled 'For each Batch', which is associated with 'Creating a CaptureFlow'. Below this is a 'Process' step (green circle) labeled '0 Values'. This leads to an 'End' step (red circle) labeled 'Delete Batch', also associated with '0 Values'.</p>	
2	<p>(Optional) Define the batch creating steps. Set the trigger level for each step.</p> <p>You can replace <b>ScanPlus</b> by a different step or have several batch creating steps in parallel.</p>	Adding Batch Creating Steps
3	<p>(Optional) Select the batch processing steps between the <b>Create</b> point and the <b>Process</b> point.</p> <p>Set the trigger level for each step. Specify error handling for each step.</p>	Adding a Batch Processing Step
4	<p>(Optional) Assign a step to a department if the tasks created for this step must be routed to a certain machine(s) running in a department.</p>	Assigning a Department to a Step
5	<p>(Optional) Add a decision where you need to split a chart into two or more branches with steps and every time execute a particular branch that is selected conditionally.</p> <p>Specify conditions, add steps to each branch.</p> <p>Set trigger levels for all included steps.</p>	Adding a Decision Block

Step	Job	Link
6	(Optional) Specify custom values and use them in conditions.	<a href="#">“Defining a custom value for a CaptureFlow” on page 122</a>
7	(Optional) Add a jump if you need to return to a previous condition or a step and repeat the processing.	<a href="#">Adding a Jump</a>
8	(Optional) Add synchronization between elements where the next step works on a higher level than the preceding one.  Set the synchronization level equal to the level of the step that executes next.  Synchronization puts the next step on hold until all preceding lower-level tasks are completed and input data is collected for the next step.	<a href="#">Adding Synchronization</a>
9	Configure value assignments between steps.  If necessary, add custom values.	<a href="#">Configuring IA Value Assignments</a>
10	(Optional) Add scripting to a step if value assignment required preliminary calculation, data conversion, or other complex processing.	<a href="#">“Adding scripting” on page 136</a>
11	Configure the end of the CaptureFlow.	<a href="#">Configuring the End of a Process</a>
12	Save the CaptureFlow chart.	<a href="#">Saving a CaptureFlow</a>

### 4.3.3 Clipboard Operations

You can edit the CaptureFlow chart using the drag-and-drop operations and the standard Clipboard operations, such as copy, cut, paste, delete, undo, and redo. You can do it using menu commands or standard Windows keyboard shortcuts. When editing a chart, you can select only one element at a time.

[“Clipboard operations supported in a CaptureFlow chart” on page 101](#) describes the Clipboard operations supported in a CaptureFlow chart.

**Table 4-9: Clipboard operations supported in a CaptureFlow chart**

CaptureFlow element	Copy/paste/cut/delete (on a chart and between charts)	Drag-and-drop (on a chart only)	Notes
Text	Yes	No	
Value assignments between elements	Yes	No	
Step (except batch creating steps: <b>ScanPlus, Standard Import, Web Services Input</b> )	Yes	Yes	When you duplicate a step on the same CaptureFlow chart or move it to a different CaptureFlow chart, the scripts created for that step are not copied.  You cannot delete the only batch creating step from the chart. You cannot delete a batch creating step if it has other elements on its branch.  IA value assignments before and after the relocated element are not copied, cut, pasted, or dragged to a new location. They are merged.
Synchronize element	Yes	Yes	
Decision block	Yes	Yes	

CaptureFlow element	Copy/paste/cut/delete (on a chart and between charts)	Drag-and-drop (on a chart only)	Notes
Branch (part of a decision block)	Yes	No	<p>A branch is copied and pasted with all of its included elements. Scripting is not copied with the steps.</p> <p>A default branch is copied or pasted as a conditional branch. A default branch not including steps cannot be copied.</p> <p>To reorder conditional branches in a decision block, click the branch name and then click <b>Earlier</b> (to move to the left) or <b>Later</b> (to move to the right). Moving a branch to the right raises its priority.</p> <p>The default branch cannot be reordered.</p>
Jump	No	No	

### 4.3.4 Adding batch creating steps

By default, a CaptureFlow chart includes **ScanPlus** as a batch creating step. However, a chart can include several such steps in parallel branches. During production, a user creates a batch by triggering any of these steps.

Adding a batch creating step can be accomplished by creating an entirely new step and customizing its properties and scripting, or by copying an existing step and modifying only some of the copied properties or scripting. For more information, see the following:

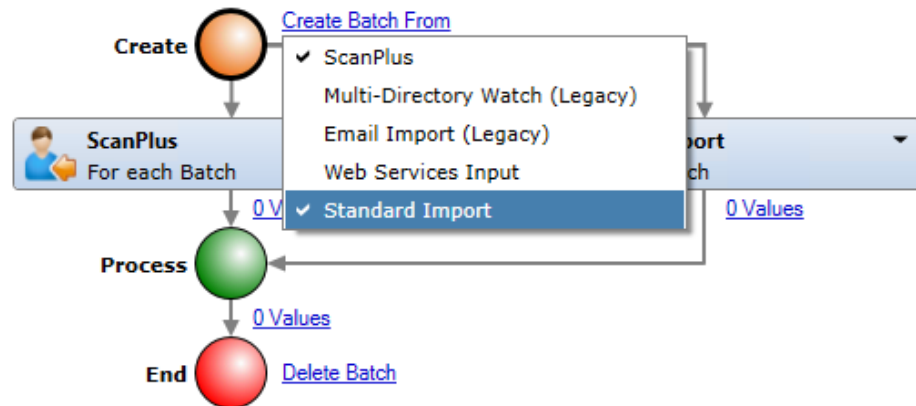
- [“Adding a new batch creating step” on page 102](#)
- [“Copying a batch creating step” on page 104](#)

#### Adding a new batch creating step

**To add a batch creating step:**

1. Open the CaptureFlow chart in CaptureFlow Designer.

2. Click the **Create Batches From** link on top of the chart.
3. Check a step that you want to add to the chart. You can choose from the following options (Figure 4-4):



**Figure 4-4: Step options**

- **ScanPlus:** Receives images from a scanner.
- **Multi-Directory Watch:** Fetches images from a folder.
- **Email Import:** Receives images and data in email messages from the mail server.
- **Web Services Input:** Receives images from the Web as attachments, URL, or Base64 strings. This option requires a client application implemented by a third-party.
- **Standard Import:** Imports image files from directories, email and attachments from an email server, and files and batch node values from the Web Client as batches. The option requires a standard import profile created for the preferred type of files to be captured.

You can choose multiple options. Each step can be added to the chart but once.

4. If necessary, you can add other elements from the **Steps** palette between the batch creating steps and the **Process** point.
5. To delete a batch creating step from the chart, click the **Create Batches From** link and uncheck the option on the menu.



**Note:** You cannot delete the only batch creating step from the chart. You cannot delete a batch creating step if its branch includes other elements.

## Copying a batch creating step

### To copy/paste a batch creating step:

1. In a CaptureFlow, click the step you need to copy and press **CTRL+C**, or use the **Copy** option from the **main toolbar**.
2. Open the CaptureFlow which you need the step to be placed to and press **CTRL+V**, or use the **Paste** option.
3. The copied step is inserted into the placing CaptureFlow with all its properties and scripting. The following may occur:
  - If the CaptureFlow chart already contains a step which stands for the same batch creating module, a user needs to confirm the replacement action. When **Replace** is selected, the existing step is replaced entirely with the copied step.
  - If the CaptureFlow chart already contains a step which stands for any different module but named identically, the copied step is pasted with a new automatically generated name.
  - When a copied step includes scripting, the code refactoring is performed to conform the placing CaptureFlow and new step name after pasting.



### Notes

- The copied step cannot be pasted if its scripting contains syntax errors.
- If the scripting language of the copied step differs from the language of the placing CaptureFlow, the step is pasted without scripting and a warning displays in the **Problems** panel. However, if the CaptureFlow already contains a step which stands for the same batch creating module, the pasting is impossible to perform.

### Related Topics:

[“Adding batch processing steps” on page 104](#)

## 4.3.5 Adding batch processing steps

To define the batch processing flow, add a sequence of steps to your CaptureFlow chart between the **Process** and **End** points.

Each step stands for a particular Intelligent Capture module that implements the batch handling logic. When a CaptureFlow is executed in production, modules are triggered in the order they are defined in the CaptureFlow chart.

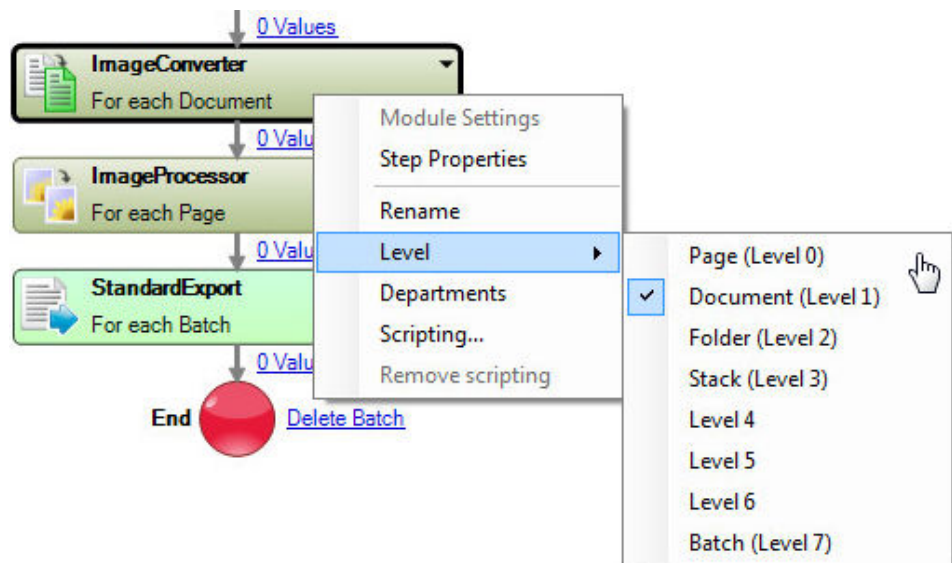
Adding a batch processing step can be accomplished by creating an entirely new step and customizing its properties and scripting, or by copying an existing step and modifying only some of the copied properties or scripting. For more information, see the following:

- “Adding a new batch processing step” on page 105
- “Copying a batch processing step” on page 106

## Adding a new batch processing step

### To add a batch processing step:

1. Select a step on the **Steps** panel.
2. Drag the step to the chart, keeping the left mouse button pressed.  
The green arrows on the chart show where the step can be dropped. The chart arc selected for insertion becomes green.
3. Right-click the step and select **Level**. Set the trigger level from the context menu. This setting specifies the level of the task that can trigger step during production.



**Figure 4-5: Batch processing step level**



**Note:** Some modules are restricted to a particular trigger level. For example, for **Image Converter** you can only assign trigger level 1 (default) or 0.

4. (Optional) Specify error handling for the step. Right-click the step and select **Step Properties**. In the **Step Properties** dialog box, check the box against the required error handling:
  - **Put the task on hold for an administrator to fix it:** (Default) The task is retried up to three times, after which the task is in error and requires an administrator to clear the error and re-trigger the task.

- **Continue on to the next step:** The task continues, the `<ErrorCode>` and `<ErrorText>` IA values on the task level are initialized accordingly.



**Note:** Some modules provide error handling parameters in setup mode as well. If the step's error handling parameters were configured on a CaptureFlow chart and then during setup, both settings are executed to handle an error in production. The module handles an error according to its setup settings, and then the step's error handling scenario is executed.

5. (Optional) In the **Step Properties** dialog box, add your comment to the **Comments** edit box.
6. Close the **Step Properties** dialog box by clicking **Close X**.

## Copying a batch processing step

### To copy/paste a batch processing step:

1. In a CaptureFlow, click the step you need to copy and press **CTRL+C**, or use the **Copy** option from the **main toolbar**.
2. Open the CaptureFlow which you need the step to be placed to and press **CTRL+V**, or use the **Paste** option.
3. The copied step is inserted into the placing CaptureFlow with all its properties and scripting. For more information, see the following:
  - There is no limitation on pasting several steps which stand for the same batch processing module.
  - If the CaptureFlow chart already contains a step named identically, the copied step is pasted with a new automatically generated name.
  - When a copied step includes scripting, the code refactoring is performed to conform the placing CaptureFlow and new step name after pasting.



### Notes

- The copied step cannot be pasted if its scripting contains syntax errors.
- If the scripting language of the copied step differs from the language of the placing CaptureFlow, the step is pasted without scripting and a warning displays in the **Problems** panel.

### Related Topics:

[“Adding batch creating steps” on page 102](#)

### 4.3.6 Renaming a step

Steps are added to the CaptureFlow chart with the module's name by default.

#### To rename a step:

1. Select the step in the CaptureFlow chart, right-click and select **Rename** from the context menu.

2. Enter the step name.

The step name must start with the *ASCII* alphabetic character ('A-Z', 'a-z') and contain alphanumeric characters and underscore. The character set must be ASCII.

Words `Start` and `End` are reserved and cannot be used as a step name. The step name cannot include the following reserved words as a prefix: `Common`, `Common_Tree`, `Tree`, `XPP`, `AG`.

Any attempt to give an invalid name to a step fails with an error message.

3. Click outside the step to complete renaming.

If the renamed step implements scripting, the source code file of this step is renamed to include the new step name. The `<step nname>_Prepare` and `<step nname>_Finish` event handlers included in this source code file are renamed accordingly. The changes take effect in the script editor immediately, the **Save** operation is not required.

### 4.3.7 Assigning departments to a step

You can specify the department users who are allowed to process batches for that step by assigning the applicable departments to a step.

You can also use departments to route tasks of a step to a module running on one specific machine. After configuring the CaptureFlow process, start up the module where the `-department` command line parameter specifies the same department specified in the CaptureFlow process step. This technique does not apply to the Distributed Web Client module.

CaptureFlow Designer supports two ways of assigning a department to a step:

- **Static department:** The department name is specified in the step with a string (a constant). This works when the tasks of a given step must always be executed in a particular department, and the name of that department is known at design time.
- **Conditional department:** A conditional expression specifies which department is used in each case. The department can be specified as a constant or as an expression that is resolved at runtime.

The syntax and use cases for expressions are explained in [Programming Reference](#).


 **Note:** If you used CaptureFlow Designer 6.5, adding a department to a step is the equivalent to setting the `IATaskRouting` value in *IPP*.

1. Right-click a step in the CaptureFlow chart and select **Departments** from the context menu.
2. In the **Step Departments** dialog box, enter the name of the department (a regular string) to which a task will be sent by default.  

You can specify more than one department separated by a comma. In this case, the server will try to find an available module within the first department in the list. If failed, the server will try the next department, and so on.
3. To specify a conditional department, click the **Add condition** link in the **Step Departments** dialog box. You can specify several conditions, each sending data to a certain department. The conditions are prioritized according to their position in the list.
  - a. Click **<Department Expression>** in the new line and enter the department name. You can enter a regular string or an expression. For example, the following expression defines the department name in the format “dept-`<scan machine name>`”:  

```
“dept - ” & ScanPlus:0.ScanMachine
```
  - b. Click the **<Condition>** placeholder and enter a Boolean expression. For example, the following condition sends data to the specified department if the document is flagged for re-scanning and the scan machine is “Alice” or “Catherine”:  

```
Desktop:1.DocFlagged AND (ScanPlus:0.ScanMachine = “Alice” OR ScanPlus:0.ScanMachine = “Catherine”)
```
4. Close the **Step Departments** dialog box by pressing the **X** icon.
  - c. (Optional) Repeat to add more conditions.
  - d. (Optional) Change the priority of a condition using the **Move Up** and **Move Down** buttons.

 **Note:** To auto-complete an expression, press **CTRL+SPACE** or **CTRL+SHIFT+SPACE**.

### 4.3.8 Adding a decision block

You can add a decision block at any point of the CaptureFlow chart. A decision block includes two or more branches with steps, each starting with a conditional expression. The conditions are checked according to their priority in the block. If any condition evaluates to True, its branch is executed and other conditions and branches in that block are ignored. If none of the conditions are True, the default branch executes.

#### To add a decision block:

1. Select the **Decision** element on the **Steps** panel. Keep the left mouse button pressed and drag it to the chart.

The green arrows on the chart show where the element can be dropped. The arc selected for insertion becomes green.

2. Specify the trigger level of the decision block. Click the **(Level N)** link next to the decision block (diamond) icon and select the level from the list.

For example, if you select level 0 (page level), each page will be evaluated by conditional expressions.

3. Specify the conditional expression for each branch in the block, except for the default branch. Click the **ConditionN** link on the branch and enter the name of the condition and your comment. Enter a boolean expression that will be checked before this branch is selected or not. If the expression evaluates to True, the branch is selected and other branches are skipped.

For example, the following expression evaluates to True for all PDF pages:

```
ScanPlus:0.OutputFileType = "pdf"
```

The expression syntax is described in *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.



**Note:** To auto-complete an expression, press **CTRL+SPACE** or **CTRL+SHIFT+SPACE**.

4. (Optional) Add steps, routing elements, and nested decision blocks to the branches. The added elements must be at or below the trigger level of the decision block.



**Note:** You cannot add a jump to the default branch.

5. Perform value assignments in all **N Values** links in the decision block.
6. (Optional) Add more conditional branches. Click the **Add Branch** button on a branch to add a conditional branch to the right.
7. (Optional) To change the processing order of a branch, click its **ConditionN** link. Use buttons **Earlier** and **Later** to move the branch to the lower or higher priority position, respectively.

### Related Topics:

[“Configuring data transfer between steps” on page 112](#)

[“Adding a jump” on page 111](#)

## 4.3.9 Prioritizing tasks for a step

By default, task priority is defined by the batch priority. You could also set the priority of a specific task by setting the IATaskPriority IA Value as follows:

- Set a step’s **Task priority**. Using conditions and expressions, you can specify different priorities (and thus different IATaskPriority IA Values) for each incoming task.
- Directly setting the IATaskPriority IA Value in code-behind scripts or value assignment expressions. Value assignment expressions specify the priority for an outgoing task to any destination step. For more information, see [“Reserved IA values” on page 487](#).

The IATaskPriority IA Value is assigned to the task in the following order:

1. Code-behind scripts
2. Value assignment expressions
3. **Task Priority** expressions



**Note:** Processing of lower priority tasks could be substantially delayed under the following conditions:

- A heavy load of high priority tasks is being processed.
- The difference in priorities between tasks is large.

### To set a step’s Task priority:

1. Right-click the step, select **Task priority**, and enter an integer between 0 and 99.

The lower the integer, the higher the priority.

Default: A value of 0 (zero) indicates to use the batch priority.

2. (Optional) To add conditions, click **Add condition** and specify the conditions and expressions.

To set a default priority (if none of the conditions are satisfied), specify an integer in **Otherwise, set priority below or leave this blank to use batch priority**.

### 4.3.10 Adding a jump

You can add a jump to a decision block to route data processing to a previous step in the capture process and repeat the flow.

#### To add a jump:

1. Select the **Jump** element from the **Steps** panel.
2. Keep the left mouse button pressed and drag the jump to a conditional branch of a decision block.

The green arrows on the chart show where the jump can be dropped. The arc selected for insertion becomes green.



**Note:** You cannot add a jump to the default branch of a decision block.

3. Click the **Jump** link near the inserted jump and specify the jump name.  
The name must start with the ASCII alphabetic character (A-Z, a-z) and contain alphanumeric characters and underscore. The character set can be ASCII or based on the Regional and Language settings of the local machine.  
Words **Start** and **End** are reserved and cannot be used as a jump name. The following reserved words cannot be part of a jump name: **Common**, **DecisionStart**, **DecisionEnd**, **Tree**, **XPP**, **AG**.
4. Click the **To: <Click to select a step>** link and select the name of the destination element. The list of destinations shows elements located outside the decision block and having trigger levels equal or lower than the element from which you jump.  
The selected destination is marked on the chart with the jump icon and the **From: <jump\_name>** parameter.
5. (Optional) If you need to jump to an element with a higher trigger level, you can add the higher-level synchronization between the initial element and the jump. Then you can configure the jump destination as described in the previous step.

#### Related Topics:

[“Adding synchronization” on page 112](#)

[“Configuring data transfer between steps” on page 112](#)

### 4.3.11 Adding synchronization

If the next element in the CaptureFlow chart uses a higher trigger level than the given one, it is recommended to add synchronization between these elements. The synchronization element puts the next step on hold until all tasks of the previous steps are completed.

It is recommended to add synchronization before and after a decision block and before a task that includes big part of a batch. Synchronization can also be used as a jump destination.

#### To add synchronization:

1. Select **Synchronize** from the **Steps** panel.
2. Keep the left mouse button pressed and drag the synchronization element to the chart.

The green arrows on the chart show where the synchronization element can be dropped. The arc selected for insertion becomes green.

3. Click the **Synchronize** link and specify the name of the element.

The name must start with the ASCII alphabetic character (A-Z, a-z) and contain alphanumeric characters and underscore. The character set can be ASCII or based on the Regional and Language settings of the local machine.

Words **Start** and **End** are reserved and cannot be used as a synchronization name. The name cannot include the following reserved words as a prefix: **Common**, **Common\_Tree**, **Tree\_XPP**, **XPP\_AG**, **AG\_**.

4. Click the **(Level N)** link and select the synchronization level.

The step located below the synchronization will not be triggered until the previous tasks at or below the synchronization level are all completed.

### 4.3.12 Configuring data transfer between steps

After you have added steps and other routing elements to the CaptureFlow chart, you need to configure data transfer between these elements. Every step has input and output IA values defined in the MDF file of the executable module that stands behind. These values are created and assigned during the batch processing:

- Input values of a given step are typically assigned from the steps that have already executed. A step must receive all mandatory input IA values by the moment when it is triggered with a task. If any mandatory IA value is not assigned, the task may not be finished and it may cause the entire process to stop responding.
- Output values are assigned right before the step has finished. The step's output can be copied to input values of other steps that will be executed next.

IA value mappings between steps and routing elements can be configured using any of the options provided in the following table.

**Table 4-10: Setting value assignments: options**

Option	Description	Find more in:
Automatic value mapping	When you add steps on the chart, some value assignments are generated automatically.	<a href="#">“Automatic value mappings” on page 115</a>
Value assignments	You can add all required value mappings on the CaptureFlow chart.	<a href="#">“Adding value mappings on the chart” on page 113</a>
Setup settings	Particular (but not all) values are assigned when you set up the step.	The individual client module guides (as listed in <i>OpenText Intelligent Capture - Module Reference Help (ECPCORE-H-CMD)</i> ) provide more information.
CaptureFlow scripting	You can assign values in the CaptureFlow event handlers. This way is preferable when a value is assigned after a series of calculations or data conversion, or if additional logging is needed.	<a href="#">CaptureFlow scripting</a>

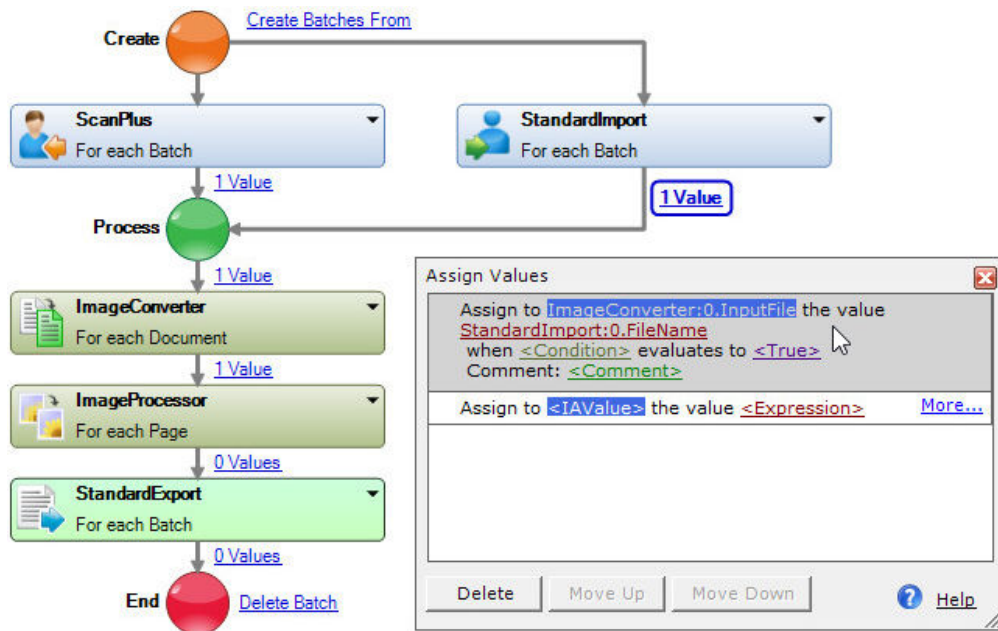
If you combine these options in one CaptureFlow, learn the following:

- The IA value assignment defined in the chart takes precedence over the similar value mapping defined in scripting.
- The value mapping defined on the chart or in scripting takes precedence over the similar mapping in setup.

For more information about all values used by a particular module, see *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*.

#### 4.3.12.1 Adding value mappings on the chart

You can configure data transfer between steps right on a CaptureFlow chart. The chart displays the <N> **Value(s)** link between each two steps. Clicking this link opens the **Assign Values** dialog in which you can specify the value mappings between these steps. For instance, to pass an imported image from the **StandardImport** step to the **ImageConverter** step, the following IA value assignment is required:




**Figure 4-6: Assigning value mappings**

The **Assign Values** dialog allows you to map IA values defined in the *MDF* files of the included steps, **custom values**, and dynamic values that will be created at runtime. To reference a particular value, use the following syntax:


- MDF values: `<step_name>:<level>.<value name>`
- Custom values:
  - `<step_name>:<level>.<value name>` – if a custom value is added to a step.
  - `CustomValues:<level>.<value name>` – if a custom value is added to a CaptureFlow.
- Dynamic values: `<step_name>:<level>.$Runtime.<value name>`

When creating an IA value assignment, the **Assign Values** dialog helps you complete the value path by showing drop-down lists with available options sorted alphabetically. IA values located on top of the list and marked with bold font are mandatory.

 **Note:** When using the escape sequence characters to assign IA values, they can be handled improperly under some locales. Use CaptureFlow scripting to make assignments in such cases.

Autocompletion works for MDF values and custom values that are known at design time. Dynamic IA values are not displayed for autocompletion. However, you can type a dynamic value path in the **Assign Values** dialog, following the syntax described above.

**To configure value mappings on the chart:**

1. Click the **N Values** link between two elements on the CaptureFlow chart. **N** indicates the number of value assignments already done.
  2. Select the last entry in the **Assign Values** dialog box.
  3. Click the **<IA Value>** placeholder and specify the IA value to be assigned with data. Enter the value path manually or using autocompletion. While typing a value name, autocompletion filters the choice list to show items containing the entered text.
  4. Click **OK**.
  5. Click the **<Expression>** placeholder and specify a constant, an expression, or an value path that will be resolved and assigned. The expression syntax is described in [Programming Reference](#).  
Click **OK**.
  6. (Optional) To assign an IA value conditionally, add a conditional expression:
    - a. Click the **More** link to display the condition.
    - b. Click **<Condition>** and enter a **boolean expression** that will be evaluated.
    - c. Click **<True>** and select **True** or **False** to evaluate the specified condition.
    - d. (Optional) Adjust the execution order of the conditional assignments using the **Move Up** and **Move Down** buttons.
    - e. (Optional) Click **<Comment>** and enter a description of the value assignment.
  7. (Optional) Add more assignments. Reorder assignments using the **Move Up** and **Move Down** buttons.
-  **Note:** Grouping similar value assignments together in the list can improve server performance. For the best performance, group value assignments where the destination value level is the same, or where the condition is identical.
8. Click **Close (X)** to close the **Assign Values** dialog box.

**4.3.12.2 Automatic value mappings**

When you add steps to the CaptureFlow chart one after another, some default IA value mappings between steps are added automatically. You can **edit** the automatically added value mappings and add more mappings to meet you business needs.

Notice that if you insert a step above the existing one, automatic mappings are not added. If you delete a step, value mappings related to this step are not deleted automatically. Inconsistencies and value mapping conflicts are indicated on the **<N> values** link. You need to fix them manually.

The following table summarizes the cases when default mappings are added on the chart automatically.

Added step	Transferred data	Output IA value	Input IA value
All modules	Document page (image)	<OutputImage>, <Level0_OutputImage>, or <Image>	<InputImage>, <Level0_InputImage>, or <Image>
Classification (creates OCR data)	Data extracted from the image (OCR data)	<OcrDataCache>	<OcrDataCache>
Identification Extraction Collector	Document type data and values of a particular document page (XML data)	<OutputXmlData>, <PageDataXml>, or <OutputPageDataXml >	<InputXmlData>, <PageDataXml>, or <InputPageDataXml>
Extraction (creates UIM data) Completion Standard Import (used for Web Client)	Data entered by the operator in the index fields (UIM data)	<UimData>	<UimData>

### Related Topics:

[“Adding value mappings on the chart” on page 113](#)

#### 4.3.12.3 Special use cases

This section describes special use cases that require special value mappings between steps. These IA value mappings help you improve the productivity of the CaptureFlow in runtime and allow you to combine in one CaptureFlow executable modules that support different document data models.

#### 4.3.12.4 Transferring document data

Modules like Identification, Extraction, Completion, and Standard Export hold document data in a single <UIMdata> IA value in which document fields and table fields are stored in the array structure.

If you need to transfer document data between UIM-enabled steps in a CaptureFlow, you only need to create a mapping for the <UIMdata> value, or use the default mapping if such was created automatically. For instance, the **Assign Values** dialog opened between two sequential steps **Extraction** and **Completion** must include the following mapping:

Assign to Completion:1.UIMdata the value Extraction:1.UIMdata

Modules with UIM support can read and write any document field from the <UIMdata> value directly.

Expressions and non-UIM modules cannot work with UIM-formatted document data. If your CaptureFlow includes decision blocks that route tasks depending on the document field content, or document data must be passed to a non-UIM step such as an old-version exporter, find the solution in the topics to follow:

#### 4.3.12.5 Copying document fields to IA values

Modules that support UIM data can be set up to flatten document fields into a collection of IA values where each value is intended for a particular document field, or a table array field, or a table cell. The flattened values can be accessed by other CaptureFlow steps or expressions that will execute after data flattening.


When setting up a UIM-enabled step, you specify the destination for the document data output. You can choose a CaptureFlow step that executes later, or the **Custom Values** step.

Next, you need to prepare output IA values. Fields of any document are defined in the document type assigned to this document at runtime. These fields cannot be known in advance, that's why modules do not define field-specific values in their MDF files. You need to construct or select the option to create dynamic values at runtime, or use both. For more information, see [Defining a custom value for a CaptureFlow](#).

You can construct custom values for all or particular UIM fields. If the module fails to find the matching custom value for the field, that field is not output. To create custom values for fields, use the rules provided in the following table.

**Table 4-11: Creating output values for data flattening**

Fields	Requirements	Value path in Assign Values /Expression Editor dialog:
Document fields	Use the following rules: <ul style="list-style-type: none"> <li>• <b>Name:</b> Use the name of the document field as defined in the document type.</li> <li>• <b>Data type:</b> Must be compatible with the data type of the field defined in the document type.</li> <li>• <b>Level:</b> 1 (Document level).</li> <li>• <b>Module:</b> Select the destination step specified during setup.</li> </ul>	<i>&lt;Destination_step&gt;.1.&lt;FieldName&gt;</i>

Fields	Requirements	Value path in Assign Values /Expression Editor dialog:
Table array fields	<p>Use the following rules:</p> <ul style="list-style-type: none"> <li>• <b>Name:</b> Use the table column name as defined in the document type.</li> <li>• <b>Data type:</b> String.</li> <li>• <b>Level:</b> 1 (Document level).</li> <li>• <b>Module:</b> Select the destination step specified during setup.</li> </ul> <p> <b>Note:</b> Create an additional IA value in the format <code>&lt;TableName&gt;_count</code>. This value will hold the number of output table rows.</p>	<p><code>&lt;Destination_step&gt;:1.&lt;Array_FieldName&gt;</code></p> <p><code>&lt;Destination_step&gt;:1.&lt;TableName&gt;_count</code></p>

If you set up the module to use dynamic values, custom values are not obligatory. If the module fails to find a custom value for any field, data will be output into a dynamically created IA value with the following path:

`<Destination_step>:1.$Runtime.<FieldName>`

`<Destination_step>:1.$Runtime.<Array_FieldName>`

The following path is used if you output each table cell in a separate value. Rows are numbered beginning with 1:

`<Destination_step>:1.$Runtime.<Array_FieldName>_<row #>`

`<Destination_step>:1.$Runtime.<TableName>_count`



**Note:** Dynamic values are not displayed in auto-completion help of the **Assign Values** dialog and **Expression Editor**. You need to type the full path.



### 4.3.12.6 Copying IA values to document fields

When a non-UIM module needs to write data into a document field, such as values from a barcode or other, this can be done using CaptureFlow scripting or by assigning IA values as described in this topic.

For example, the CaptureFlow includes a sequence of steps **Extraction** (Extraction), **Calculation** (.NET Code), and **Validation** (Completion). The code recalculates some document data and needs to pass it to **Validation** to be added to UIMdata before the fields are populated and validated by rules.

Open the **Assign Values** dialog right above this step and add the value assignments provided in the following table.

**Table 4-12: Examples of writing external IA values to UIM data fields**


Assign to	Value	Comment
Validation:1. UimDataImportMode	1	Allow document data binding for the <b>Validation</b> step. By default, modules have their <code>&lt;UimDataImportMode&gt;</code> IA value set to 0 (binding values to UIM data is not allowed).
Validation:1.\$Runtime .InUimData_Date	Completion:1.NewDate	Write the <code>&lt;NewDate&gt;</code> IA value to the <b>Date</b> document field.   <b>Note:</b> Since you have set the import mode to 1, you can import all IA values at level 1 (Document level).
Validation:1.\$Runtime .InUimData_Quantity_ _1	Completion:1.Qty	Write the <code>&lt;Qty&gt;</code> IA value to the table cell in the <b>Quantity</b> column and in the second row.   <b>Note:</b> For import mode 1 all table fields are imported at level 1 (Document level). The array field name must be separated from the row index by two underscore characters. The first row is indexed 0.

 **Note:** The mappings in the example above use dynamic `<InUimData>` values. You can also construct **Creating custom values** with the `<InUimData_>` prefix as described in “Copying document fields to IA values” on page 117 and use them to create value mappings.

When the **Validation** step is triggered with a task, all values assigned to `<InUimData_>` constructs are imported into document fields where the `InUimData_<FieldName>` matches the field name as specified in the document type. Ensure that the field names in the `<InUimData_>` constructs are valid for the document type and that values are either of the correct type or can be converted to the correct type. For example, the binding will convert “4” (String) to 4 for a numeric field, but will fail if the input value for the field is “abc”.

After that the `<UimDataImportMode>` IA value is reset to 0 automatically.

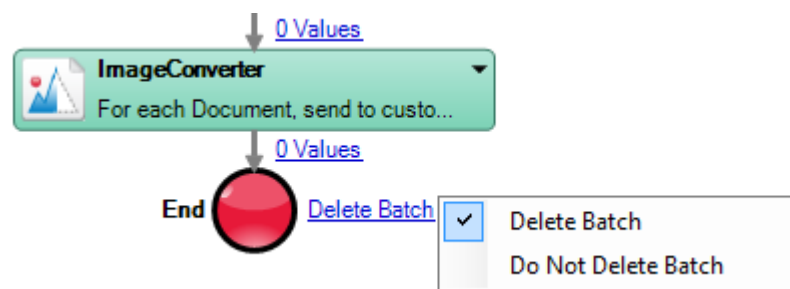
The Completion module allows you to import IA values to document fields whose index level in the document type is higher than 1 (Document). In this case, set the import mode to 2 and construct `<InUimData_>` values at the index level specified in the document type for each field.

 **Note:** You can use `<InUimData>` to update existing index fields, including existing rows in array fields. However, you cannot populate table cells in rows that do not exist using `<InUimData>`. If you want to create and populate new fields or rows, use the .NET Code module. For more information, see *OpenText Intelligent Capture - Utilities Modules Guide (ECPCORE-CMU)*.

### 4.3.13 Configuring the end of a CaptureFlow

You can choose between two batch handling scenarios that complete the CaptureFlow:

- **Delete batch:** When the last step in the CaptureFlow is completed, the process automatically deletes the batch structure and all associated files from the disk space.
- **Do not delete batch:** The batch structure and all associated files are not deleted from the disk space.



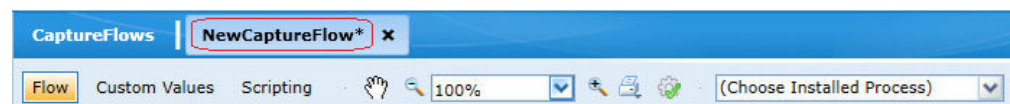
**Figure 4-7: Configuring the end of a CaptureFlow**

**To configure the end of a CaptureFlow:**

1. Select the **End** point on your CaptureFlow chart and right-click it.
2. Select the preferred option from the context menu:
  - **Delete Batch**
  - **Do Not Delete Batch**

**4.3.14 Saving a CaptureFlow**

Unsaved changes in the CaptureFlow chart, custom values, or CaptureFlow scripting are indicated by the asterisk (\*) sign on the CaptureFlow button near the CaptureFlow name:

**Figure 4-8: Saving a CaptureFlow**

Changes are saved in the XPP file when you do one of the following:

- Press the **Save** button in the toolbar and select the **Save** menu option.  
If you select **Save All**, changes will be saved for all CaptureFlows opened in CaptureFlow Designer.
- Click the “close” icon on the CaptureFlow tab. Select the **Save changes** from the prompt dialog box.
- Close CaptureFlow Designer. Select the **Save changes and exit** option from the prompt dialog box.

**4.3.15 Saving a CaptureFlow with a different name**

Save the existing CaptureFlow with a different name and use the clone for designing a new CaptureFlow. The clone inherits the CaptureFlow structure, IA values, all step setup settings, and optionally the installed CaptureFlow instances from the original CaptureFlow.

**To clone a CaptureFlow:**

1. Open the original CaptureFlow chart in CaptureFlow Designer.
2. Press the **Save** button and select the **Save As** menu option. The **CaptureFlow - Save As** window opens.
3. Enter the CaptureFlow name in the **New CaptureFlow Name** edit box. When entering the name, follow the [naming convention on page 481](#) for CaptureFlows.

4. (Optional) When cloning a CaptureFlow which already includes installed process instances, you can additionally add these process instance settings into the new CaptureFlow. To do this, select the required processes from the **Existing Process Name** list. For each of the copied process, you can change its name in the **New Process Name** column; otherwise, the copied processes will be saved with the default names.
5. Press **Save**. A CaptureFlow chart with the new name opens for editing.



**Note:** For CaptureFlows cloned with process instances, the process names appear in the **(Choose Installed Process)** drop-down list although they are not installed. **Install** these processes to the server or **delete** them from the CaptureFlow as appropriate.

The original CaptureFlow is closed without a prompt window. If the original CaptureFlow had unsaved changes, including scripting, they are lost.

### 4.3.16 Printing a CaptureFlow chart



1. Click the **Print** button (CTRL+P)
2. Set up print options, such as paper size and whether to print page numbers.  
You can also preview the printout prior to sending it to the printer..

## 4.4 Defining a custom value for a CaptureFlow

You can extend the scope of default IA values by adding custom values for special needs, such as making decisions, passing special information between steps, and others. Like other IA values, custom values can be used in IA value assignments, expressions, and in scripting.


You can manage custom values in the **Custom Values** panel of the CaptureFlow Designer.


### 4.4.1 Creating custom values

**To create a custom value:**

1. Click the **Custom Values** button in the CaptureFlow Designer toolbar.
2. From the **Custom Values** pane, click <New Value> in the **Name** column to enter a custom value name.
3. Specify the parameters for the new custom value as stated in the following table.

**Table 4-13: Custom Values panel: fields**

Column	Mandatory	Description
Name	Yes	Specifies the custom value name that must comply with the <b>naming convention on page 485</b> .
Module	Yes	Indicates the place where the given custom value will be declared.  See the following options: <ul style="list-style-type: none"> <li>• <b>CustomValues</b>: the value will be declared in the list of custom values of the given CaptureFlow.</li> <li>• <b>&lt;module_name&gt;</b>: the value will be declared in the list of IA values of the given step.</li> </ul>
Type	Yes	Specifies the data type of the value. See the following options: <ul style="list-style-type: none"> <li>• Boolean</li> <li>• Date</li> <li>• File (saves a copy of a stage file)</li> <li>• Float</li> <li>• Integer</li> <li>• String</li> </ul> <p> <b>Note:</b> Custom values are restricted to standard value types and cannot use user-defined object types. To workaroud this limitation, you can define your object type values in a data-only MDF file, <b>add this file as a custom module</b> to CaptureFlow Designer, and use it as a step in your CaptureFlow chart.</p>

Column	Mandatory	Description
Level	Yes	Specifies the level in the batch. The value will be created in every batch node on the specified level. Options: 0 to 7. Default: 0.
Initial Value	No	Specifies the data to be assigned to the custom value. Is applicable for custom values of type Integer and String.   <b>Note:</b> If the initial value does not match the data type, an error occurs during batch processing.
Comment	No	Specifies the description of the custom value.

 **Note:** To speed up the creation of a large number of values with similar properties, use [Clipboard operations](#).

- Click the **Flow** button on the CaptureFlow Designer toolbar to return to the chart.

### Related Topics:

[“Copying values between CaptureFlows” on page 124](#)

[“Using Clipboard operations” on page 125](#)

## 4.4.2 Copying values between CaptureFlows

CaptureFlow Designer allows you to define custom values in any CaptureFlow and them quickly move them all or only selected values to other CaptureFlows. You can copy custom values within the same capture system and between different capture systems.

### To copy custom values from a different CaptureFlow:

- Open the source CaptureFlow and the target CaptureFlow in the same CaptureFlow Designer tool. Copying data between different CaptureFlow Designer windows is not supported.
- Open the **Custom Values** panel of the source CaptureFlow and select one or several lines from the **Custom Values** grid.
- Copy the selected lines using the **Edit (button) > Copy** menu command or the **CTRL+C** keystroke.

4. Switch to the target CaptureFlow and open the **Custom Values** panel.
5. Select the <**New Value**> line and insert the copied custom values using the **Edit (button) > Paste** menu command or the **CTRL+V** keystroke.

The copied lines are inserted at the end of the target **Custom Values** panel, with the following exceptions:

- If a line or several lines with the same **Name**, **Module**, and **Level** parameters already exist in the target **Custom Values** panel, a dialog box displays. Choose one of the following options:
  - **Yes**: Click this button to overwrite the existing custom values with the copied ones.
  - **No**: Click this button not to overwrite the lines that already exist.
  - **Cancel**: Click this button to cancel the paste operation for all currently copied lines.
- If a line's **Module** parameter defines a step name that does not exist in the target CaptureFlow, a dialog box displays a warning message. Click **OK**. The line is not copied to the target **Custom Values** panel.

All inserted and modified lines in the **Custom Values** panel are selected.

6. Save the changes and click the **Flow** button on the toolbar to return to the chart.

### Related Topics:

[“Creating custom values” on page 122](#)

[“Using Clipboard operations” on page 125](#)

## 4.4.3 Using Clipboard operations

You can edit custom values in the **Custom Values** grid using the standard Clipboard operations, such as copy, paste, delete, undo, and redo. Select a line or multiple lines in the **Custom Values** grid and apply one of the following:

- **Edit** menu commands: **Copy**, **Paste**, and others.
- Standard Windows keyboard shortcuts: **CTRL+C**, **CTRL+V**, and others.

[“Clipboard operations supported in the Custom Values grid” on page 125](#) provides Clipboard operations supported in the **Custom Values** grid.

**Table 4-14: Clipboard operations supported in the Custom Values grid**

Clipboard operation	Supported	Notes
Copy/paste/delete on the same grid	Yes	The <b>Cut</b> operation is not supported.

Clipboard operation	Supported	Notes
Copy/paste/delete between grids (that is, between different CaptureFlows)	Yes	
Drag-and-drop	No	

**Related Topics:**

[“Creating custom values” on page 122](#)

[“Copying values between CaptureFlows” on page 124](#)

## 4.5 Developing scripting

CaptureFlow Designer allows you to extend or override the logic of a capture process by implementing CaptureFlow scripting. To let you write the script code for your CaptureFlow, the **Scripting** panel of CaptureFlow Designer implements the functionality of a standard script editor.

Topics in this section provide the guidelines for creating CaptureFlow scripting, describe the **Scripting** panel capabilities, and explain how to manage scripting in CaptureFlow Designer.

### 4.5.1 Understanding CaptureFlow scripting

A CaptureFlow exposes several event handlers in which you can add custom code. To make these event handlers available for editing, you need to include them in the CaptureFlow manually at design time. In particular, you can include a set of event handlers for any CaptureFlow step, an event handler for batch deletion, and an event handler to define default error handling for any CaptureFlow step which fails to execute.

When you include step scripting or batch specific scripting, a script file with auto-generated code is added to a CaptureFlow. To edit auto-generated code, use the **embedded script editor** available from the **Scripting** tab of CaptureFlow Designer. Besides the auto-generated code, you can create static functions in any script file. Then, you can call the static functions from any of the other steps.

### 4.5.1.1 Supported scripting languages


You can implement CaptureFlow scripting in C# or VB.NET. The scripting language is selected for each CaptureFlow when adding the first script file to it. All other script files added to that CaptureFlow automatically use the selected language. If you need to change the selected language, the only way is to delete scripting (all files) from the CaptureFlow and start creating it anew.


### 4.5.1.2 Event handlers

When you **add scripting** for a step or batch, a script file with empty event handlers is added to your CaptureFlow automatically.

“CaptureFlow scripting: event handlers” on page 127 provides a description of CaptureFlow script files.

**Table 4-15: CaptureFlow scripting: event handlers**

Script file	Method	Description	Note
<code>&lt;step_name&gt;.cs</code> or <code>&lt;step_name&gt;.vb</code>	<code>&lt;step_name&gt;_Prepare()</code>	Executes before the triggered module starts the execution of the task.	<p>The <code>Prepare()</code> and <code>Finish()</code> event handlers are provided for initializing IA values and passing data between steps, especially when it requires preliminary calculations, data conversion, or other kinds of complex processing.</p> <p> <b>Note:</b> When preliminary data processing is not required, assign values in the <b>Assign Values</b> window that you can open from the step's context menu on the CaptureFlow chart. It is generally recommended to avoid scripting where value</p>

Script file	Method	Description	Note
	<code>&lt;step_name&gt;_Finish()</code>	Executes after the task is finished.	assignments can be performed in the <b>Assign Values</b> window.
	<code>&lt;step_name&gt;_Error()</code>	Executes if the step execution fails.	The <code>Error()</code> event handler is typically used for error logging and other operations related to step error handling.
Global.cs or Global.vb	DefaultStepError()	Implements default error handling behavior applicable to any step. This method is called for a failed step that does not implement its own <code>&lt;&lt;step_name&gt;&gt;_Error()</code> event handler. Otherwise, the step specific error handler is triggered instead.	 <b>Note:</b> To trigger the <code>DefaultStepError()</code> method for a failed step that implements scripting, add the <code>DefaultStepError()</code> call inside the <code>&lt;&lt;step_name&gt;&gt;_Error()</code> method.
	Batch_Delete()	Overrides the default batch delete behavior. This method is called before the batch is physically deleted at the end of the CaptureFlow, or when the administrator deletes the batch manually in Intelligent Capture Administrator.	

### 4.5.1.3 Scope of operations

The scope of operations allowed in CaptureFlow scripting is defined by the `<System>` namespace (.NET) and the `<EMC.InputAccel.CaptureFlow>` namespace. For more information about the CaptureFlow scripting API, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

CaptureFlow scripting cannot be used for setting any triggers, it can only manipulate the data in the batch. CaptureFlow scripting is not meant for calling external libraries and executing time and resource consuming operations since their execution takes place synchronously on the server. For more information about the operations not recommended in scripting, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

CaptureFlow scripting can also be used to perform logging using two functions as follows:

- `LogMessage()`: Allows you to log a message to the `debug.out` file in the IAS directory of the Intelligent Capture Server.
- `LogEvent()`: Allows you to log an event to the Windows Event Log.

These functions are accessible on the `IBatchNodeData` interface that is passed to all optional CaptureFlow event handlers. These logging functions are recommended for use instead of .NET Tracing through the `<System.Diagnostics>` namespace. .NET Tracing is not supported in CaptureFlow scripting.



**Note:** The above logging functions behave similar to `IALogDebug` and `IALogNTMessage` in Process Developer. For `LogMessage()`, configure the server's trace level in the same way you do with `IALogDebug`. For more information, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

### 4.5.1.4 Code execution order for step initialization

When designing a capture process, you can define initialization of step IA values in several ways as follows:

- In the `<step name>_Prepare()` script method
- By adding **IA value assignments** in the **Assign Values** dialog box in the CaptureFlow chart
- During step setup (only particular values)

When choosing the way(s) to initialize an IA value, keep in mind the following execution order in a batch step:

1. Initialization performed by `<step name>_Prepare()` (server side)
2. Input IA value assignments (server side)
3. Step execution (client side; setup settings are loaded from the server)

4. Initialization performed by `<step name>_Finish()` (server side)
5. Output IA value assignments (server side)


In the above execution sequence, the IA value assignment applies unconditionally, overwriting the IA value if it was set by a preceding script method. The setup setting applies only if the value is not initialized; otherwise, the setup setting is ignored.

#### 4.5.1.5 Code execution order for a step in error

Depending on the step's **error handling scenario** (Put on Hold or Continue on Error), the step's error handling methods can be called in a different order. **“Script code execution for a step in error” on page 130** summarizes all possible scenarios and provides the script code execution order in each case.

**Table 4-16: Script code execution for a step in error**

Script methods defined	Put on hold	Continue on error
<code>&lt;step&gt;_Prepare</code> <code>&lt;step&gt;_Finish</code> (no error handling)	—	<code>&lt;step&gt;_Finish()</code>
<code>&lt;step&gt;_Prepare</code> <code>&lt;step&gt;_Finish</code> <code>&lt;step&gt;_Error</code>	<code>&lt;step&gt;_Error()</code>	<ol style="list-style-type: none"> <li>1. <code>&lt;step&gt;_Error()</code></li> <li>2. <code>&lt;step&gt;_Finish()</code></li> </ol>
<code>&lt;step&gt;_Prepare</code> <code>&lt;step&gt;_Finish</code> DefaultStepError	DefaultStepError()	<ol style="list-style-type: none"> <li>1. DefaultStepError()</li> <li>2. <code>&lt;step&gt;_Finish()</code></li> </ol>
<code>&lt;step&gt;_Prepare</code> <code>&lt;step&gt;_Finish</code> <code>&lt;step&gt;_Error</code> DefaultStepError	<code>&lt;step&gt;_Error()</code>	<ol style="list-style-type: none"> <li>1. <code>&lt;step&gt;_Error()</code></li> <li>2. <code>&lt;step&gt;_Finish()</code></li> </ol>
DefaultStepError	DefaultStepError()	DefaultStepError()

 **Note:** The `<step>_Error()` method is not defined if it is commented out or deleted from the script file. If the script file includes an empty method, it is considered defined and will be triggered.

#### Related Topics:

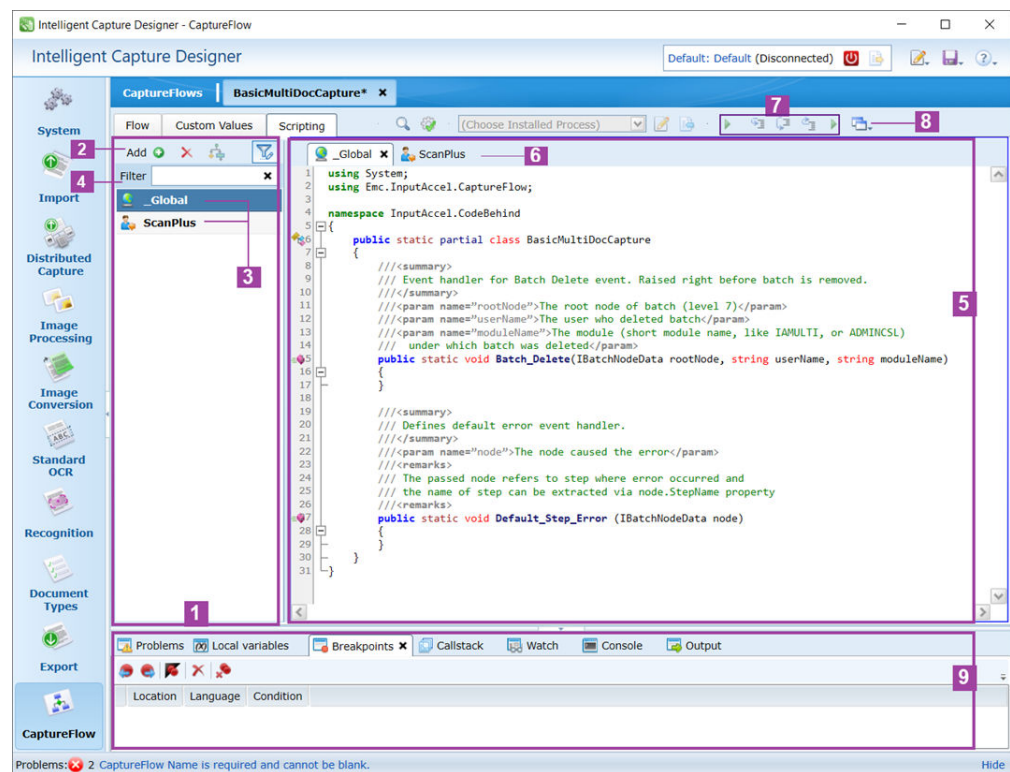
**“Adding batch processing steps” on page 104**

## 4.5.2 Script editor

CaptureFlow scripting is created and edited in the **Scripting** tab of CaptureFlow Designer. This tab provides an embedded script editor where you can create and debug the C# and VB.NET code (Figure 4-9).

### 4.5.2.1 User interface

To view the **Scripting** tab, open any CaptureFlow chart in CaptureFlow Designer and click the **Scripting** button located in the toolbar next to the **Flow** and **Custom Values** buttons. To switch back to the chart, click the **Flow** button.



**Figure 4-9: Scripting tab**

The **Scripting** tab includes the areas and elements outlined in the following table.

**Table 4-17: CaptureFlow Designer: Scripting tab elements**

Area	Description and included elements
Navigation panel (1)	<p>Serves for navigating between the script files of the CaptureFlow. Includes the following elements:</p> <ul style="list-style-type: none"> <li>• Navigation toolbar (2): displays the buttons for manipulating script files in a CaptureFlow.</li> <li>• Steps list (3): displays CaptureFlow steps with scripting and the <b>Global</b> item (batch scripting), ordered alphabetically.</li> <li>• Filter panel (4): serves for filtering items in the steps list (3). Is hidden by default, appears by the <b>Filter</b> button click.</li> </ul>
Navigation toolbar (2)	<p>Displays the buttons for managing scripting in a CaptureFlow:</p> <ul style="list-style-type: none"> <li>• <b>Add</b> button: opens the <b>Add Script</b> window where you can select a step or the <b>Global</b> item (batch) to be enhanced with scripting.</li> <li>• <b>Remove</b> icon: removes the selected scripting (3) from the CaptureFlow.</li> <li>• <b>Navigate to the chart</b> icon: opens the CaptureFlow chart and sets focus on the step that is currently selected in the steps list (3). Is dimmed when <b>Global</b> is selected in the steps list (3).</li> <li>• <b>Filter</b> icon: displays the filter panel (4) which is hidden by default.</li> </ul>
Steps list (3)	<p>Displays the list of CaptureFlow steps with scripting and the <b>Global</b> item (batch scripting). The items are displayed in alphabetical order.</p>
Filter panel (4)	<p>Displays a filter box where you can enter a string pattern to filter the steps list (3). Is hidden by default, displays by toggling on the <b>Filter</b> icon in the navigation toolbar (2).</p>
Editor panel (5)	<p>Displays a tab for each script file opened for editing. The tab buttons (6) display the file names and serve for switching between the opened script files.</p>

Area	Description and included elements
Debug toolbar (7)	<p>Displays the buttons for managing the debug session. Includes the following buttons:</p> <ul style="list-style-type: none"><li>• <b>Start/stop debugging</b> icon: starts or stops the debug session.</li><li>• <b>Step into</b> icon: steps inside the subroutine and execute the next code operation.</li><li>• <b>Step over</b> icon: skips an entry point to a subroutine, such as a method or event handler, and executes the next code operation.</li><li>• <b>Step out</b> icon: executes the code till the exit from the subroutine is reached.</li><li>• <b>Continue</b> icon: continues the script execution stopped by a breakpoint.</li></ul> <p>All buttons on the debug toolbar (7) are grayed out at design time and become available during debugging.</p>
Debug panel menu (8)	<p>Expands by a button click. Displays the names of auxiliary debug panels (9) that can be selected: <b>Breakpoints, Call Stack, Console, Locals, Watch, Output</b>. The selected panel appears in the debug panel (9). Unchecked auxiliary panels are hidden from the debug panel.</p>

Area	Description and included elements
<p>Debug panel (9)</p>	<p>Displays several auxiliary panels for tracing the intermediate code execution results and debugging status. All panels except <b>Problems</b> can be displayed or hidden from the <b>Debug panel</b> menu (8).</p> <p>Can optionally display the <b>Problems</b> panel and the <b>Search</b> panel. The <b>Problems</b> panel is hidden by default and appears only to display unresolved errors and warnings generated for the capture system. The <b>Search</b> panel is displayed or hidden by clicking the toggle button on the toolbar.</p> <p>The list of debugging panels can include:</p> <ul style="list-style-type: none"> <li>• <b>Local Variables:</b> displays the list of variables defined inside the current method. This list is created automatically.</li> <li>• <b>Breakpoints:</b> displays breakpoints set in all script tabs (files). The toolbar buttons allow the user to navigate between the breakpoints, enable/disable all breakpoint, delete the selected breakpoint, and delete all breakpoints. The check box serves to enable/disable the breakpoint. The <b>Condition</b> field serves to enable the breakpoint only if the conditional expression entered in C# or VB.NET evaluates to "true".</li> <li>• <b>Callstack:</b> displays the current execution stack during debugging.</li> <li>• <b>Watch:</b> displays the list of expressions that will be calculated and their results displayed during debugging. The expressions are added by the user.</li> <li>• <b>Console:</b> displays the C# or VB.NET code that will be executed and the result displayed during debugging. The code is added by the user. The language can be selected from the drop-down list inside the debug console.</li> <li>• <b>Output:</b> displays the compilation and debugging log entries. The toolbar controls allow the user to switch between the <b>Build</b> and <b>Debug</b> modes, clear all output, and toggle/toggle off the word wrap mode.</li> </ul>

### 4.5.2.2 Clipboard operations

You can edit your script code using the drag-and-drop operations and the standard Clipboard operations, such as copy, cut, paste, delete, undo, and redo. You can apply these commands to the selected text from the **Edit** menu, or from the context menu called with a secondary mouse button click in the code, or using standard Windows keyboard shortcuts.

“Clipboard operations supported in the script editor” on page 135 provides a description of the Clipboard operations supported in the script editor.

**Table 4-18: Clipboard operations supported in the script editor**

Clipboard operation	Edit menu	Context menu	Windows shortcuts	Notes
Copy, Cut, Paste	Available	Available	CTRL+C CTRL+X CTRL+V	These operations work in the same script file and between opened script files, including script files belonging to different CaptureFlows.
Delete	Grayed out	Not displayed	DELETE	When you work in the script editor, the <b>Delete</b> operation is only available through the standard Windows shortcut.
Undo, Redo	Grayed out	Available	CTRL+Z (undo) CTRL+Y (redo)	These operations are not available from Designer' <b>Edit</b> menu. The script editor keeps the changes in the code in its own stack. To undo and redo the changes, use the context menu commands or standard Windows shortcuts.

### 4.5.3 Adding scripting

There are two ways how you can add scripting to your CaptureFlow:

- [“Adding scripting in the script editor” on page 136](#): You can add all kind of scripting in the embedded script editor (**Scripting** tab).
- [“Adding step scripting in the chart” on page 137](#): You can add scripting to a particular step right in the chart using the step's context menu.

#### 4.5.3.1 Adding scripting in the script editor

When working in the embedded script editor (**Scripting** tab), you can add any kind of scripting to the CaptureFlow.

##### To add scripting in the Scripting tab:

1. Open the CaptureFlow chart in CaptureFlow Designer and click the **Scripting** button on the toolbar. The **Scripting** tab opens.  
The navigation panel displays the CaptureFlow steps that already have scripting, and the **Global** item if batch specific scripting is added.
2. Click the **Add** button in the navigation toolbar.  
The **Add Script** window opens with the list of CaptureFlow items where you can add scripting. The list includes the step names and the **Global** item (optional) ordered alphabetically. The list does not include items where scripting has been added.
3. Select the CaptureFlow item for which you need to create custom code. You can select multiple items in the **Add Script** window and add script files for all of them with a single button click.  
To narrow the search, start typing the step name in the **Filter** edit box. While you are typing, the list of items is filtered by the specified string pattern.
4. Click **OK** to confirm the selected item(s).
5. If the opened CaptureFlow doesn't have scripting, new scripts will be in C# programming language only. If the opened CaptureFlow has a VB.Net script, it can be edited and new scripts using VB.net programming language are created.  
CaptureFlow Designer generates the new script file(s), displays the new items in the navigation panel, and opens the new file(s) for editing.  
To switch between several opened script files, click the `<<script_file_name>>` tab button of the required file, or press **CTRL+TAB**, or double-click the script file in the navigation panel.
6. Add the code inside the event handlers as described in [“Understanding CaptureFlow scripting” on page 126](#).



**Note:** You can exclude any unused event handler from scripting using comments. You can also delete unused auto-generated code from the

script file. In that case, be aware that auto-generated code cannot be restored after deletion automatically.

7. Save the updated script file as described in [“Saving scripting” on page 141](#).

### Related Topics:

[“Adding step scripting in the chart” on page 137](#)

[“Saving scripting” on page 141](#)

[“Understanding CaptureFlow scripting” on page 126](#)

## 4.5.3.2 Adding step scripting in the chart

You can quickly add scripting to a particular step right in the CaptureFlow chart. This operation adds a script file with all event handlers specific for a step.

### To add scripting to a step:

1. Open the CaptureFlow chart in CaptureFlow Designer.
2. Right-click the step and select **Scripting** from the context menu.
3. (Optional) If the CaptureFlow chart does not implement scripting, select the programming language the **Script Options** window: **C#** or **VB.NET**. Confirm your choice.

A new script file opens in the **Scripting** tab for editing. The file includes the following empty event handlers: `<step>_Prepare()`, `<step>_Finish()`, and `<step>_Error()`.



**Note:** You can exclude any unused event handler from step scripting using comments. You can also delete unused auto-generated code from the script file. In that case, be aware that auto-generated code cannot be restored after deletion automatically.

### Related Topics:

[“Adding scripting in the script editor” on page 136](#)

[“Saving scripting” on page 141](#)

[“Understanding CaptureFlow scripting” on page 126](#)

## 4.5.4 Deleting scripting

There are two ways how you can delete scripting from a CaptureFlow:

- [“Deleting scripting in the script editor” on page 138](#): You can delete scripting in the embedded script editor (**Scripting** tab).
- [“Deleting step scripting in the chart” on page 139](#): You can delete scripting of a particular step right in the chart using the step's context menu.

### 4.5.4.1 Deleting scripting in the script editor

When working in the embedded script editor (**Scripting** tab), you can delete any script file from the CaptureFlow.

#### To remove scripting in the Scripting tab:

1. Open the CaptureFlow chart in CaptureFlow Designer and click the **Scripting** button in the toolbar. The **Scripting** tab opens.
2. In the navigation panel, select scripting to be deleted from the CaptureFlow scripting.

The navigation panel displays the CaptureFlow steps that have scripting, and the **Global** item if batch specific scripting is added. To narrow the search, click the **Filter** icon in the navigation toolbar. Start typing the step name in the **Filter** edit box. While you are typing, the list of items is filtered by the specified string pattern.

3. Click the **Remove** icon in the navigation toolbar.

The selected scripting disappears from the navigation panel.

To restore the deleted scripting, use the **Undo** menu command. After you have saved the CaptureFlow, the deletion cannot be reverted.

#### Related Topics:

[“Deleting step scripting in the chart” on page 139](#)

[“Saving a CaptureFlow” on page 121](#)

[“Understanding CaptureFlow scripting” on page 126](#)

### 4.5.4.2 Deleting step scripting in the chart

You can quickly delete scripting of a particular step right in the CaptureFlow chart. This operation deletes the script file of that step with all of its event handlers.

#### To remove step scripting in the chart:

1. Open the CaptureFlow chart in CaptureFlow Designer.
2. Right-click the step and select **Remove scripting** from the context menu.
3. Save the CaptureFlow.

To restore the deleted scripting, use the **Undo** menu command. After you have saved the CaptureFlow, the deletion cannot be reverted.

#### Related Topics:

[“Deleting scripting in the script editor” on page 138](#)

[“Saving a CaptureFlow” on page 121](#)

[“Understanding CaptureFlow scripting” on page 126](#)

### 4.5.5 Opening scripting

#### To open a script file:

1. Open the CaptureFlow chart in CaptureFlow Designer and click **Scripting** to open the **Scripting** tab.
2. In the navigation panel, select scripting to open.

The navigation panel displays all CaptureFlow steps with scripting and the **Global** item for batch scripting (optional) ordered alphabetically. To narrow the search, click the **Filter** icon on top of the navigation panel. Start typing the step name in the **Filter** edit box. While you are typing, the list of items is filtered by the specified string pattern.

3. Double-click the selected scripting. The script file opens in a separate `<script_file_name>` tab in the editor panel.

To switch between the script files opened in the **Scripting** panel, click the `<script_file_name>` tab button, or use **CTRL+TAB**, or double-click the step name or event handler name the navigation panel.

## 4.5.6 Editing scripting

### 4.5.6.1 Inserting script methods

Every time you add scripting to a CaptureFlow, you actually create a script file with empty methods (event handlers) specific for this kind of scripting. For instance, a script file added for a step includes three methods – `<step>_Prepare()`, `<step>_Finish()`, and `<step>_Error()`. Each method is added with an empty body and commented instructions. You can delete any of these predefined code patterns if they are not required. Later, you can easily restore the deleted code pattern using the autocompletion context help.

Also, you may need to insert missing script methods to a CaptureFlow converted from an earlier-version XPP. During conversion the original script code is copied to the new XPP as is, which means that step scripting does not include the new `<step>_Error()` event handler. Use autocompletion to insert this missing method where necessary.

#### To restore a deleted code pattern:

1. Drop the cursor inside the `<CaptureFlow>` class. If the class includes other public methods, position the cursor above or below the existing method's definition.
2. Start typing the name of the required method. The context help expands to show the names of the matching code constructs, including the method names.
3. Select the required method and click **TAB** to insert the code pattern.

#### Related Topics:

[“Understanding CaptureFlow scripting” on page 126](#)

[“Opening a non-versioned CaptureFlow” on page 93](#)

### 4.5.6.2 Shortcuts for editing code

The script editor supports the shortcuts provided in the following table for code editing.

**Table 4-19: Shortcuts keys for quick code editing**

Shortcut keys	Description
CTRL+DELETE	Delete the word suffix
CTRL+BACKSPACE	Delete the word prefix
CTRL+A	Select all
CTRL+D	Delete the current line

Shortcut keys	Description
CTRL+F	Open the popup panel for incremental search in the code
F3	Incremental search: find the next inclusion in the code
SHIFT+F3	Incremental search: find the previous inclusion in the code
CTRL+SPACE	View all members of the current object in the drop-down list
CTRL+I	Restore default indents for all selected lines
TAB	Increase indent for the selected line of code
SHIFT+TAB	Decrease indent for the selected line of code
CTRL+SHIFT+L	Expand/collapse all constructs in the code
CTRL+SHIFT+M	Expand/collapse the currently selected construct in the code
CTRL+~	Add/remove commenting characters in front of the selected line(s)
F9	Set or remove a breakpoint on the selected line
F10	Step over or skip an entry point to a subroutine
CTRL+F5	Stop debugging
CTRL+scroll the mouse wheel up/down	Zoom in/out

### 4.5.7 Saving scripting

You can save updates in CaptureFlow scripting in one of the following ways:

- Close the *<script file name>* tab in the editor panel and click **Yes** in the prompt window. The script file is closed and the changes made in this file are saved in the XPP file.
- Save the entire CaptureFlow as described in [“Saving a CaptureFlow” on page 121](#).

#### Related Topics:

[“Saving a CaptureFlow” on page 121](#)

## 4.5.8 Building scripting

To build an assembly, you need to compile the CaptureFlow in CaptureFlow Designer as described in [“Compiling a CaptureFlow” on page 443](#).

### Related Topics:

[“Compiling a CaptureFlow” on page 443](#)

## 4.5.9 Debugging scripting

To debug scripting, you need to install a CaptureFlow on a test server, set up its steps, and create a batch for testing in production.

Before you start debugging, make sure that Designer is connected to a single server rather than to a ScaleServer group. This guarantees that the batch for testing is created on the same server to which Designer is connected.

### To debug a batch:

1. Start the batch creating module of your process in production mode.
2. In the module’s window, specify the name of the process for which you need to create a batch. Click the **Create Batch** button.
3. Specify the name of the batch file that you want to create.
4. In CaptureFlow Designer, open the required CaptureFlow chart and select the installed process from the list.



### Notes

- If the required CaptureFlow is displayed with a yellow triangle icon, debug cannot be started. If you try to open such a file, an error message informs you that this CaptureFlow is created in an earlier version of CaptureFlow Designer and thus has a different format. To debug the batch, use the proper version of CaptureFlow Designer.
  - To perform debugging successfully, batch version must match the version of the selected process. Otherwise, for example, if a batch is non-versioned or the version of the batch and the version of the process mismatch, a user will be notified that debugging results might lead to unexpected behavior.
5. Click the **Start Debug** button on the CaptureFlow Designer toolbar, or press **CTRL+F5**.
  6. Select the batch in the **Debug Batch** dialog box and click the **Debug** button.  
This command opens the source code of your CaptureFlow in the **Scripting** tab of CaptureFlow Designer. Until the debug session is ended, CaptureFlow Designer is locked to display the **Scripting** tab with the batch code being debugged.

To trace the code execution, you can click at the selected line in the code to add or remove breakpoints. Optionally, you can specify a conditional expression in the **Condition** field to execute the breakpoint only if the condition evaluates to “true”. A conditional expression can use C# or VB.NET syntax, irrespective the language of the script file. The language of the conditional expression is detected automatically in the **Language** field.



**Note:** When entering a conditional expression for a breakpoint, make sure to correct all syntax errors if any reported. During debugging, breakpoints with incorrect conditional syntax lead to unpredictable behavior rather than to an exception.

You can also use watchers, execute expressions, execute the code in the debug console, and view the call stack and the values assigned to variables inside the current method. These options are available from the auxiliary panels located next to the **Problems** panel. The auxiliary panels can be displayed or hidden, depending on their selection in the debug panels menu.

You can also use standard debug commands available in the debug toolbar:

- **Break:** stop the script execution at the current execution point.
  - **Step into** (or **F11**): step inside the subroutine and execute the next code operation.
  - **Step over** (or **F10**): skip an entry point to a subroutine, such as a method or event handler, and execute the next code operation.
  - **Step out** (or **SHIFT+F11**): execute the code till the exit from the subroutine is reached.
  - **Continue debugging** (or **ALT+F5**): continue the script execution stopped by a break command or a breakpoint.
7. To stop debugging, click the **Stop debugging** button in the debug toolbar, or press **CTRL+F5**.



**Note:** Any changes in the source code must be made after the debugging is finished. Before you start debugging the modified scripting, you need to reinstall the capture process on the server, set up the process steps, and create a batch for testing.

### Related Topics:

[“Installing a CaptureFlow” on page 443](#)

[“Editing an installed CaptureFlow” on page 450](#)

[“Setting up CaptureFlow steps” on page 448](#)

## 4.6 Monitoring warnings and errors

The **Problems** panel displays errors and warnings generated for technical errors and user interaction at design time. For example, CaptureFlow Designer generates an error when you add a step to a branch and the step's level exceeds the branch level.

When an error is detected in any element of the currently selected *capture system*, the **Problems** status bar appears. Click the error message link in the **Problems** status bar to focus the step that generated that error. The step displays an error icon.

You can expand the **Problems** panel and view the detailed list of detected errors and warnings.

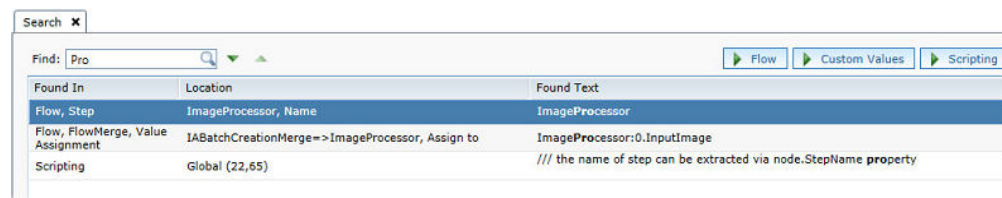
### To view the Problems panel:

1. On the **Problems** status bar, click **More** to expand the **Problems** panel.
2. Click the **Warnings** and **Errors** toggle buttons to view/hide the descriptions of the warnings and critical errors, respectively.
3. Click **Hide** to collapse the **Problems** panel.

After fixing the problem, the corresponding error or warning is cleared from the **Problems** panel. When you fix all errors and warnings, the **Problems** status bar and panel are closed.

## 4.7 Searching for text

The **Search** panel helps you find all inclusions of the specified text in a particular CaptureFlow. Click the **Search** button in the CaptureFlow Designer toolbar, or use the **CTRL+SHIFT+F** hot key.



**Figure 4-10: Text search**

### To find a string pattern:

1. In the **Find** box, specify a sequence of characters to be searched. The search is case-insensitive. Spaces are considered characters to be searched. For instance, you can search for a string like "s" (a space followed by "s"), or for a space or multiple spaces. Wildcards are not supported.
2. Press **ENTER** or click the **Search** icon next to the **Find** box. The search retrieves all inclusions of the specified string pattern from the CaptureFlow chart

---

elements, custom values, and scripting. The search results are displayed in the grid with the following fields:

- **Found In:** Specifies the CaptureFlow Designer tab and the element displayed on that tab where the inclusion is found.
- **Location:** Specifies the exact location of the inclusion, such as a step name, or a script file with the line number(s), or other.
- **Found Text:** Specifies the whole text in which the inclusion is found. The searched pattern is emphasized with bold font.



**Note:** When searching in the **Flow** and **Custom Values** tabs, only the first inclusion found in plain text (such as a comment, or other) is identified and displayed in the grid.

3. To filter the search results in the grid, toggle on/off the **Flow**, **Custom Values**, and **Scripting** buttons. Use each button to show or hide the inclusions found in the respective CaptureFlow Designer tab.
4. Navigate to a particular grid entry using scrolling, or **Show Next** and **Show Previous** buttons located on the panel, or keys **F3** (find next) and **SHIFT+F3** (find previous). The focus set to a particular grid entry opens a respective tab and/or a dialog where you can view the inclusion as highlighted text and do the editing.

If you do the search and then edit your CaptureFlow, the content in the **Search** panel is not updated automatically and may get out of sync with the CaptureFlow. When you move the focus to a grid item that does not indicate a valid inclusion any more, the **Find** box displays the **Refresh** icon and the tooltip tells you to refresh the search results. Press the **Refresh** icon to synchronize the content in the **Search** panel with the updated CaptureFlow.

If you work with several CaptureFlows at a time, the **Search** panel can be used for each CaptureFlow independently. When you switch to a different CaptureFlow, the content in the **Search** panel updates accordingly.



## Chapter 5

# Designing file and email import

### Overview

Import profiles specify the following items for import:

- Image files from directories
- Email and attachments from an email server

The Standard Import module performs the actual import.

## 5.1 High-level steps

“[Creating import profiles: high-level steps](#)” on page 147 provides a description of high-level steps for creating import profiles.

**Table 5-1: Creating import profiles: high-level steps**

Step	Description	See
1	Depending on the type of the import profile, you might need to perform prerequisite steps.	<a href="#">Email Import</a> <a href="#">File System Import</a>
2	Create and save the profile.  You must deploy email and file system profiles to the Standard Import module instances.	<a href="#">Email Import</a> <a href="#">File System Import</a> <a href="#">Deploying Email and File System Profiles to Standard Import Module Instances</a>
3	Upload the profile to the server.	<a href="#">Uploading Service Components</a>
4	In CaptureFlow Designer, create a new or open an existing CaptureFlow that includes the Standard Import step. Set up the step by selecting the import profile that you have uploaded to the server.	<a href="#">Adding a CaptureFlow</a> <a href="#">Creating a process</a> <a href="#">Setting Up Process Steps</a>

## 5.2 Email import profile

The Email type of import profile enables an Intelligent Capture system to set options for monitoring an email box for emails as they arrive. The options include monitoring details and settings for scheduling polls, connecting email servers, processing emails, assembling batches, and processing attachments. The profile settings or configurations determine how Standard Import monitors an email box and creates batches from the emails it finds.

An Email import profile is available to any capture process that contains a Standard Import module step. When Standard Import starts, it reads the profiles from Intelligent Capture Server. Then, based on set configurations, it executes the appropriate profile; polls the email box for emails per the poll scheduler settings, prepares a list of files, and finally creates batches from that list.

Encrypted email, including attached email, is decrypted before being imported, if possible. If an error occurs, then the encrypted email is moved to the \$Error folder as specified in the **Output Root Folder** property.

You can create the profile, which stores these profiles in the server.

### 5.2.1 Creating an email import profile

1. Click **Import**.
2. In the **Import Profiles** list window, click **New**.
3. Specify a profile name following the **naming convention**. For the **Type**, select **Email**.



**Note:** After you create a profile for a specific type, you cannot change its type.

4. Specify the profile options.  
See [“Email profile options” on page 149](#).  
The profiles are saved as \*.config files in the following folder:  
<Capture System Folder>/GlobalData/EmailImportProfile
5. After creating a profile, you must deploy it to each Standard Import module instance.

For more information, see [“Deploying email and file system profiles to standard import module instances” on page 174](#).

### 5.2.1.1 Email profile options

Windows does not allow certain characters in file names. If the following characters are in an attachment's file name, the attachment is still processed, but those characters are replaced with underscores ( \_ ):

:  
"  
/  
\  
|  
?  
\*

“Email profile options” on page 149 provides a description of email profile options.



**Table 5-2: Email profile options**

Option	Description
Common Options section	
CaptureFlow Name	<p>The name of the CaptureFlow process flow that creates batches from emails in the email box that is monitored. The selected process must contain the Standard Import module as one of its batch creation steps.</p> <p>If a CaptureFlow with this name is not available at runtime, batch creation fails.</p>


Option	Description
<p><b>Batch Name Schema</b></p>	<p>The batch naming pattern to generate unique batch names.</p> <p>Batch names, after the schema is resolved, must follow the Intelligent Capture Server batch naming rules.</p> <p>If the schema generates non-unique batch names, Standard Import fails to create batches.</p> <p>The <i>“Variables for batch name schema”</i> on page 173 section provides the list of variables available for creating unique batch names.</p>
<p><b>Batch Priority</b></p>	<p>The priority number that is assigned to each new batch. The number must be in the range of 1 to 99.</p> <p>By default, task priority is defined by the batch priority. You could also set the priority of a specific task by setting the IATaskPriority IA Value. For more information, see <i>“Reserved IA values”</i> on page 487.</p>

Option	Description
<b>Output Root Folder</b>	<p>The folder into which Standard Import saves emails from the designated email box, also known as the watch email box. You can specify any local or UNC path.</p> <p>You can define the folder by using a system environment variable, for example, %USERPROFILE%\Desktop\MyFolder. Alternatively, you can create your custom environment variable in the System Properties of the machine (<b>Advanced &gt; Environment Variables</b>). Type the variable name between %, for example, %variablename%. A system restart is required after defining the system or custom variable.</p> <p>The email messages are first read from the email server and then persisted in the local file system before being moved to the appropriate subfolders within the designated <b>Output Root Folder</b>. This method allows you to retrieve the file data in the case of power outages or any other unrecoverable errors. Furthermore, if the output root folder is inaccessible or does not exist, then the batch associated with the emails is not created; however, execution of the rest of the import continues.</p> <p>The subfolders are created within the designated <b>Output Root Folder</b> to move files depending on the result of the import process:</p> <ul style="list-style-type: none"> <li>• <b>\$NewFiles</b>: All emails are first saved in this subfolder and then moved to \$Skipped, \$Success, or \$Error directory, after being processed. Saving the emails in \$NewFiles allows you to locate the messages even if batch creation fails for some reason.</li> </ul> <p>The emails are saved either in .eml or .msg format with a file name in this format: <b>Message - number .MSG</b> or <b>Message - number .EML</b>. The 'number' suffix, an internally running number, ensures uniqueness in the file names and prevents collision when multiple messages are moved to a batch.</p>


Option	Description
	<p>The body of an email is saved as another file and retained in the <b>\$NewFiles</b> subfolder. This file is never moved to any other subdirectory. The file is created with a name in one of the following formats:</p> <ul style="list-style-type: none"> <li>– <b>MessageBody-number.htm</b></li> <li>– <b>MessageBody-number.txt</b></li> <li>– <b>MessageBody-number.mht</b></li> </ul> <p>Email attachments, if successfully parsed by <b>Email Filter Rule</b>, are also saved in <b>\$NewFiles</b></p> <p>In addition, the new files created using custom scripts are also moved to this subfolder.</p> <ul style="list-style-type: none"> <li>• <b>\$Success</b>: All emails that are successfully processed are moved to a newly created subfolder within this subfolder. The messages are saved either in <b>Message*.EML</b> or <b>Message*.MSG</b> format.</li> </ul> <p>The number of subfolders within <b>\$Success</b> equal the number of successful batches.</p> <ul style="list-style-type: none"> <li>• <b>\$Error</b>: The files that have encountered an error, including email decryption errors, during runtime are moved to a newly created subfolder within this subfolder. The messages are saved either in <b>Message*.EML</b> or <b>Message*.MSG</b> format.</li> <li>• <b>\$Skipped</b>: The files that do not qualify for import (filtered using <b>Email Filter Rule</b> and custom scripting) to a newly created subfolder within this subfolder. The messages are saved either in <b>Message*.EML</b> or <b>Message*.MSG</b> format.</li> </ul> <p>This movement ensures that the skipped messages are not reprocessed during every cycle.</p> <ul style="list-style-type: none"> <li>• <b>\$ExtractedFiles</b>: The extracted files from a zip file are moved to a newly created subfolder within this subfolder. Moves the zip file itself to <b>\$Success</b>, <b>\$Error</b>, or <b>\$Skipped</b> according to the processing result.</li> </ul>


Option	Description
	<p> <b>Notes</b></p> <ul style="list-style-type: none"> <li>• The different file formats enables you to identify how the file is built, whether for the whole email or for the email body or for email attachments.</li> <li>• If you do not find any of these subfolders, it means that some unrecoverable error has occurred. However, you can retrieve the emails from <b>\$NewFiles</b>. When you restore the messages, everything including all attachments also get restored.</li> </ul>
<b>Output Folder Purge Interval</b>	<p>The minimum time interval (in hours) to retain files in the <b>Output Root Folder</b> subfolders: <b>\$Success</b>, <b>\$Error</b>, <b>\$Skipped</b>, <b>\$ExtractedFiles</b>, and <b>\$NewFiles</b>. After the lapse of this interval, a low-priority task runs to delete these subfolders whenever system resources become available.</p> <p>The minimum and maximum values are 0 hours and 2160 hours (90 days), respectively.</p>
<b>Polling Period</b>	<p>The minimum number of seconds to wait before Standard Import checks for emails in the designated watch email box. This wait time is always applied after a poll cycle is completed. If you specify a <b>Polling Rule</b>, it is applied after the lapse of the <b>Polling Period</b>. The combination of <b>Polling Period</b> and <b>Polling Rule</b> determines the polling frequency.</p> <p> <b>Note:</b> If you specify zero or a negative value, a wait of up to 500 ms is enforced to avoid continuous polling.</p>


Option	Description
<p><b>Polling Rule</b></p>	<p>A boolean expression for controlling polling schedules to poll the designated email box for new emails. The <b>Polling Rule</b> is an additional criteria that is applied after the lapse of the <b>Polling Period</b>. A new polling cycle is initiated whenever the expression evaluates to True.</p> <p>If you do not specify a rule, the <b>Polling Period</b> is used as the polling frequency.</p> <p>The “Polling rule variables” on page 160 section provides the list of variables available for creating polling rule expressions.</p>
<p><b>Uses Custom Scripting</b></p>	<p>Select this option if you want to use a custom script (DLLs) for initiating the batch creation process. You can create script actions and associate them with specific events. The script is executed when the associated event occurs.</p> <p>If you select this option and Standard Import fails to locate the script, the process fails.</p> <p>For more details about creating and using custom scripts, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p>
<p><b>Custom Value</b></p>	<p>An optional string value that is used to provide additional custom design-time information to the script at run time. The value is also copied to the batch data (Batch Level 7 IAValue 'ScriptTag') to allow batch scripts to access this configuration.</p>
<p><b>Mail Server Options</b> section</p>	
<p><b>Incoming Email Connection</b></p>	<p>Multiple email connection profiles for incoming email messages. Valid protocols are the following:</p> <ul style="list-style-type: none"> <li>• POP3</li> <li>• IMAP</li> <li>• Exchange WebDAV</li> <li>• Exchange Web Services</li> <li>• Microsoft Graph</li> </ul>

Option	Description
<b>Outgoing Email Connection</b>	<p>The predefined email connection profile for outgoing email messages. Valid protocols are the following:</p> <ul style="list-style-type: none"> <li>• SMTP</li> <li>• Exchange Web Services: in this case, Basic, NTLM, or OAuth authentication should be used.</li> </ul>
<b>Outgoing Sender Address</b>	<p>The 'From' email address for sending read receipts and forwarding email messages.</p> <p>This email address must be the same as specified in the <b>User</b> property in the corresponding outgoing email connection profile.</p>
<b>Email Processing Options</b> section	
<b>Email Filter Rule</b>	<p>A boolean expression that manages email processing. If email parsing expression evaluates to True, the emails are processed for import. If you do not specify a filter rule, the evaluation is always considered as True. The expression can include the following variables:</p> <ul style="list-style-type: none"> <li>• <i>Subject</i></li> <li>• <i>From</i></li> <li>• <i>To</i></li> </ul> <p>To check for an email address in an email's <i>To</i> field that contains multiple addresses, you could use the <b>IndexOf</b> function; for example, if the email's <i>To</i> field includes the following email addresses:</p> <p>DiegoAlias@incapture.onmicrosoft.com, AlineAlias@incapture.onmicrosoft.com</p> <p>Then the following returns true:</p> <pre>IndexOf(To, "DiegoAlias@incapture.onmicrosoft.com", 0, false) &gt;= 0</pre> <ul style="list-style-type: none"> <li>• <i>AttachmentNames</i></li> </ul> <p> <b>Note:</b> <i>AttachmentNames</i> is a comma separated list of names (including extensions) of email attachments.</p> <p>The handling of emails that fail the filter rule are determined by the <b>Keep Emails Excluded by Filter</b> option.</p>

Option	Description
<p><b>Keep Emails Excluded by Filter</b></p>	<p>If selected, then emails that fail the <b>Email Filter Rule</b> and that are set to be skipped by custom scripting are skipped and not deleted from the email server.</p> <p>Keeping skipped emails on the email server allows you to use another email profile to process them.</p> <p>If this option is not selected, then emails that fail the <b>Email Filter Rule</b> and that are set to be skipped by custom scripting are deleted and moved to a subfolder within the <b>SSkipped</b> folder.</p>
<p><b>Saved Email Format</b></p>	<p>The format in which emails are saved in the various subdirectories within <b>Output Root Folder</b>. You can save emails in the following formats:</p> <ul style="list-style-type: none"> <li>• .eml (Outlook Express)</li> <li>• .msg (Outlook)</li> </ul>
<p><b>Skipped Forwarding Address</b></p>	<p>The email address to which the skipped emails have to be forwarded.</p>
<p><b>Success Forwarding Address</b></p>	<p>The email address to which the successfully processed emails have to be forwarded.</p>
<p><b>Error Forwarding Address</b></p>	<p>The email address to which the emails that failed processing have to be forwarded.</p>
<p><b>Batch Assembly Options</b> section</p>	
<p><b>Import Attachments As Pages</b></p>	<p>Select to save attachments at the page level. If selected, one page node is created for each file. By default, attachments are saved at the document level (one document node for each file).</p>
<p><b>Email Sort Order</b></p>	<p>The order in which incoming emails are processed. The valid options are:</p> <ul style="list-style-type: none"> <li>• <b>Oldest First</b> (the default value)</li> <li>• <b>Most Recent First</b></li> </ul>

Option	Description
<b>Email Body Handling</b>	<p>The method in which the body of the email is processed. The valid options are:</p> <ul style="list-style-type: none"> <li>• <b>Discard</b>: Not saved as an attachment</li> <li>• <b>Save as plain-text (.txt)</b>: Formatted as plain text</li> <li>• <b>Save as HTML (.htm)</b>: Formatted as an HTML file</li> <li>• <b>Save as MIME HTML (.mht)</b>: Formatted as an HTML that includes inline images as well.</li> </ul>
<b>Include Headers With Body</b>	<p>Select to include header information at the top of the email body text at the time of saving the email body as an attachment. This option is valid only if you select <b>Email Body Handling</b> method as <b>Save as plain-text (.txt)</b> or <b>Save as HTML (.htm)</b>.</p> <p>Header information includes: From, To, CC, Subject, Date Sent and Attachments (list of names of email attachments).</p>
<b>Maximum Emails Per Batch</b>	<p>The maximum number of email messages to be included in a batch. This value can be between 1 and 1000. Additional emails, if any, are included in the next batch.</p>
<b>Attachment Handling Options section</b>	
<b>Attachment Filter Rule</b>	<p>A Boolean expression that manages the processing of email attachments. During parsing, if the expression evaluates to <b>True</b>, the attachment is processed.</p> <p>You can build the expression using the following variables:</p> <ul style="list-style-type: none"> <li>• Name</li> <li>• ByteCount</li> <li>• PixelWidth</li> <li>• PixelHeight</li> </ul> <p>If you want to process all attachments (that is, no filtering is applied), then enter <b>True</b> for the expression. The default (that is, when no expression is specified for this field) is <b>False</b>.</p> <p> <b>Note:</b> This filter applies only to the outer email attachments and not to the contents of a zip file.</p>

Option	Description
<p><b>Unzip Attachments</b></p>	<p>When this option is selected:</p> <ul style="list-style-type: none"> <li>• Non-encrypted zip files (with the .zip extension) are extracted and the contents are saved as separate attachments. The original zip file is not included in the batch. Only the outer level files within the extracted list are attached.</li> <li>• Encrypted zip files are processed as single attachments and a separate node is created for each of them.</li> </ul> <p>You can find out whether all attachments are unzipped by checking <i>OutputFileType</i> for type ".zip" at the Standard Import attachment level in the CaptureFlow. For example, for attachments saved at document level, the IA Value assignment will be the following: "Assign to <i>StandardImport:1.OutputFileType</i> the value "ZIP"."</p> <p>If you do not select this option, a zip attachment does not get extracted and is saved as a zip file itself.</p> <p> <b>Note:</b> A zip file is processed only if it is parsed successfully by the <b>Attachment Filter Rule</b> expression. The filter rule is applied before processing the zip files.</p>

Option	Description
<p><b>Process Attached Emails</b></p>	<p>Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Handle as Attachments:</b> Select to process the attached emails as attachments only (that is, using <b>Attachment Filter Rule</b>).</li> <li>• <b>Handle as Attached Emails:</b> Select to merge the body of message attachments with the email body. An attached email is saved as an attachment. The <b>Attachment Filter Rule</b> expression is applied before processing the email files. During processing, the body text of the attached email file is appended to the body text of the parent email with a new line between them. If the attached email file has an attached child email file, these are processed recursively.</li> <li>• <b>Save Bodies in Separate Nodes:</b> In addition to being processed as described for the <b>Handle as Attached Emails</b> option, separate nodes with bodies of attached emails are also created. The body of each attached email is processed in the same manner as the body of the main email. Therefore, the attached emails are also processed according to the <b>Email Body Handling</b> and <b>Include Headers With Body</b> options.</li> </ul>
<p><b>Ignore Inline Attachments</b></p>	<p>If this option is selected, then inline attachments, such as email signatures, are excluded from imported emails. This option is applied before the <b>Attachment Filter Rule</b> expression.</p> <p> <b>Note:</b> If a document is inserted into a Microsoft Outlook email as an inline attachment, then it is wrapped in an *.ms0 file and is not considered to be an inline attachment.</p> <p>If this option is not selected, then inline attachments are processed as determined by the <b>Attachment Filter Rule</b>.</p>

Option	Description
<b>Ignore Attachments without Extension</b>	<p>If this option is selected, then attachments without file extensions are excluded from imported emails. This option is applied before the <b>Attachment Filter Rule</b> expression.</p> <p>If this option is not selected, then attachments without file extensions are processed as determined by the <b>Attachment Filter Rule</b>.</p>

### 5.2.1.2 Polling rule variables

You can use the different variables to create polling rule expressions for scheduling a poll. The expressions can use any date expressions based on the “Now()” function and in addition use the variables outlined in [“” on page 160](#).

**Table 5-3:**

Variable	Description
CycleStartTime	The time of the last poll. This time is the local time on the server machine. Initialized to 1970/1/1.
CycleStartAgeSeconds	The number of seconds that has elapsed since the start of the last cycle.
CycleNumber	The cycle number for the last cycle; increases by 1 for consecutive cycles. '0' indicates the first cycle.
CycleSuccess	The success status for the last cycle. 'True' for success and 'False' for failure.
CycleBatchesCreated	The number of batches created during the last cycle.
TotalBatchesCreated	The total number of batches created for all profiles since the server was started.

## 5.3 File system import profile

### Overview

The File System type of import profile enables an Intelligent Capture system to set options for monitoring multiple directories (folders) for image, non-image, and data files, including XML. The options include monitoring details; settings for poll scheduler, file list creation, assembling batches, and creating batches. The profile settings or configurations determine how Standard Import monitors a directory and creates batches from the files it finds in the directory.

A File System import profile is available to any capture process that contains a Standard Import (SI) module step. When Standard Import starts, it reads the profiles from IA Server. Then, based on set configurations, it executes the appropriate profile; polls the folder for files per the poll scheduler settings, prepares a list of files, and finally creates batches from the list.

You can create the profile, which stores these profiles in IA Server.

### 5.3.1 Creating a file system profile

1. Click **Import**.
2. In the **Import Profiles** list window, click **New**.
3. Specify a profile name following the **naming convention**. For the **Type**, select **File System**.



**Note:** After you create a profile for a specific type, you cannot change its type.

4. Specify the profile options.  
See [“File system profile options” on page 162](#).  
The profiles are saved as \*.config files in the following folder:  
<Capture System Folder>/GlobalData/FileImportProfile
5. After creating a profile, you must deploy it on each Standard Import module instance.

For more information, see [“Deploying email and file system profiles to standard import module instances” on page 174](#).


### 5.3.1.1 File system profile options

“” on page 162 describes the file system profile options.



**Table 5-4:**


Option	Description
<b>Common Options</b> section	
<b>CaptureFlow Name</b>	<p>The name of the CaptureFlow that creates batches for files detected in <b>Watch Root Folder</b>. The selected process must contain the Standard Import module as one of its batch creation steps.</p> <p>If a CaptureFlow with this name is not available at runtime, batch creation fails.</p>
<b>Batch Name Schema</b>	<p>The batch naming pattern to generate unique batch names.</p> <p>Batch names, after the schema is resolved, must follow the Intelligent Capture Server batch naming rules.</p> <p>If the schema generates non-unique batch names, Standard Import fails to create batches.</p> <p>The “Variables for batch name schema” on page 173 section provides the list of variables available for creating unique batch names.</p>
<b>Batch Priority</b>	<p>The priority number that is assigned to each new batch. The number must be in the range of 1 to 99.</p> <p>By default, task priority is defined by the batch priority. You could also set the priority of a specific task by setting the IATaskPriority IA Value. For more information, see “Reserved IA values” on page 487.</p>


Option	Description
<b>Output Root Folder</b>	<p>The folder into which Standard Import moves files from the watch folder. The subfolders, if any, are also automatically created under the output folder. You can specify any local or UNC path. If the output root folder is inaccessible or does not exist, then the batch associated with the files is not created; however, execution of the rest of the import continues.</p> <p>You can define the folder by using a system environment variable, for example, %USERPROFILE%\Desktop\MyFolder. Alternatively, you can create your custom environment variable in the System Properties of the machine (Advanced &gt; Environment Variables). Type the variable name between %, for example, %variablename%. A system restart is required after defining the system or custom variable.</p> <p>A unique 'list_id' is assigned to each instance of an import cycle. This ID helps you identify the files that were moved to the output folder during each import.</p> <p>The subfolders are created within the designated <b>Output Root Folder</b> to move files depending on the result of the import process:</p> <ul style="list-style-type: none"> <li>• <b>\$Success</b>: Moves all files that are successfully processed to a newly created subfolder within this subfolder. Names the new subfolder using the convention: &lt;list_id&gt;_B&lt;IA Server batch id&gt;. The number of subfolders within <b>\$Success</b> equal the number of successful batches.</li> <li>• <b>\$Error</b>: Moves the files that have encountered an error during runtime to a newly created subfolder within this subfolder. Uses the unique &lt;list_id&gt; to name the subfolder.</li> <li>• <b>\$Skipped</b>: Moves the files that do not qualify for import (filtered using <b>File Name Filter</b>) to a newly created subfolder within this subfolder. Uses the unique &lt;list_id&gt; to name the subfolder. This movement ensures that the skipped files are not reprocessed during every cycle.</li> </ul>


Option	Description
	<ul style="list-style-type: none"> <li>• <b>\$ExtractedFiles</b>: Moves the extracted files from a zip file to a newly created subfolder within this subfolder. Uses the unique <i>&lt;list_id&gt;</i> to name the subfolder. Moves the zip file itself to \$Success, \$Error, or \$Skipped according to the processing result.</li> <li>• <b>\$NewFiles</b>: Moves the new files created using custom scripts to a newly created subfolder within this subfolder. Uses the unique <i>&lt;list_id&gt;</i> to name the subfolder.</li> </ul>
<b>Output Folder Purge Interval</b>	<p>The minimum time interval (in hours) to retain files in the <b>Output Root Folder</b> subfolders: \$Success, \$Error, \$Skipped, \$ExtractedFiles, and \$NewFiles. After the lapse of this interval, a low-priority task runs to delete these subfolders whenever system resources become available.</p> <p>The minimum and maximum values are 0 hours and 2160 hours (90 days), respectively.</p>
<b>Polling Period</b>	<p>The minimum number of seconds to wait before Standard Import checks for new images in the designated watch directory. This wait time is always applied after a poll cycle is completed. If you specify a <b>Polling Rule</b>, it is applied after the lapse of the <b>Polling Period</b>. The combination of <b>Polling Period</b> and <b>Polling Rule</b> determines the polling frequency.</p> <p> <b>Note:</b> If you specify zero or a negative value, a wait of up to 500 ms is enforced to avoid continuous polling.</p>
<b>Polling Rule</b>	<p>A boolean expression for controlling polling schedules to poll the designated directory, also known as the watch directory, for new image files. The <b>Polling Rule</b> is an additional criteria that is applied after the lapse of the <b>Polling Period</b>. A new polling cycle is initiated whenever the expression evaluates to True.</p> <p>If you do not specify a rule, the <b>Polling Period</b> is used as the polling frequency.</p> <p>The “<a href="#">Variables for polling rules</a>” on page 172 section provides the list of variables available for creating polling rule expressions.</p>

Option	Description
<b>Uses Custom Scripting</b>	<p>Select this option if you want to use a custom script (DLLs) for customizing poll schedule and batch structure. Custom scripts are not used if you do not select this option.</p> <p>If you select this option and Standard Import fails to locate the script, the batch creation process fails.</p> <p>For more information about creating and using custom scripts, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p>
<b>Custom Value</b>	<p>An optional string value that is used to provide additional custom design-time information to the script at run time. The value is also copied to the batch data (Batch Level 7 IValue 'ScriptTag') to allow batch scripts to access this configuration.</p>
<b>Folder Processing Options section</b>	
<b>Watch Root Folder</b>	<p>The main folder to check for incoming files. This folder can contain subfolders.</p> <p>You can define the folder by using a system environment variable, for example, %USERPROFILE%\Desktop\MyFolder. Alternatively, you can create your custom environment variable in the System Properties of the machine (Advanced &gt; Environment Variables). Type the variable name between %, for example, %variablename%. A system restart is required after defining the system or custom variable.</p>




Option	Description
<p><b>File Name Filter</b></p>	<p>The list of valid file name patterns that you must express by using regular expressions (REGEX). This pattern applies to file names and their extensions; the directory name is not considered. The file name pattern is not case-sensitive.</p> <p>The files that do not match the pattern are skipped and moved to a subfolder within the <b>SSkipped</b> folder.</p> <p>You can specify multiple expressions separated by a new line.</p> <p> <b>Note:</b> This filter is not applied to contents of a zip file. The file is selected if it matches with any one of the given patterns.</p>
<p><b>Sort Criteria</b></p>	<p>The file property based on which the files detected in <b>Watch Root Folder</b> are sorted to create the list of files for a batch. The valid options are:</p> <ul style="list-style-type: none"> <li>• <b>Date:</b> Files are sorted by the last modified date.</li> <li>• <b>Name:</b> Files are sorted by file names, including the extensions. The sorting is case insensitive.</li> </ul> <p>If subfolders are present, Standard Import sorts the subfolders first, and then sorts the files within them. This method of sorting ensures that all the files within a subfolder continue to exist as a group.</p> <p> <b>Note:</b> You can also use a custom script to reorder, remove items, or add new items.</p>
<p><b>Sort Descending</b></p>	<p>Select to sort the list of files, chosen for batch creation, in the decreasing order.</p>

Option	Description
<p><b>Unzip Files</b></p>	<p>Select to perform the following:</p> <ul style="list-style-type: none"> <li>• Extract unencrypted files with the ZIP extension. Applicable only if <b>Assembly Method</b> is set to <b>Entire Root Folder Only</b>. Standard Import reads the zip files, extracts them into an output directory, and then creates a single batch for the entire zip file. If <b>Assembly Method</b> is not set to <b>Entire Root Folder Only</b>, then the zip files are processed as any other input file.</li> <li>• Process encrypted zip files as single attachments and create a separate node for each of them.</li> </ul> <p>If extraction fails for any reason, then the files are moved to a subfolder within the <b>\$Error</b> folder.</p> <p>If you do not select this option, all the zip files, irrespective of the folder level, are processed as any other data file. Zip files are processed before the application of <b>File Name Filter</b>. A separate batch is created for each zip file as long as the zip file has at least one entry.</p> <p> <b>Note:</b> Only ZIP formatting is supported. You must use custom scripting to extract files from encrypted ZIP or from other formats such as RAR, 7Zip, and so on.</p>
<p><b>Completed Folder Wait</b></p>	<p>This configuration is applicable only if you have set <b>Assembly Method</b> to <b>Batch Per Subfolder</b>.</p> <p>The minimum period (in seconds) that Standard Import waits for more files after the most recent incoming file has been written into a subfolder under the watch folder. If no new files are added within this period, then writing files into the subfolder is complete and all files needed for batch creation are now available.</p> <p>The value can be between 0 seconds (minimum) and 1 hour (maximum).</p>

Option	Description
<b>Empty Subfolder Purge Wait</b>	<p>The minimum waiting period (in seconds) after which Standard Import removes the empty subfolders within the watch directory.</p> <p>If you set this value to zero, empty subfolders are never deleted.</p>
<b>Batch Assembly Options</b> section	
<b>Import Node Level</b>	<p>The node level at which files are imported from the watch folder. The valid options are:</p> <ul style="list-style-type: none"> <li>• <b>Page:</b> Creates one page node for each file.</li> <li>• <b>Document:</b> Creates one document node for each file and places the file in the document node. For example, if there are five images in the main watch directory, Standard Import imports the files into five document nodes, one image in each document.</li> <li>• <b>Document and Page:</b> Creates a document node and a child page node for each file. Places a copy of the file in both the nodes.</li> </ul> <p> <b>Note:</b> You can use custom scripts to create additional levels or move files to higher levels.</p>

Option	Description
<b>Data File Extension</b>	<p>The file extension type for data files, including XML, that accompany the image files. Standard Import first processes the non-data (image) files as nodes, and then reads the data files with names that match the names of the files associated with each node, and finally creates IA values for those nodes. For example, if the data file extension is 'data', the data file for the image 'NewInvoice.tiff' must be 'NewInvoice.data'. If data file exists but the corresponding node file is missing, the data file is saved as any other file.</p> <p>The data file and the non-data/image file are matched only if they exist in the same directory. If there are more than one image file with the same name as the data file, the data is merged into one of those image files, but which one cannot be predicted.</p> <p>If you do not specify an extension, data files are not processed.</p> <p> <b>Note:</b> The nature of the node, whether page, document, or both is determined by the <b>Import Node Level</b> option.</p> <p>In the data file (except for XML), each <code>name=value</code> pair is specified on a separate line. If the name is an IAValueName, then the IAValue is used. If the <code>name</code> does not match any IAValueName for the node, then a dynamic value is created. The <code>value</code> must be specified according to the data type syntax rules in <i>“Data types” on page 460</i>, except that strings must be enclosed in double quotes.</p> <p>XML files are processed as follows:</p> <ul style="list-style-type: none"> <li>• If either the XML file or the image file exist, but not both at the same time, the files are simply ignored until the counterpart file exists (i.e. if the XML file exists but not the image file, the XML file is ignored by Standard Import until the associated image file exists, and vice versa).</li> <li>• XML data is parsed and saved as IA values in a tag/value pair structure. Each IA value is at the same level as its corresponding tag/value pairs. These</li> </ul>

Option	Description
	<p>pairs are numbered sequentially for each IA node level (0-7). The syntax of these pairs are as follows.</p> <ul style="list-style-type: none"> <li>- Each XML tag structure is assigned to the following IA value:  Tag &lt;{m}&gt;_Level&lt;{n}&gt;</li> <li>- Each corresponding XML value is assigned to the following IA value:  Value&lt;{m}&gt;_Level&lt;{n}&gt;</li> </ul> <p>where:</p> <ul style="list-style-type: none"> <li>- &lt;m&gt; is greater than or equal to 0.</li> <li>- &lt;n&gt; is 0 to 7, inclusive.</li> </ul>

Option	Description
<b>Assembly Method</b>	<p>The method in which Standard Import creates or assembles a batch. The valid options are:</p> <p> <b>Note:</b> If you have selected the <b>Unzip Files</b> option, these options do not apply to the top-level zip files.</p> <ul style="list-style-type: none"> <li>• <b>Entire Root Folder Only:</b> Processes only the files in the watch folder. Skips the subfolders and their contents.</li> </ul> <p> <b>Note:</b> The <b>Unzip Files</b> option is applicable only for this assembly method.</p> <ul style="list-style-type: none"> <li>• <b>Root Folder And Subfolders:</b> Processes files in the watch folder and its subdirectories after recursively combining them into a single list.</li> <li>• <b>Batch Per Subfolder:</b> Creates a single batch for each subfolder under the watch folder if the subfolder has one or more qualified files and skips the top-level files.</li> <li>• <b>Shared Root Folder:</b> Processes the files in the top-level folder to create a batch. Skips the subdirectories and their contents. You can configure multiple profiles to monitor the same watch folder. To enable sharing: <ul style="list-style-type: none"> <li>– Set <b>Assembly Method</b> to <b>Shared Top Level Watch Folder</b> for each profile.</li> <li>– Ensure that the <b>File Name Filter</b> for each profile guarantees that every file in the watch folder is processed by one and only one profile. This restriction applies to both data and zip files.</li> </ul> </li> </ul> <p> <b>Note:</b> Ensure that you enable one and only one profile to process zip files because zip files are not filtered by <b>File Name Filter</b>.</p> <ul style="list-style-type: none"> <li>– Configure all profiles into a single Standard Import module runtime instance or ensure that at any time, all runtime instances are executing.</li> </ul>

Option	Description
<b>Minimum Files Per Batch</b>	<p>When <b>Assembly Method</b> is set to <b>Entire Root Folder Only</b>, <b>Root Folder And Subfolders</b>, or <b>Shared Root Folder</b>, specify the minimum number of files that is required to create a batch.</p> <p>If the number of detected files is below this value, Standard Import does not create a batch.</p>
<b>Maximum Files Per Batch</b>	<p>When <b>Assembly Method</b> is set to <b>Entire Root Folder Only</b>, <b>Root Folder And Subfolders</b>, or <b>Shared Root Folder</b>, specify the maximum number of files to be used to create a batch.</p> <p>If the number of files exceeds this value, the remaining files are included in another batch.</p>

### 5.3.1.2 Variables for polling rules

You can use the different variables to create polling rule expressions for scheduling a poll. The expressions can use any date expressions based on the “Now()” function and in addition use the variables outlined in the following table.

**Table 5-5:**

Variable	Description
CycleStartTime	The time of the last poll. This time is the local time on the server machine. Initialized to 1970/1/1.
CycleStartAgeSeconds	The number of seconds that has elapsed since the start of the last cycle.
CycleNumber	The cycle number for the last cycle; increases by 1 for consecutive cycles. '0' indicates the first cycle.
CycleSuccess	The success status for the last cycle. 'True' for success and 'False' for failure.
CycleBatchesCreated	The number of batches created during the last cycle.
TotalBatchesCreated	The total number of batches created for all profiles since the server was started.

This is an example of a polling rule expression for scheduling a poll every 30 minutes between 8 a.m. and 2 p.m. except on Saturdays and Sundays.

```
(GetDatePart(Now(), 3) >= 8 AND GetDatePart(Now(), 3) < 14) AND
GetDatePart(Now(), 6) <> 0 AND GetDatePart(Now(), 6) <> 6 AND
CycleStartAgeSeconds >= 30 * 60
```

### 5.3.1.3 Variables for batch name schema

The batch name schema that you use to create a batch must be unique when batches are imported into Intelligent Capture Server. A resolved batch name must not exceed 120 characters. You can include these characters in the name: A-Z, a-z, 0-9, space, dash, colon, and underscore. To help you create unique names, you can supply any Intelligent Capture Format Expression function. The [Programming Reference](#) section provides the list of functions.

In addition, you can use the three format tokens for providing unique names as outlined in the following table.

**Table 5-6:**

Syntax	Definition	Examples
<p>Any supported static function in Intelligent Capture Expression Language.</p> <p>[&lt;any text&gt;]{[&lt;Format Specification&gt; ]&lt;Expression&gt;}[&lt;any text&gt;]</p>	<p>The expression language functions allow the user of a GUID or unique time string to be a part of the Batch Name.</p>	<p>See the following two examples:</p> <ul style="list-style-type: none"> <li>“batchName”: “MyBatch_{Tddhhmss Now()}_{NextIndex}” The above expression produces this name on the server: MyBatch_09064934_102600000003</li> <li>“batchName”: “MyBatch_{S CreateGuid(0)}” The above expression produces this name on the server: MyBatch_82fcd238-2fb7-44ac-9acc-a13ce406241d</li> </ul>
{NextIndex}	<p>This variable provides a 64-bit integer unique counter stored in the database on the Intelligent Capture Server. The value is generated upon module-server interaction and cannot be changed manually.</p>	<p>“batchName”: “MyBatch_{NextIndex}” The above expression produces this name on the server: MyBatch_1026</p>
{Machine}	<p>The name of the machine where the current instance is running.</p>	<p>“Batch_{NextIndex}_{Machine}”</p>

Syntax	Definition	Examples
{ProfileName}	The current profile name.	"Batch_{NextIndex}_{Machine}_{ProfileName}"

## 5.4 Deploying email and file system profiles to standard import module instances

After creating Email or File System profiles, you must deploy them to Standard Import module instances. When a profile is created or updated and redeployed to Intelligent Capture Server, then, within a brief amount of time, the profile is deployed to all Standard Import module instances for which the profile has been enabled.

To enable profiles for a Standard Import module instance you simply add the profile names to the module on the command line and restart it. You can add profiles by specifying them directly on the command line or through a profile name configuration file, which you also specify on the command line. Using a profile name configuration file is a convenient way to add multiple Email and File System profile names to each Standard Import module instance. In addition, by using a profile name configuration file, you are not restricted to the maximum Windows command line length.

For the command line syntax, see the `-FileProfileNames`, `-EmailProfileNames`, and `-ProfilesConfig` arguments in *Intelligent Capture Module Reference*.

### To create a profile name configuration file:

1. Click **Import > Import Profiles > Create Configuration File**.
2. Select all the profiles that you want to add and click **Create**.

## Chapter 6

# Designing a profile for the Web Client

### Overview

The **Distributed Capture** profile is used by the Intelligent Capture Web Client to set options for scanning paper documents and importing document image files. These options include scanner settings, image color choices, document assembly, batch settings, and the processing of pages.

After logging into Intelligent Capture Web Client, an operator can select either to create batches (**Create Batches** in the Intelligent Capture Web Client user interface) or process existing batches (**Process Batches** in the Intelligent Capture Web Client user interface). When creating batches, the operator selects a **Distributed Capture** profile and uses its settings while scanning and importing documents and then submits the batch to the Intelligent Capture Server; whereas, when processing batches, an operator selects a batch, which already has a **Distributed Capture** profile associated with it. An operator can override some profile settings, such as scanner settings. Other profile settings further automate processing, such as the following:

- The batch automatically uses the specified CaptureFlow, Image Processing profile, and Recognition project.
- The operator can only select the specified document types.
- Classification and extraction is automatically performed on every page.

### Required and optional components

The following are required and optional components that are used in conjunction with a **Distributed Capture** profile or Distributed Web Client module:

- (Required) For batch creation, a **Distributed Capture** profile that specifies a capture flow, which also includes the Standard Import module in the batch-creating step.
- (Required) For batch processing, a **CaptureFlow** that includes the Distributed Web Client module, which specifies a **Distributed Capture** profile.
- (Optional) Document type.
- (Optional) Image Processing profile.
- (Optional) Recognition project.

### Recommendations and limitations

Intelligent Capture Web Client has the following recommendations and limitations:

- Scanning

- The recommended minimum scan resolution for good quality OCR is 200 DPI.
- The maximum size of an image is 100MB, uncompressed. However, this maximum could be lower depending on the amount of memory that your computer has available at the time of scanning. In most cases, your scanning needs would fall below the maximum image size. Furthermore, the size of an image is determined by the scanned page's dimensions, scan resolution, and color depth. For example, typical—but not absolute—maximums for a page are the following:
  - o A4 or US letter
  - o 600 DPI (scan resolution)
  - o 24-bit color

If a page fails to process because the available memory is exceeded, you would adjust the scan resolution or color depth. For example, you could change the color depth to grayscale.

- If a scanner does not support a profile setting (such as **color, lossless** for **Color Mode**), then a suitable fallback setting is selected for that scanner. In addition, the operator can select a more appropriate scanner setting.
- Supported file formats for importing
  - The maximum file size is 70MB.
  - In addition to plain text, Microsoft Word, and Microsoft Excel file formats, Intelligent Capture Web Client supports the following subset of PixTools for Web file formats:
    - o JPEG (\*.JPG)
    - o Adobe (\*.PDF)
    - o PDF/A (\*.PDF)
    - o PDF/Web (\*.PDF)
    - o Portable Network Graphics (\*.PNG)
    - o TIFF (\*.TIF)



**Note:** Although the most common color depth and compression formats for these file formats are fully supported, the exact color depth and compression formats are determined by the PixTools for Web. For more information, see the PixTools for Web product documentation.

- Supported patch codes:
  - Patch Code type II, III, or T
  - Patch Code type II
  - Patch Code type III

– Transfer Patch Code

## 6.1 Creating a Distributed Capture import profile



- Click **Distributed Capture**, **New**, and then specify the options. See “Options” on page 177.

The profile name must be unique in the **Profiles** list. It must begin with an alphabetic character and include only alphanumeric characters and single spaces. The maximum allowed length is 64 characters.








**Note:** Distributed Capture profiles are saved as XML documents in the following default path: C:/Users/<username>/My Documents/<product\_name\_and\_version>/<Capture System Name>/GlobalData/ScanImportProfile


### Options

“Import profile options” on page 177 provides a description of the import profile options.


**Table 6-1: Import profile options**


Option	Description
<b>Common Options</b> section	
<b>Batch Name Schema</b>	<p>A format expression used to generate a unique name for each batch. Each batch name must be unique. If this field is blank, then a unique batch name is automatically generated.</p> <p>For more information about format expressions, see “Programming Reference—Expressions” on page 457.</p>
<b>Batch Priority</b>	The default priority for a batch.
<b>CaptureFlow Name</b>	<p>The name of the <b>CaptureFlow</b> to which this profile applies.</p> <p>Required for batch creation.</p> <p>Not required for batch processing.</p> <p> <b>Note:</b> The <b>CaptureFlow</b> must specify the Standard Import module in the batch-creating step.</p>
<b>Country or language</b>	Select the countries or languages to use for the full-page OCR processing of multi-language documents that do not have an assigned document type.


Option	Description
<p><b>Departments</b></p>	<p>Specify a comma-separated list of departments for which this profile is available when creating batches. This profile is displayed on the Web Client <b>Create batches</b> tab only for users in the specified departments.</p> <p>If this field is empty, then this profile is available to any user when creating batches.</p> <p> <b>Note:</b> This field only applies to creating batches (that is, in a <b>Standard Import</b> step). To specify the department users for processing batches, you specify the departments in the associated <b>Distributed Web Client</b> step.</p>
<p><b>Scan Only</b></p>	<p>During batch creation, the operator can only scan documents.</p> <p> <b>Note:</b> This option does not apply to batch processing.</p>
<p><b>Scanner Options</b> section</p>	
<p><b>Auto-Detect Page Size</b></p>	<p>Whether to automatically detect the size of the pages to be scanned. Overrides <b>Page Size</b>. The scanner must support this feature.</p>
<p><b>Brightness</b></p>	<p>Adjusts the brightness level as pages are scanned as follows:</p> <ul style="list-style-type: none"> <li>• -3: Automatic</li> </ul> <p> <b>Note:</b> If the scanner does not support Automatic, then Normal is used.</p> <ul style="list-style-type: none"> <li>• -2: Darken</li> <li>• -1: Normal</li> <li>• 0: Lighten</li> <li>• 1 - 100: Matches the scanner's brightness settings. Higher values are brighter.</li> </ul>
<p><b>Color Mode</b></p>	<p>The color format to which to convert.</p> <p> <b>Note:</b> Although lossy scanning produces smaller images, some quality is lost.</p>

Option	Description
<b>Custom Scanner Tags</b>	<p>Semicolon-separated series of custom scanner tags in a name-value pair format as follows:</p> <pre data-bbox="964 447 1109 470">&lt;tag&gt;=&lt;value&gt;</pre> <p>For example:</p> <pre data-bbox="964 537 1260 560">1234=3;456=150/1;789="text"</pre> <p>The value can be one of the following data types:</p> <ul data-bbox="964 653 1425 814" style="list-style-type: none"> <li>• <b>Integer:</b> (Default) An integer.</li> <li>• <b>Rational:</b> A fraction identified by a forward slash (/) between the numbers.</li> <li>• <b>String:</b> Identified by enclosing the value in quotes.</li> </ul>
<b>Page Orientation</b>	<p>Either <b>Portrait</b> or <b>Landscape</b>. If the scanner does not support <b>Landscape</b>, then <b>Portrait</b> is used.</p>
<b>Page Size</b>	<p>The default size of the pages to be scanned.</p>
<b>Resolution</b>	<p>The default resolution (DPI) at which to scan. If the scanner does not support the selected resolution, then the next higher supported resolution is used to ensure data integrity. For example, if the scanner supports 150, 200, or 300 DPI and 175 was selected, then 200 DPI is used.</p>
<b>Scan Source</b>	<p>The default type of paper input to the scanner.</p> <p> <b>Note:</b> <b>Automatic</b> selects either <b>Feeder</b> or <b>Flatbed</b>, depending on the paper's location in the scanner.</p>
<b>Scanner Lock</b>	<p>Operators cannot change scanner settings.</p>
<b>Import Options section</b>	
<b>Imported Image Format</b>	<p>On import, whether to keep the original color format or convert it as follows:</p> <ul data-bbox="964 1556 1425 1724" style="list-style-type: none"> <li>• <b>Keep</b> Keeps the original color format.</li> <li>• <b>Convert</b> Converts it to the color format specified in <b>Color Mode</b>.</li> </ul>



Option	Description
<p><b>Process PDF</b></p>	<p>See the following options:</p> <ul style="list-style-type: none"> <li>• Selected                     <p>Furthermore, the user can manipulate the structure of the PDF by moving, deleting, editing pages, as well as splitting them off from the original PDF to create documents.</p> </li> <li>• Not selected                     <p>On import, the original PDF is not processed and is only passed through as a file for batch submission. Furthermore, the PDF is restricted as follows:</p> <ul style="list-style-type: none"> <li>– The PDF is displayed as a single placeholder and can only be viewed if Adobe Reader is installed.</li> <li>– The PDF can only be manually classified by the user; it cannot be processed through automatic classification and data extraction.</li> <li>– The PDF's pages cannot be modified; for example, its pages cannot be split from the PDF to create a document.</li> </ul> </li> </ul>
<p><b>New Page Processing Options section</b></p>	
<p><b>Classify Pages</b></p>	<p>Select to apply classification to every page. You must specify the <b>Advanced Recognition</b> or <b>Information Extraction</b> project to use in <b>Recognition Project</b>.</p>





Option	Description
<b>Detect Barcodes</b>	<p>If selected, barcode and patch code recognition through the PixTools for Web is enabled with the barcodes and patch codes specified in <b>Patch Code and Barcodes</b>.</p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• If your document types require barcode population through PixTools for Web, then enable this feature; otherwise, disabling it is recommended. Furthermore, if barcode recognition through PixTools for Web is not required, then disabling this feature might improve performance.</li> <li>• This feature does not affect barcode recognition that is provided by scanners; that is, barcodes provided by scanners are always processed.</li> <li>• Barcode recognition through PixTools for Web is also automatically enabled as follows: <ul style="list-style-type: none"> <li>– <b>Document Break Mode</b> is set to <b>Barcode Page</b> or <b>Patch Code Page</b>.</li> <li>– <b>Drop Patch Code Pages</b> is selected.</li> </ul> </li> <li>• When used in a document type field name, the string <b>Barcode &lt;#&gt;</b>, where &lt;#&gt; is an integer of any length, is reserved for use with PixTools for Web. Therefore, even if this option is disabled, PixTools for Web is used and overwrites the server-side value with a blank value when the following conditions exist: <ul style="list-style-type: none"> <li>– The name of the document type field is a string <b>Barcode &lt;#&gt;</b>, where &lt;#&gt; is an integer of any length.</li> <li>– The document type field uses population rules.</li> </ul> </li> </ul>
<b>Extract Content</b>	<p>Select to apply classification and extraction to every page. You must specify the <b>Advanced Recognition</b> or <b>Information Extraction</b> project to use in <b>Recognition Project</b>.</p> <p> <b>Note:</b> When this option is selected, the <b>Classify Pages</b> setting is ignored.</p>

Option	Description
<b>Full Page OCR</b>	Select to perform full-page OCR and generate an OCR data cache. An OCR data cache is required to enable the operator to populate index fields with data (such as amounts, dates, and entire words as well as individual characters) directly from the image by clicking or dragging over the data.
<b>Image Processor Profile</b>	Specify an Image Processing profile name that is used to apply image filters. You must select <b>Process Images</b> .
<b>Patch Code and Barcodes</b>	Specify the patch codes or barcodes that are detected when <b>Detect Barcodes</b> is enabled.
<b>Process Images</b>	Select to apply the Image Processing profile specified in <b>Image Processor Profile</b> to every page. If an Image Processing profile is not specified, then no image filter is applied.
<b>Use auto-learning with</b>	<p>With an <b>Information Extraction (IE)</b> project and classification, extraction, or both features enabled, documents that have been reviewed are sent to the auto-learning service (applicable to IE only). That is, to ensure that valid data is learned, only documents that have been loaded into the client's <b>Review</b> step are learned—even if they are subsequently skipped.</p> <p> <b>Note:</b> The REST Server does not perform Production Auto-Learning (PAL) for the Web Client like it does for Information Extraction (IE). Instead, it collects the information needed for the Collector module and adds it to the batch. You must add a Collector step to the CaptureFlow and set the IA values to use the new output from the Web Client.</p> <p>If you select an Advanced Recognition (AR) project, the only learning options are:</p> <ul style="list-style-type: none"> <li>• <b>None:</b> No learning is performed.</li> <li>• <b>Both:</b> Learning to classify and extract.</li> </ul>
<b>Recognition Project</b>	<p>Select <b>Advanced Recognition</b> or <b>Information Extraction</b>.</p> <p>If <b>Use auto-learning with</b> is selected, then <b>Information Extraction</b> <i>must</i> be specified.</p>

Option	Description
OCR Engine	Advanced Cloud OCR and CRE are available only when classification or extraction is enabled with an information extraction-based project.
<b>Batch Assembly Options</b> section	
Allowed Document Types	<p>Semicolon-separated list of document type names used for manual selection of the document type. If the list is empty, then no document types are available for the operator to select. Use an asterisk (*) as a wildcard to append to the end of a string. Specifying an asterisk by itself displays all document types.</p> <p>Other modules (such as Completion and modules with scripting) can send tasks to be processed by a Distributed Web Client module. Therefore, the document types used in the modules that you expect to send tasks to the Distributed Web Client modules using this profile should be specified as allowed document types.</p>
Blank Page Tolerance	<p>To set the general amount of noise that can be on an otherwise blank page for it to still be considered blank, select one of the following choices:</p> <ul style="list-style-type: none"> <li>• <b>Pristine white:</b> No noise is tolerated.</li> <li>• <b>Dirty white:</b> Some noise is tolerated.</li> <li>• <b>Very dirty white:</b> A lot of noise is tolerated.</li> <li>• <b>One line OK:</b> One line of text is tolerated.</li> <li>• <b>Two lines OK:</b> Two lines of text are tolerated.</li> </ul> <p>Blank page detection is used in <b>Document Break Mode</b> and <b>Drop Blank Pages</b>.</p> <p> <b>Note:</b> An exact number of non-white pixels is not used to determine a tolerance; rather each choice attempts to approximate realistic conditions.</p>
Default Document Type	Select the default document type to be assigned to unclassified documents. It must be a document type in <b>Allowed Document Types</b> .
Delete Empty Nodes	Select to discard empty, non-page nodes from the batch when scanning is complete.

Option	Description
<b>Document Break Codes</b>	Select the barcodes and patch codes on which to break a document when <b>Document Break Mode</b> is set to <b>Barcode page</b> or <b>Patch code page</b> .


Option	Description
<p><b>Document Break Mode</b></p>	<p>The kind of page or event that starts a new document.</p> <p>The first page in the initial scan or import of a batch always starts a new document and thereafter automatic document breaks are applied according to this option's settings.</p> <p>After the initial scan, if the operator wants to scan or import more pages and also wants automatic document breaking to be performed, then the operator must do one of the following before starting the scan or import:</p> <ul style="list-style-type: none"> <li>• Highlight the last page in the batch (that is, the last page of the last folder's last document).</li> <li>• Do not highlight any folders or documents.</li> <li>• Highlight a folder or document without any pages.</li> </ul> <p>Otherwise, if any page—other than the last page in the batch—is highlighted, no automatic document breaking is performed; that is, all pages are inserted into the current document after the highlighted page.</p> <p> <b>Note:</b> If multiple pages have been selected, then automatic document breaking is determined by the first page that was highlighted. For example, if the operator selects the last page in the batch first and then another page, then automatic document breaking is performed.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>• <b>None:</b> The batch is not automatically divided into documents.</li> <li>• <b>Page Count:</b> Enter the number of pages after which a new document is started in <b>Document Page Count</b>.</li> <li>• <b>Blank Page:</b> A blank page is scanned.</li> <li>• <b>Barcode Page:</b> A page with a barcode is scanned. Specify the barcodes on which to break in <b>Document Break Codes</b>.</li> </ul> <p> <b>Note:</b> When this value is selected, barcode recognition through CCT is</p>



Option	Description
	<p>also enabled as described for <b>Detect Barcodes</b>.</p> <ul style="list-style-type: none"> <li>• <b>Patch Code Page:</b> A patch code page is scanned. Specify the patch codes on which to break in <b>Document Break Codes</b>.</li> </ul> <p> <b>Note:</b> When this value is selected, barcode recognition through CCT is also enabled as described for <b>Detect Barcodes</b>.</p> <ul style="list-style-type: none"> <li>• <b>Classification:</b> When classification results in a new document type being applied to a page.</li> </ul>
<b>Document Page Count</b>	Automatically set the number of pages that each document is to contain. After a document reaches the specified number of pages, then a new document is started. You must specify <b>Page Count</b> in <b>Document Break Mode</b> .
<b>Drop Blank Pages</b>	Select to discard blank pages from the batch. If <b>Document Break Mode</b> is set to <b>Blank Page</b> , then the page is dropped after the document breaks.
<b>Drop Patch Code Pages</b>	<p>Select to discard patch code pages from the batch. If <b>Document Break Mode</b> is set to <b>Patch Code Page</b>, then the page is dropped after the document breaks.</p> <p> <b>Note:</b> When this option is enabled, barcode recognition through CCT is also enabled as described for <b>Detect Barcodes</b>.</p>
<b>Require Classification</b>	<p>If selected, then all documents must be classified before submission to the server.</p> <p> <b>Note:</b> If the <b>Move Pages and Documents</b> permission is not selected for batch processing, then this option is ignored.</p>
<b>Require Valid Documents</b>	<p>If selected, then all data must be validated before submission to the server.</p> <p> <b>Note:</b> If the <b>Review</b> step is not available, then this option is ignored.</p>

## 6.2 Setting up batch processing

To set up batch processing in Intelligent Capture Web Client, you add the Distributed Web Client module to a capture flow and set up the module's properties. "Distributed Web Client module properties" on page 187 provides a description of the Distributed Web Client module properties.

**Table 6-2: Distributed Web Client module properties**

Property	Description
<b>Common</b>	
<b>Profile</b>	Specify a capture profile to use.   <b>Note:</b> To specify the department users who can process batches with this capture profile, specify the departments by right-clicking the Distributed Web Client step and then selecting <b>Departments</b> .
<b>Permissions</b>	If none of the permissions in this section are selected, then operators have the following permissions: <ul style="list-style-type: none"> <li>• <b>Scan</b> step (only for loading; that is, no scanning or importing)</li> <li>• <b>Organize</b> step</li> <li>• <b>Submit</b> step</li> <li>• View images</li> <li>• Flag images</li> <li>• Change document types</li> </ul>
<b>Edit images</b>	Enables operators to edit images.
<b>Scan or Import Files</b>	Enables operators to scan documents and import computer files.
<b>Copy Pages and Documents</b>	Enables operators to copy pages and documents.
<b>Move Pages and Documents</b>	Enables operators to move pages and documents, including splitting and merging documents as well as changing document types.
<b>Delete Pages and Documents</b>	Enables operators to delete pages and documents.
<b>Review and Edit Data Fields</b>	Enables operators to make changes to the data in fields.

Property	Description
<p><b>Flagging</b></p>	<p>Operators use flags to indicate documents, pages, or fields that have issues which can be reviewed and fixed at a later date. Flags are displayed as an orange flag (  ) on a page, an orange flag above a document thumbnail, or with an orange outline around a field.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• Using the options in this section to set up flagging is only applicable to batch processing mode. Although you can use server-side scripting to implement flagging for batch creation mode, operators cannot modify such flags nor correct their issues.</li> <li>• Whereas errors or unclassified documents could prevent batch submission—when <b>Require Valid Documents</b> or <b>Classification Required</b> is set—flags do not. In fact, you could flag all field-level errors, unclassified documents, or both to submit the batch.</li> </ul>
<p><b>Flag reasons for fields</b></p>	<p>Create flag reasons for operators to use to flag fields with issues to review and fix later.</p>
<p><b>Flag reasons for pages</b></p>	<p>Create flag reasons for operators to use to flag pages with issues to review and fix later.</p>
<p><b>Flag reasons for documents</b></p>	<p>Create flag reasons for operators to use to flag documents with issues to review and fix later.</p>

## Chapter 7

# Designing image processing

You design image processing profiles that make changes to the images and their output quality as well as retrieve information and features from the images. They can be used for:

- Enhancing images by creating an image based on previously defined filters combined into profiles.
- Detecting some properties from the nature of the image.
- Extracting specific data such as barcodes or patchcodes.

### What is an Image Processing profile?

Image Processing profiles are created and configured in the Image Processing editor and use special scripts or third-party image filters. For more information, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

The profile is a sequence of *image filters with settings* and *annotations* optionally applied on the top of the final filter result. Users can set up various Image Processing profiles according to their needs to achieve some specific image processing goal, for example, improving the quality of faxed images.

Image processing profiles let you apply filters to test images and view results immediately. Thus, you gain full control on each setting and the time that the image processing step may take. Later the Image Processing profiles can be used by other components, for example, for exporting enhanced images:

- To test classification and recognition accuracy using Recognition Designer.
- To use in Completion and test the manual data entry.

### How does image processing work?

Images processing functionality is represented by the Image Processor module. The module can process *batches* after Image Processing step is set up and configured. The Image Processor step is added to a process using CaptureFlow Designer and is later set up by selecting one of the previously customized Image Processing profiles which were uploaded to the capture server.

Image Processing step is set up in CaptureFlow Designer by selecting one of the available Image Processing profiles previously created using the **Image Processing** area.

Image Processing profiles are stored on the machine where the Image Processor module is installed. Profiles are saved as XML documents in the following folder:

Documents/<product\_name\_and\_version>/<Capture System Name>/GlobalData/ImageProcessing

## 7.1 Managing Image Processing profiles

Image processing profiles are task-oriented and intended to perform the following tasks:

- **Enhance images quality:** create an image.
- **Detect images:** enhance images often requiring a region for which the detection is applied. Filters of this type do not modify the images though some of properties are detected based on the nature of the image.
- **Extract data:** extract some information from the image, for example, barcodes. Filters of this type do not need a region specified.


The **Image Processing Profiles** list displays all image processing profiles existing in the currently selected **Capture System** and saved on the developer's machine. As with other types of profiles, these profiles are global and can be used in any Intelligent Capture process.

View the **Image Processing Profiles** list by selecting the **Image Processing** area.

The following table describes the layout and options available in the list window.

**Table 7-1: Image Processing profiles window**

Element	Description
Section Bar	<p>Displays buttons for each open profile as well as the <b>Image Processing Profiles</b> list. Click a section button to navigate to a different profile or to return to the <b>Image Processing Profiles</b> list. Close individual profiles by clicking the <b>X</b> button on the section bar for the profile.</p> <p>If the number of open profiles is such that some buttons are hidden from view, paging controls and a drop-down button appear on the right side of the section bar. Click &lt;&lt; or &gt;&gt; to scroll left or right, respectively. Or, click the drop-down button to select from a list of open profiles or the <b>Image Processing Profiles</b> list.</p>
<b>Image Processing Profile</b>	<p>The column displays names of all Image Processing profiles existing in the Capture System. To open a profile, double click it in the list.</p>

Element	Description
<b>Description</b> column	For the selected Image Processing profile, the column contains its description or usage instructions. To edit a description of the profile, click the existing description.
<b>Last Saved</b> column	For the selected Image Processing profile, the column contains the date when this profile was last saved on the designer's machine.
<b>Delete</b> button	Deletes the selected Image Processing profile from the capture system and physically from the disk of your <i>PC</i> .   <b>Note:</b> If you delete a profile that is set up for an Image Processor step in a process, the system will not be able to run this step and process batches. The corresponding error will be shown.
<b>New</b> button	Use the button to create an Image Processing profile. For more information on Image Processing profiles naming restrictions, see <a href="#">“Adding a new profile” on page 263</a> section.
<b>Open</b> button	Use the button to open the selected Image Processing profile for viewing or editing its settings.

The number of selected profiles is displayed in the lower-left part of the profiles list while the number of available profiles – in the lower-right corner. Use the arrow keys to navigate between the profiles in the **Image Processing Profiles** list.


### Related Topics

[“Designing an image processing profile” on page 191](#)

## 7.2 Designing an image processing profile






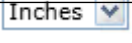
You design an image processing profile by creating filters and annotations that you apply to a set of test images.





1. To specify a folder that contains test images to which to apply the profile, click the path displayed in the **Test Images** field.

 **Note:** Only TIFF images are supported.

2. To design an optimal image processing profile, you add filters and set their properties, arrange them in the desired sequence, and evaluate them against a set of test images as outlined in the following table.

**Table 7-2: Filters tab properties**

Action	Icon/Tooltip	Description
<b>Filters</b>		
Add a filter	 <b>Add new filter</b>	To add a filter, on the <b>Filters</b> tab, click <b>Add new filter</b> and then click a filter and specify its properties.  See <i>“Image filters reference” on page 200.</i>
Delete a filter	 <b>Remove selected filter</b>	Select a filter and press <b>DELETE</b> .
Select or clear a filter	  <b>Select previous filter</b> <b>Select next filter</b>	Click a filter.
Expand or collapse a filter’s properties	N/A	Double-click a filter.
Expand or collapse the properties of all filters	 <b>Expand or collapse all filters</b>	N/A
Set a filter’s unit of measurement		Specifies the unit of measurement that a filter uses for setting up a region or the coordinates for previewing its properties (and any other measurement settings for the Image Processing profile).
<b>Test Images</b>		
Load test images into the profile editor	<b>Test Images</b>	Click the <b>Test Images</b> path link and browse to the folder containing the test images. Thumbnail images are displayed below <b>Test Images</b> .

Action	Icon/Tooltip	Description
Load an image for testing filters	Thumbnail	<p>Click a thumbnail.</p> <p>The selected image is displayed in the <b>Before</b> and <b>After</b> areas.</p> <p> <b>Note:</b> If the image is unsupported or corrupt, then its thumbnail is changed to a red X.</p>
Compare image results with different filters applied	<p><b>Compare Side-By-Side</b></p> <p><b>Compare Top-Bottom</b></p>	<p>If you apply new filters to the selected image, then the image before the new filters were applied is displayed in the <b>Before</b> area and the image with new filters applied is displayed in the <b>After</b> area.</p> <p>The image processing time with the new filters is displayed next to the file name.</p> <p>See “Evaluating image processing quality with test images” on page 199.</p>
Display the original image without any filters applied	<b>Original</b>	N/A
Display the final image with all filters applied	<b>Final</b>	N/A
Zoom the image in and out	<p>100%  </p> <p><b>Increase zoom / Decrease zoom</b></p>	N/A 

Action	Icon/Tooltip	Description
Save the resulting images to a folder	<b>Save Results</b>	The resulting test images are saved as single-page files. To avoid overwriting the <b>Test images</b> files, the resulting images are saved in a directory at the same level as the <b>Test images</b> directory; that is, <test-images-path>/<last_dir_name>-results. Any identically-named files in this directory are overwritten. Each resulting image file has the same name as its original.  Annotation text is saved to an XML file.

Use the following *filter categories*:

- **Detect Barcodes/Blank pages/Color Marks/Colorfulness/Patchcodes**: a single detection filter that verifies the type of desired content.
- **Remove Background/Black Bars/Holes/Lines**: a set of filters that lets you remove different distractions and redundant elements of the image.
- **Adjust Overall Colors/Specific color, Convert to Black-White (Advanced), and Invert black-White**: a set of filters which allows you to invert colors or perform specific color adjustments to ease further image processing and enhancement.
- **Adjust Lighting/Thickness, Remove Specks, and Smooth Edges**: The **Remove Specks** filter allows you to specify the noise width and height in order for the object to be considered a speck. The **Smooth Edges** filter removes bumps and spurs that appear on text characters or graphics objects in an image.
- **Crop/Deskew/Rotate/Scale**: A set of filters which allow to modify and correct the page layout and its properties.

For a complete list of filters, see [“Image filters reference” on page 200](#).



**Note:** In addition to the predefined filters, you can develop *custom image filters* to implement more intelligent filtering behavior or wrap third-party filters. A custom filter deployed on your Designer machine is available in the **Custom Filters** category list.

To learn more about how to develop and deploy custom filters, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

## 7.2.1 Adding filters

Once you have created an image processing profile and opened it, you can add filters.

### To add a filter to the profile:

1. Navigate to the **Filters** tab, click the green plus button, expand the required set of filters, and select a filter. For a complete list of filters and their description, see [“Image filters reference” on page 200](#).
2. Select a test images directory, wait until the test images are loaded, and select one of the images in the thumbnail view. The selected test image now has a red border around it and is displayed in the preview area, **Before** and **After** areas.
3. Specify the required filter properties and add as many filters as you need. As you go on adding filters, you can preview their results in the **After** area of the screen. The number of filters you specify for the profile appears in the **Filters** tab name.
4. Click the **Save** or **Save All** button on the main toolbar to save changes to filters.



**Note:** After you click the **Save Results** button, filters are applied to all images in the test directory, and resulting test images are exported as single-page files in a directory next to the image sources directory. If such a directory already exists and contains processed images, files in this folder are overwritten; each image file has the same name as the original. *Filters settings* are saved together with Image Processing Profile. Annotations texts are saved to an XML file.

For the full list of supported image and non-image file formats that the Image Processor module can handle, *OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)*. For annotations, use the same file formats as for Completion described in *OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)*.

## 7.2.2 Reordering filters

After you have selected several filters to be applied to the test images, you may need to reorder them some time later. For example, if you start with color images, then *color filters* should be applied first. Otherwise, if you use some *binary filters*, all subsequent filters will apply to a monochrome image.

### To reorder filters:

1. Navigate to **Image Processing Profiles** list, select and open a required profile.
2. Navigate to the **Filters** tab and click a desired filter in the filters list.
3. Drag the filter to a new place in the filters sequence.


### 7.2.3 Removing a filter

Use the procedure below to remove one or more filters you no longer need.

**To remove a filter:**


1. Navigate to the **Image Processing Profiles** list, then select and open the desired profile.
2. In the **Filters** list, select the filter that you want to delete.
3. Delete the filter by doing one of the following:
  - Click the red **X**.
  - Select the **Delete** command from the toolbar.
  - Press the **DELETE** key.

The selected filter is deleted from the local machine.


 **Note:** Removing a filter does not remove an Image Processing profile.

## 7.3 Annotating images

An annotation is a special filter that places the specified text or image over the captured image. If the Image Processing profile defines annotations, they are added to the captured images in production after all other filters have been applied.

 **Note:** If an Image Processing profile contains at least one annotation, then all initial annotations of the input image are overwritten during the Image Processor step and the output image contains only annotations added by the Image Processing profile.

Annotations can only be applied to *TIFF* images.


 **Note:** When applying an annotation on a TIFF image, the Image Processor module does not modify the image sector of the file. The annotation is appended as metadata next to the image data. For this reason, some image viewers may display an annotated image without annotations. To burn an annotation to the image, you need to have your TIFF image additionally processed by Image Converter, with the **Merge Annotations** property set in the image conversion profile. For more detail about image conversion, see [“Image Conversion profiles editor” on page 232](#).

**To add annotations to the profile:**


1. Navigate to the **Image Processing** area, locate and open a required image processing profile or [create one](#).
2. (Optional) [Load the test images](#) if they are not displayed.

3. Navigate to the **Annotations** tab.
4. Click the icon in the **Add** section to select the type of annotation from the options provided in the following table.

**Table 7-3: Annotations palette: annotation types**

Annotation type	Description
Redact	A rectangular area on the image. Is generally used to hide sensitive information.
Highlight	A rectangular area on the image. Is generally used to highlight and attract visual focus to a specific area.
Arrow	A line on the image. The line style can be configured, including the line cap (endpoint), color, and thickness.
Stamp	<p>A selected image (a stamp) placed on the page. The stamp file can be picked from the list of recently selected images, or you can browse for an image file.</p> <p> <b>Note:</b> A stamp image must be opaque and can have any format, compression, and color depth. Transparent stamps are not supported and, if applied, appear in the image viewer as black opaque rectangular stamps.</p>
Text	The specified text overlaid on the image.
Comment	<p>The specified text added on the image as a popup comment. Comment annotations display when you hover over the annotation with the mouse cursor.</p> <p>An icon displays on the image to identify the location of the popup comment. The comment icon is for visual purposes only and is not saved with the image.</p>

5. In the **Edit** section, specify the properties of the selected annotation.

 **Note:** For annotations of type **Text** and **Comment**, the **Text** property can specify a format expression with IA values. For more detail, see [“Adding IA values to the annotations” on page 198](#).

6. To place your annotation on a page, keep the right mouse button pressed and draw a rectangle over the selected test image. Release the mouse button.

A new annotation appears as a rectangle area on the test image. The **Annotations** tab button displays the actual annotations counter in round braces.

7. (Optional) To modify the position or the size of an annotation on the page, move or resize the rectangle area with a mouse.  
To navigate between annotations placed on the image, use the **TAB** key or arrows.
8. (Optional) Repeat the above steps to add more annotations to the profile.
9. (Optional) You can copy the existing annotation and paste it on the same image or on a different test image. The copied annotation has the same property values as the original one. Select the annotation that you want to copy and click **Edit > Copy (Annotation)**. On the destination page, select **Edit > Paste (Annotation)**. The annotation is placed at the center of the destination page and can be moved or modified as necessary.



**Note:** You cannot copy and paste annotations between different Image Processing profiles.

10. (Optional) You can apply the annotate the test images
11. Click the **Save Results** button to save the Image Processing profile.



#### Notes

- You cannot undo and redo your actions when working in the **Annotations** tab.
- If the original image has different horizontal and vertical DPIs, they get changed to the same *DPI* after processing the annotated image.

### 7.3.1 Adding IA values to the annotations

The text of the annotation can be a format expression with *IA* values. In production, the *IA* values are resolved at runtime and inserted in the text of the annotation as values.

To insert an *IA* value in your annotation, use the following syntax:

```
{step_name:level.value}.
```

#### To add *IA* values to annotations:

1. Create an Image Processing profile, navigate to **Annotation** tab and type a desired *IA* value in the **Pop-up Text** area.
2. Type the *IA* values expressions in curly brackets { }.  
For example, to add the *Doc Control Number*, type `{CustomValues:0.DCN}`. Expressions are evaluated at runtime, and their values appear in annotation text.



**Note:** Autocompletion is not supported when editing text for annotations. When setting up the particular Image Processing profile, remember that filters are not aware of the CaptureFlow process settings; thus, you have to know the desired IA values.

3. Click **Save Results** button to save changes to the filter annotations.
4. Upload the changes to the profile to the Capture Server.
5. Set up a process using an Image Processor step and select a desired Image Processing profile containing IA Value in annotation.
6. Create a batch and process it running the Image Processor step.
7. View the results in Completion, **Annotations** view.

## 7.4 Evaluating image processing quality with test images

A number of sample images for testing image processing filters configurations are provided. You can select a different directory to test the filters of specific types using particular test images.

You should also evaluate the processing speed of the filters. Some filters enable you to define image regions, which might improve performance. Furthermore, some filters such as **Deskew** and **Detect Barcodes** can provide a good recognition result but require more time to process, depending on the image size and format.

### To select a directory containing sample images:

1. Create an Image Processing Profile or select an existing one.
2. Click the **Test Images** link. The **Browse for Folder** dialog opens.
3. In the dialog, navigate to a desired folder containing test images with *TIFF* format and click **OK**.  
Previews of images from a new directory appear in the **Test Images** area.

You have a number of options for previewing the results of applying image filters, namely:

- Change the *preview area* of before and after images dragging the image to the desired location.
- Change the *zoom* of previewed images using the zoom in and zoom out buttons and zoom slider.
- Select the desired *preview mode*.

### To view and compare image processing results

1. Click the desired test image in the image preview area.

A red border appears around the selected image in the sample images thumbnail view, and it is scrolled to the center in the **Before** and **After** results area.

2. Drag to the desired location of the test image to view the processed images.
3. Select the desired zoom and preview mode.

After you have added the required filters and modified their properties, you can save the results. To do so, click the **Save Results** button under the test images thumbnail area. All images in the test folder are saved and the selected filters are applied after you click the **Save Results** button. All images are saved as single page images in the `<Images Folder Name> - results` folder.

## 7.5 Image processing filters usage example

Imagine that the company has received an invoice containing toner smudges, wrinkles, coffee stains, and that it is skewed. The scanned or imported image cannot be processed by the Completion operator, thus, you would need to create an Image Processing Profile using a number of filters available in **Image Processing**.

To improve the recognition quality of the image, you can create an image processing profile with the following filters:

- Deskew to straighten skewed image
- Remove Specks
- Remove Noise to clean up the stains on the image



**Note:** If you pass a color image to the Image Processor module during processing and you have a binary filter selected in the profile, the module will convert it to binary (black and white) image before processing it.

## 7.6 Image filters reference

This section provides a detailed description of Image Processing filters and their properties.

All Designer system filters support *TIFF* images only. The output format is always TIFF of the same color format and compression except when filter logic changes the color format.



**Note:** The third-party filters included with Image Processor may have the capability to support input and output image formats other than TIFF, although it cannot be guaranteed.

For “region” and “distance” types, measurement units are defined by user preferences in design mode, but settings are always stored in inches. For example, “region” data might be stored as 0, 0, 1.25, 2. Boolean type values are represented by the string `true` or `false`.

For information on the scripting reference for these filters, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

## 7.6.1 Detection filters

A set of detection filters, which verify the type of content on an image.

### 7.6.1.1 Detect barcodes

A filter enabling detection of various types of barcodes; any number of barcodes can be detected on one page.



**Note:** Test the filter on images similar to those used in production mode.

In some cases you may need to perform additional image processing before applying the barcode filter. For example, you might want to use the following filters:

- **Deskew:** to straighten skewed images.
- **Remove Noise:** to clean up dirty images.
- **Remove Lines:** to repair the lines in the barcode itself.
- **Smooth Edges:** to smooth the lines in the barcode itself.

“[Detect Barcodes filter properties](#)” on page 201 provides a description of barcode filter properties.

**Table 7-4: Detect Barcodes filter properties**

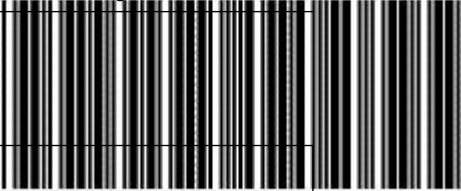
Property name	Description
Mode	See the following options: <ul style="list-style-type: none"> <li>• <b>Enhanced:</b> Provides better results by performing additional image preprocessing; however, processing is slower.</li> <li>• <b>Normal:</b> Enables quick barcode detection.</li> </ul>
Orientation	The orientation of the barcodes that are to be detected. <b>Note:</b> Any means horizontal, vertical, or diagonal.
Decode	Whether to decode barcode results into readable strings.
Exact Length	Whether barcodes with a specific number of characters only are to be detected; that is, only barcodes having this length would be in the filter’s output. Specify the number of characters in the <b>Characters</b> property.

Property name	Description
Characters	The number of characters to be used with the <b>Exact Length</b> property.
Min. height	The minimum height (in inches) of the barcode.
Scan Distance (1D)	<p>The scan distance (in pixels) between line sweeps. Useful when searching for 1D type barcodes.</p> <p>Reducing the value improves the detection of barcodes that are short relative to their height.</p>
Barcode Types	<p>Select the types of barcodes to be detected.</p> <p>For a description of the different barcode types, see <a href="#">“Supported barcode types” on page 202</a>.</p>
Use Checksum	Whether checksums are used.
Use Region	<p>Whether a region on the image in which barcodes are to be detected is to be used. Specifying a region for barcode detection can improve the detection process. The format describes a rectangle where the x and y coordinates (in inches) specify the top left corner to the bottom right corner of a rectangle as follows:</p> <p>x-top-left,y-top-left, x-bottom-right,y-bottom-right</p>

**7.6.1.1.1 Supported barcode types**

“Supported barcode types” on page 203 describes supported barcode types and examples.


**Table 7-5: Supported barcode types**

Barcode type	Description	
Code 39	<p>Code 39 is an alphanumeric barcode. This 1D barcode is used in industrial application and has both original and extended versions.</p> <p>The original version enables to encode 43 characters including digits 0 to 9, letters A to Z, 6 symbols, plus one special character (*) that marks the beginning and end of the barcode. This character is not read during recognition.</p> <p>The extended version enables encoding of all <i>ASCII</i> table characters (128 characters). The 39 barcode has a variable, bidirectional length. Its name comes from its structure, 3 of 9 and is sometimes called Code 3 of 9 code or USD-3. Each character is encoded by 9 elements (5 bars, 4 spaces), of which 3 are large (1 binary) and 6 straight (0 binary). All characters are separated by a space, which are not counted as characters.</p>	 <p>*ABCDEF*</p>

Barcode type	Description	
1D barcode	1D barcodes store a short message, such as a room number, customer number or serial number.	 <p>Airline 2 of 5 CODABAR UPC-A UPC-E Interleaved 2 of 5 EAN-8 EAN-13 Matrix 2 of 5</p>
2D barcodes	2D barcodes such as <i>PDF417</i> , QR, and DataMatrix store large amounts of data such as the description of the content of a package for example. The PDF417 barcode is the most widely used 2D barcode. Its capacity is 1850 ASCII characters.	 <p>Datalogic 2 of 5 QR DataMatrix</p>

### 7.6.1.2 Detect patch codes

A binary filter enabling detection and decoding patch codes on images.

 **Note:** Test the filter on images similar to those used in production mode.

In some cases you may need to perform additional image processing before applying the filter. For example, use the following filters:

- **Deskew:** to straighten skewed images.
- **Remove Background:** to clean up dirty images.
- **Remove Lines:** repair the lines in the barcode itself.
- **Smooth Edges:** to smooth the lines in the barcode itself.

“[Detect patch codes filter options](#)” on page 205 provides a description of patch codes filter options.

**Table 7-6: Detect patch codes filter options**

Property	Description
Detect Direction	Direction to detect the patch code: <ul style="list-style-type: none"> <li>• <b>Vertical:</b> Detects vertical patch codes</li> <li>• <b>Horizontal:</b> Detects horizontal patch codes</li> <li>• <b>Both:</b> Detects both horizontal and vertical types of patch codes</li> </ul>
Min Bar Width	Minimum width of a patch code. Valid values are a fractional decimal value ranging from 0 to 100.
Max Bar Width	Maximum width of a patch code. Valid values are a fractional decimal value ranging from 0 to 100.
Use Region	A region you can select to apply the barcode to improve the patch code detection process.

### 7.6.1.3 Detect blank pages

A filter enabling detection of blank pages using *Preset* and *Black Area Ratio* methods and specifying margins to exclude from the image when determining if the image is blank.

You can specify the precise number of black pixels to consider the page as blank. Ratio is the amount of black pixels divided by the amount of all region pixels.

“Detect blank pages filter properties” on page 206 provides a description of blank pages filter properties.

**Table 7-7: Detect blank pages filter properties**

Property	Description
Mode	A method used for detecting the blank page: <ul style="list-style-type: none"> <li>• <b>Black Area Ratio:</b> detection using <i>BlackAreaRatio</i> property</li> <li>• <b>Preset:</b> detection using Preset property</li> </ul>
Black Area Ratio	Number of black pixels to consider the page as blank. Valid values are a fractional decimal value ranging from 0 to 1.
Preset	Selection of presets indicating whether the image contains noise or data. Valid values: <ul style="list-style-type: none"> <li>• <b>Two Lines OK:</b> Blank pages contain two lines of data.</li> <li>• <b>One Line OK:</b> Blank pages contain a single line of data.</li> <li>• <b>Very Dirty White:</b> Blank pages contain a lot of noise.</li> <li>• <b>Dirty White:</b> Blank pages contain some noise.</li> <li>• <b>Pristine White:</b> Blank pages don't contain noise.</li> </ul>
Left Margin	Left margin to exclude from detection area. Valid values are a fractional decimal value ranging from 0 to 10.
Right Margin	Right border to exclude from detection area. Valid values are a fractional decimal value ranging from 0 to 10.
Top Margin	Top border to exclude from detection area. Valid values are a fractional decimal value ranging from 0 to 10.
Bottom Margin	Bottom border to exclude from detection area. Valid values are a fractional decimal value ranging from 0 to 10.

### 7.6.1.4 Detect color marks

A filter enabling detection of color marks on scanned documents.

**Table 7-8: Detect color marks filter properties**

Property	Description
Mode	Color detection mode letting you choose between color and saturation mode. Valid values: <ul style="list-style-type: none"> <li>• <b>Color</b>: Detect by color.</li> <li>• <b>Saturation</b>: Detect by saturation.</li> </ul>
Color Sensitivity	Filter sensitivity for the selected mark color. Valid values range from 0 to 100.
Mark Color	Color mark to be detected based on selected color mode. Valid values range from 0 to 255.
Mark Saturation	Minimal mark saturation for Saturation mode.
Mark Size	Minimal mark size. Valid values are a fractional decimal value ranging from 0 to 50.
Merging Distance	Maximum distance between color regions. Valid values are a fractional decimal value ranging from 0 to 50.
Use Region	A region on an image to search for the color marks.

### 7.6.1.5 Detect colorfulness

A filter enabling detection of certain color in an image by determining percentage of the image that is “colorful”.

The filter can be used before applying binary or color filters as the **Detect Colorfulness** filter lets you improve the image processing time. For example, if the image is not a 24-bit color image, the image pixel data is not read, and the filter returns 0 as a **Colorfulness** value. Otherwise, **Colorfulness** is the percentage of samples.

**Table 7-9: Detect colorfulness filter properties**

Property	Description
Sample Size	Sample size of colorful spot. Valid values range from 1 to 7.

Property	Description
Colorfulness	Amount of color that a sample must contain to be identified as colorful. Valid values range from 0 to 256.


## 7.6.2 Removal filters

A set of filters that allow you to remove different distractions and redundant elements of the image.

### 7.6.2.1 Remove background

A filter enabling detection of the background color of a scanned document, to drop it out or flatten while capturing all color information in the foreground.

**Table 7-10: Remove background filter properties**

Property	Description
Conversion Mode	<p>Action to take when removing background:</p> <ul style="list-style-type: none"> <li>• <b>Dropout:</b> replaces background with specific color</li> <li>• <b>Smooth:</b> removes scanning and paper non-uniformity</li> <li>• <b>Patterned:</b> removes background (for example, halftone or dither pattern)</li> </ul> <p> <b>Note:</b> If you select the <b>Patterned</b> mode, the color images are processed as binary (black and white) images.</p>
New Background Color	Color to use in selected dropout mode.
Sensitivity	Sensitivity of determining whether a pixel belongs to a background. Affects total amount of background pixels to be detected. Valid values range from -20 to 20.

### 7.6.2.2 Remove black bars

A filter enabling removal of a black area around an image when using overscan option for scanning images. It reduces the physical size of the scanned image and the image file size, removing the black border (produced by scanners). For better results, use the **Deskew** filter and set the black **Fill Color** before applying the filter.

**Table 7-11: Remove black bars filter properties**

Property	Description
Mode	Action to take when removing black bars, letting you reduce the image dimensions or invert the overscan area: <ul style="list-style-type: none"> <li>• <b>Remove (Reduce Size)</b>: reduces image physical size</li> <li>• <b>Clear (Keep Size)</b>: inverts overscan area.</li> </ul>
Max Processing Bottom	Limit of black bar from the bottom. Valid values are a fractional decimal value ranging from 0 to 100.
Max Processing Left	Limit of black bar from the left. Valid values are a fractional decimal value ranging from 0 to 100.
Max Processing Right	Limit of black bar from the right. Valid values are a fractional decimal value ranging from 0 to 100.
Max Processing Top	Limit of black bar from the top. Valid values are a fractional decimal value ranging from 0 to 100.

### 7.6.2.3 Remove holes

A binary filter enabling removal of binder holes around the image edges. The filter uses the following algorithm: the value is multiplied by page height to specify area to search for binder holes. For example, if you set the value to 0.125 when processing 11-inch image, filter will look for binder holes of 1.375 inches (11 inches  $\times$  0.125) in a specific part of the image (top, bottom, left, or right).

**Table 7-12: Remove holes filter properties**

Property	Description
Left	Location to search for binder holes near the left edge of the image
Right	Location to search for binder holes near the right edge of the image

Property	Description
Top	Location to search for binder holes near the image top
Bottom	Location to search for binder holes near image bottom. This value is multiplied by the height of the page to specify the area to search for binder hole objects. For example, when processing an 11-inch high image, a value of 0.125 would attempt to find objects in a horizontal band from the bottom edge of the image to a point 11 inches $\times$ 0.125, or 1.375 inches, from the bottom edge of the image.
Use Region	Specific image region to search for the holes

#### 7.6.2.4 Remove lines

A binary filter enabling removal or reconstruction of lines on a form-based image. Filter allows you to reduce image file size and improve *OCR*.

**Table 7-13: Remove lines filter properties**

Property	Description
Mode	Type of line correction: <ul style="list-style-type: none"> <li>• <b>Reconstruct Form:</b> removes lines, redraws straight lines, and reconnects lines that were previously connected. Commonly used for tables and forms.</li> <li>• <b>Reconstruct Lines:</b> removes lines, repairs overlapped graphics and text, and redraws straight lines in place of removed lines.</li> <li>• <b>Remove lines:</b> removes lines.</li> </ul>
Horizontal Enable	A value specifying if the horizontal lines are detected. If checked, horizontal lines are detected.
Horizontal Straight Line	If checked, straight line processing algorithm is applied for horizontal lines.
Horizontal Min Length	Minimum detectable length of horizontal lines. Valid values range from 0.003 to 100.  Default value is 0.5 inches.

Property	Description
Horizontal Max Gap	Maximum white space between two line-like objects to be considered a single line. Valid values are a fractional decimal value ranging from 0.01 to 100.  Default is 0.01 inches.
Horizontal Curvature	Maximum deviation from a straight line that is allowed for a horizontal line-like object. If detected, such object is considered a line. Valid values range from 0 - 100. The default value is 50.
Vertical Enable	Value specifying whether vertical lines are detected
Vertical Curvature	Maximum deviation from a straight line for a vertical line-like object to be considered a line. Valid values range from 0 - 100. The default value is 50.
Vertical Straight Line	Usage of straight line processing algorithm for vertical lines
Vertical Max Gap	Maximum white space between two vertical lines which can be considered a single line. Valid values are a fractional decimal value ranging from 0.01 to 100.
Vertical Min Length	Minimum detectable length of vertical lines. Valid values are a fractional decimal value ranging from 0.03 to 100.  The default value is 0.5 inches.
Use Region	Region to search for and remove lines.

### 7.6.3 Color adjustment filters

A set of filters that can be used for establishing further sequence of filters (for example, color content detection or color auto detection) and scanner configuration (for example, in scanner control sheets).

### 7.6.3.1 Adjust overall color

A filter enabling image color adjustment.

**Table 7-14: Adjust overall color filter properties**

Property	Description
Algorithm	<p>Property for the automatic adjustment algorithm:</p> <ul style="list-style-type: none"> <li>• <b>Linear</b> uses <i>ContrastStretching</i> method</li> <li>• <b>Nonlinear</b> uses <i>HistogramEqualization</i> equalization method</li> </ul>
Mode	<p>Selection of filter mode:</p> <ul style="list-style-type: none"> <li>• <b>Auto</b>: based on Algorithm, ApplyMode, and LinearMode properties</li> <li>• <b>Manual</b>: manual setting of Lagrange Polynomial properties.</li> </ul> <p>In manual mode, the user modifies the color adjustment polynomial graphically. The transformation is displayed as a line on a graph. On the x-axis there are input gray levels and on the y-axis there are output gray levels.</p> <p>The polynomial can be changed when the user grabs a line with the mouse. This action creates a new bending point. The user can also grab an existing bending point. When bending point is grabbed and the user moves the mouse, the polynomial is changed. When the point is placed in a new location on the graph, the transformation is applied to the preview image. The bending point can be undone by double-clicking the point.</p>
Apply Mode	<p>Selection of the Apply mode for the algorithm:</p> <ul style="list-style-type: none"> <li>• <b>By Luminosity</b>: treat by image luminosity.</li> <li>• <b>By Channel</b>: treat each channel separately.</li> </ul>
Linear Mode	<p>Selection of an algorithm mode:</p> <ul style="list-style-type: none"> <li>• <b>Cutoff Fraction</b>: first occurrence of value bigger than histogram preset Used if the Linear algorithm is selected.</li> <li>• <b>Percentiles</b>: 5% to 95% percentiles used of the histogram.</li> </ul>

Property	Description
Polynomial	Lagrange polynomial intended for advanced color adjustments for separate color channels, namely, R, G, and B.

### 7.6.3.2 Convert specific color

A filter enabling conversion of particular colors into other colors.

**Table 7-15: Convert specific color filter properties**

Property	Description
Old Color	<p>Source color to be converted.</p> <p>Double-click the empty square to display a <b>Color Picker</b> that enables you to select a precise new color.</p> <p>Click the <b>From Image</b> button to select a color from the image preview area, move the mouse over the preview until the square shows the desired color, and then click the left mouse button.</p>
New Color	<p>A new color to convert image to.</p> <p>Double-clicking the empty square shows a <b>Color Picker</b> allowing you to select a precise new color.</p> <p>Clicking the <b>From Image</b> button you can select a color from the image preview area.</p>
Magnitude	Radius of the color sphere to apply color transformation to with value from 1 to 255.

### 7.6.3.3 Convert to black white

A filter enabling conversion of 24-bit color images to a binary image.



**Note:** Filter uses a specific threshold for changing the pixels in a color image: Images that are darker than the specified brightness and contrast, are converted to black; all pixels lighter than the threshold are converted to white.

**Table 7-16: Convert to black white filter properties**

Property	Description
Mode	Selection of a balance between image quality and performance: <ul style="list-style-type: none"> <li>• <b>Binary Fast:</b> fastest mode but less accurate</li> <li>• <b>Binary Balanced:</b> optimal speed and accuracy</li> <li>• <b>Binary Accurate:</b> maximum accuracy with lower speed</li> <li>• <b>Grayscale:</b> produces grayscale output image instead of binary image</li> </ul>
Brightness	Value from 1 to 100 that specifies image brightness.
Contrast	Value from 1 to 100 that specifies image contrast.
Dither	If checked, image is dithered.


#### 7.6.3.4 Convert to black white advanced

A filter enabling detection of the images colorfulness. If the input image is less colorful than a specified value, it is thresholded and output as a binary image.

The filter operates as follows:

1. It considers image background.
2. It counts pixels that are white and nearly white and pixels that are black or nearly black excluding these from count of pixels that are colorful.
3. It counts the colorfulness of the image according to the Detect Type property value.
4. The resulting value is output in PixIpResult object and compared to value specified in Threshold property. If the colorfulness value is less than Threshold, then the image is thresholded, and the filter outputs a thresholded (binary) version of the original image.

**Table 7-17: Convert to black white advanced filter properties**

Property	Description
Mode	Conversion mode: <ul style="list-style-type: none"> <li>• <b>None:</b> no specific mode.</li> <li>• <b>Fast:</b> the fastest mode, but less accurate.</li> <li>• <b>Balanced:</b> balanced speed and accuracy.</li> <li>• <b>Accurate:</b> maximum accuracy with lower speed.</li> </ul>
Brightness	Specific brightness of threshold images. Value can vary between 1 and 100.
Contrast	Specific contrast of threshold images. Value can vary between 1 and 100.
Detect Type	Color detection type by amount or ratio. Valid values: <ul style="list-style-type: none"> <li>• <b>Ratio:</b> Ratio of color and black pixels.</li> <li>• <b>Amount:</b> Amount of color pixel in image.</li> </ul>
Dither if Thresholded	If checked, enables dither for threshold images.
Ignore Paper Color	If checked, detects and removes a colored background before performing automatic color detection.
Threshold	Amount of color a sample must contain to be identified as colorful. Value can vary between 1 and 100. <p> <b>Note:</b> Default value: 5. At the default value, the image cannot be manipulated nor changed. Move the threshold slider to the right to increase the threshold and to output a binary version of the original image.</p>

### 7.6.3.5 Invert black white

A filter enabling inversion of a binary image to its negative equivalent actually modifying the image data: white pixels are changed to black and vice versa.

**Table 7-18: Invert black white filter properties**

Property	Description
Use Region	A specific region of interest on the image.

## 7.6.4 Image quality filters

A set of filters used to enhance the quality of the image.

### 7.6.4.1 Adjust lighting

A filter enabling adjusting brightness and contrast properties for the image.

**Table 7-19: Adjust lighting filter properties**

UI Property Name	Description
Brightness	Brightness adjustment. Valid values: -50 to 50.
Contrast	Contrast adjustment. Valid values: -50 to 50.

### 7.6.4.2 Adjust thickness

A binary filter enabling adjustment of black objects.

**Table 7-20: Adjust thickness filter properties**

Property	Description
Level	<p>Select one of the available filter values: Valid values:</p> <ul style="list-style-type: none"> <li>Value 0: reduces the size of text and image graphics in your batch images to one-pixel thick skeletons of the original objects.</li> <li>Values 1 to 10 to reduce the size of text and image graphics in your images.</li> <li>Values 12 to 21 to increase the size of text and image graphics in your images.</li> </ul> <p>Default value: 11. When this filter is applied with the default value, the image cannot be manipulated nor changed.</p>
Direction	<p>Direction of adjust thickness filter application. Valid values:</p> <ul style="list-style-type: none"> <li>None</li> <li>Diagonal</li> <li>Horizontal</li> <li>Vertical</li> </ul>
Use Region	Specific image region to adjust thickness.

### 7.6.4.3 Remove specks

A binary filter enabling you to set maximum area percentage to consider the object as a speck or noise. A filter requires setting maximum width and height to consider the noise as a speck.


**Table 7-21: Remove specks filter properties**

Property	Description
Max Area (%)	Maximum percentage of the area defined by MaxHeight and MaxWidth that an artifact can occupy to be removed as a noise. Valid values range from 0 - 100. Default value is 100.
Max Height	Maximum speck height. Below this value, a speck is removed. Valid values are a fractional decimal value ranging from 0 to 100.
Max Width	Maximum speck width. Below this value, a speck is removed. Valid values are a fractional decimal value ranging from 0 to 100.
Max Distance	Minimum distance separating a speck from other artifacts. Below this value, a speck will be removed. Valid values are a fractional decimal value ranging from 0 to 100.
Use Region	Specific image region to remove specks.

### 7.6.4.4 Smooth edges

A binary filter enabling removal of bumps and spurs appearing on image text or graphics. The filter algorithm is the following: it looks for any pixel surrounded by five or six pixels of the opposite color and inverts the center pixel based on the filter configuration.

**Table 7-22: Smooth edges filter properties**

Property	Description
Mode	Smoothing mode: <ul style="list-style-type: none"> <li>• <b>None:</b> does not remove any noise.</li> <li>• <b>Remove Black Corners:</b> removes black noise from object corners.</li> <li>• <b>Remove White Corners:</b> remove black noise pixels from the corners of objects.</li> <li>• <b>Trim First:</b> removes black noise pixels before removing white ones. If this flag is not set, then white noise pixels are removed before black ones.</li> </ul> <p> <b>Note:</b> When selecting choosing <b>Remove White Corners</b> option, beware that white noise pixels are removed before the black ones.</p>
Use Region	Select region to apply smoothing to.

## 7.6.5 Page correction filters


A set of filters that allow you to modify and correct the page layout and its properties.

### 7.6.5.1 Crop

A filter enabling setting the image margin and removing white spaces around the edge of the image.

Table 7-23: Crop filter properties

Property	Description
Mode	<p>Mode used to set margin values.</p> <p>Valid values:</p> <ul style="list-style-type: none"> <li> <b>Crop Margins:</b> You can specify the amount of area on each side of the image (<b>Top, Bottom, Left, Right</b>) to be either added or removed. Use negative numbers to cut away to the margins, or positive numbers to add additional white space to the margins.           <p>When <b>Force Symmetry</b> is enabled in the <b>Crop Margins</b> mode, only the <b>Top</b> and <b>Left</b> values are used, and they are applied equally to the opposite margin (for example, top/bottom will be equal, left/right will be equal). In this mode, values for <b>Bottom</b> and <b>Right</b> are not used.</p> </li> <li> <b>Detect Only:</b> You can automatically detect the values showing how much margin might be removed on each edge. Image is not cropped in this mode.           </li> <li> <b>Fixed:</b> You can specify the actual width and height you want the resulting image to be. If the image size needs to be reduced, it will be cut away on those margins as needed. If the image size needs to be expanded, white space will be added until the image size is correct.           <p>The additional settings for <b>Horizontal</b> and <b>Vertical</b> alignment specify how those margins will be cut/expanded. If you specify <b>Center</b>, the margins are cut/expanded equally on opposite sides. If you specify <b>Left/Right</b> or <b>Top/Bottom</b>, the margins are cut/expanded on the side you specify.</p> </li> <li> <b>Auto:</b> In this mode, the margins are automatically cut away if there is white space all along the edge. If there is even 1 pixel in the margin, this will block the crop from cutting further into that margin.           <p>In this mode, the same parameters as for <b>Crop Margins</b> are available. Once the image has been automatically cropped, you can enter the values to apply additionally to the auto cropped result. If</p> </li> </ul>

Property	Description
	<p>you leave these values at 0, the auto cropped result is not changed.</p> <ul style="list-style-type: none"> <li>• <b>Photo:</b> Attempts to detect the edges of the image in the shape of a quadrilateral and then crops the image to that quadrilateral. <b>Photo</b> mode is particularly useful for cropping images captured with mobile devices.</li> </ul> <p> <b>Note:</b> On some occasions, cropping might be too aggressive and the image might be cropped too much. The user might not be able to compensate by rescanning the image.</p>
Left	<p>Desired left margin after cropping. To reduce the margins from the image, the values should be negative.</p> <p>Value from -50 to 50. Default is 0.</p>
Right	<p>Desired right margin after cropping. To reduce the margins from the image, the values should be negative.</p> <p>Value from -50 to 50. Default is 0.</p>
Top	<p>Desired top margin after cropping. To reduce the margins from the image, the values should be negative.</p> <p>Value from -50 to 50. Default is 0.</p>
Bottom	<p>Desired value of bottom margin after cropping. To reduce the margins from the image, the values should be negative.</p> <p>Value from -50 to 50. Default is 0.</p>
Fixed Height	Image height in inches.
Fixed Width	Image width in inches.
Horizontal Alignment	<p>Image alignment on the horizontal axis.</p> <p>Valid Values:</p> <ul style="list-style-type: none"> <li>• <b>Centre</b></li> <li>• <b>Right</b></li> <li>• <b>Left</b></li> </ul>

Property	Description
Vertical Alignment	Alignment of vertical axis used to adjust cropping to: <ul style="list-style-type: none"> <li>• <b>Centre</b></li> <li>• <b>Bottom</b></li> <li>• <b>Top</b></li> </ul>
Force Symmetry	Whether to use a fixed margin for the top, bottom, left, and right margin of the image.

### 7.6.5.2 Deskew

The Deskew filter can be used to straighten binary or color images that show a slant from their correct orientation. Skewing can occur if the original document was skewed when it was fed into the scanner, fax machine, or photocopier. The Deskew filter examines the image and returns the skew angle. The skew angle is measured between the actual edge of the image data and the horizontal or vertical axis. The image data is then rotated to correct the skew angle.

Deskewing an image makes the image contents more legible and can improve OCR results. To achieve the best image quality after image processing and to find the optimal balance between processing speed and accuracy, test this filter with sample images that are similar to the real images you expect to process, and then fine tune the parameters for each filter to find the values most suitable for your document type.

**Table 7-24: Deskew filter properties**

Property	Description
Mode	Selection between different deskew modes: <ul style="list-style-type: none"> <li>• <b>Detect And Deskew</b>: Detect angle and deskew.</li> <li>• <b>Detect Only</b>: Detect angle</li> <li>• <b>Fixed Angle</b>: Rotate by a fixed angle.</li> </ul>
Direction	Direction of the skew angle measurement: <ul style="list-style-type: none"> <li>• <b>Both</b></li> <li>• <b>Horizontal</b></li> <li>• <b>Vertical</b></li> </ul>
Features	Specifying whether image contains lines of text or graphics: <ul style="list-style-type: none"> <li>• <b>Image</b></li> <li>• <b>Text</b></li> </ul>

Property	Description
<b>Fill Color</b>	<p>Color to apply at the image edge to be deskewed.</p> <p>Double-click the empty square to display a <b>Color Picker</b> that enables you to select a precise new color.</p> <p>Click the <b>From Image</b> button to select a color from the image preview area, move the mouse over the preview until the square shows the desired color, and then click the left mouse button.</p>
<b>Fixed Angle</b>	<p>The fixed angle (in degrees) used to rotate all images; used instead of automatic skew detection and correction. This setting requires the <b>Fixed Angle</b> mode selected. Values: 0 to 360.</p> <p>Default: 0.</p>
<b>Minimum Angle</b>	<p>The minimum skew angle (in degrees) that must be detected on the image to apply the deskew filter to straighten that image. This setting works in the <b>Detect And Deskew</b> mode.</p>
<b>Maximum Angle</b>	<p>The maximum skew angle (in degrees) that must be detected on the image to apply the deskew filter. This setting works in the <b>Detect And Deskew</b> mode.</p>
<b>Detection Area</b>	<p>The percent of the page area around the page center that is evaluated to detect a skewed page. Values: 1 to 100. The detection area of less than 100% excludes page margins.</p> <p>Default: 80.</p>
<b>Remove Shear</b>	<p>Selection of whether remove shearing or rotation skew from the page for binary images.</p>

### 7.6.5.3 Rotate

A filter enabling adjustment of binary and color images orientation and mirroring by flipping the page across the vertical axis. The adjustment increment is 90 degrees. The Rotate filter adjusts binary and color images orientation in 90-degree increments. It can rotate the image 90-degrees clockwise and counterclockwise. It can also rotate it 180-degrees as well as perform mirroring, which flips the page across the vertical axis so that it appears to be the mirror image of the original. It can also constrain orientation to specified one (landscape or portrait).

We recommend that you use OCR modules for auto-rotation as this filter has the following limitations:

- An image must contain at least 50 printed characters, otherwise, the algorithm returns the control.
- Filter accuracy is dependent on the quality of the characters printed on the image.
- Most of the characters must be Latin alphabetic characters and not numeric characters.
- If a rotated image contains a table or a picture in it, the image may not be auto-rotated.

**Table 7-25: Rotate filter properties**

Property	Description
Mode	Image rotation mode: <ul style="list-style-type: none"> <li>• <b>Auto</b>: Automatically detects the correct orientation of the text or images and rotates the image as close as it can to an upright position by rotating the image in 90-degree increments. That is, <b>Auto</b> might not rotate the image to a perfectly upright position; for example, an image that is rotated 98 degrees clockwise would be repositioned at 8 degrees clockwise.</li> <li>• <b>Landscape Only</b>: constrain landscape orientation</li> <li>• <b>Portrait Only</b>: constrain portrait orientation</li> <li>• <b>Clockwise</b>: rotate 90</li> <li>• <b>Counter Clockwise</b>: rotate counterclockwise</li> <li>• <b>Rotate 180</b>: rotate 180 degrees</li> <li>• <b>None</b>: no rotation is performed</li> </ul>
Mirror	Mirror image by flipping across the vertical axis.

### 7.6.5.4 Scale

A filter enabling scaling an image and changing its dimensions (DPI). You can use the filter for the following purposes:

- Leveraging image resolution
- Modifying image dimensions
- Preparing for further archiving

**Table 7-26: Scale filter properties**

Property	Description
Mode	Scaling mode selection: <ul style="list-style-type: none"> <li>• <b>Scale to Size:</b> Scales image to specified size with the same resolution.</li> <li>• <b>Scale to Resolution:</b> Scales image to specified resolution with the same size.</li> <li>• <b>Scale with Coefficient:</b> Scales image with specified coefficient. Resolution remains the same.</li> <li>• <b>Resolution Alignment:</b> Changes resolution in image properties so that horizontal and vertical resolutions are equal. In this mode, maximum one is selected for a new resolution. Bitmap itself remains as is.</li> <li>• <b>Change Resolution:</b> Changes resolution in the image properties; bitmap itself remains as is resulting in updated resolution and size.</li> </ul>
Page Width	New width available with <b>Scale To Size</b> mode selected.
Page Height	New height available with <b>Scale To Size</b> mode selected.
Scale Factor	Specifying scale factor for Scale with <b>Coefficient</b> mode. Valid values range from 1 to 10
Round Width To	Defines multiple for Round Width.
Page Resolution (DPI)	New resolution available with <b>Scale To Resolution</b> mode selected. Valid values range from 1 to 1000. Default is 96.
Smooth Edges	Defines whether to perform scaling with smoothing.

## 7.6.6 Third-party filters

A set of filters used to optimize OCR for images that were captured from mobile devices.

### Notes

For more information, see the following:

- Although these filters can be used on images from other devices, such as scanners or fax machines, they are most useful for those images that were captured from mobile devices.
- When two or more filters perform similar functions (for example, **Convert to Black and White (Adaptive)** and **Convert to Black-White**), you might want to perform testing to determine the best filter to use.

### 7.6.6.1 Convert to black and white (adaptive)

This filter converts color and grayscale images to a binary image using an adaptive algorithm. This filter attempts to compensate for the usual variation in shading in images captured from mobile devices versus other devices (such as scanners and fax machines).

### Tips

For more information, see the following:

- This filter might not produce acceptable results for the following:
  - White text on a black background (in general, the black is removed without reversing the white)
  - Images with intentional variations in shading, such as photographs or color diagrams.
- Running this filter on images that contain strictly black and white text might not result in a significant improvement.

### 7.6.6.2 Line removal

This filter deletes the following types of lines from an image:

- Solid lines, such as ones that make up boxes or tables
- Dashed or dotted lines

Use this filter to delete extraneous lines that can degrade the quality of OCR. In [Figure 7-1](#), the short, vertical line to the right of the digits might be interpreted as an 1 or a l if not removed from the image.


Invoice 1003|

**Figure 7-1: Line removal example**


Use the options in this filter only if you expect your images to contain those problematic lines. Do not use this filter if valid text contains lines, dashes, or dots. In [Figure 7-2](#), the series of hyphens are valid, but might be deleted if you select **Remove Dashed Lines**.

40 - - - 45

**Figure 7-2: Dash removal example**

 **Note:** You might not want to use this filter when using free-form extraction because some boxes and lines can be used to recognize the location of specific kinds of text; for example, if a date is not vertically aligned within an acceptable tolerance to the Date printed text, then a box surrounding the two pieces of text can help to identify the text as a valid date.

**Table 7-27: Line removal filter properties**

<b>Remove Dashed Lines</b>	Deletes dashed or dotted lines, regardless of orientation.   <b>Note:</b> Thick dashed lines might not be removed.
<b>Remove Horizontal Lines</b>	Deletes horizontal lines.
<b>Remove Vertical Lines</b>	Deletes vertical lines.
<b>Aggressiveness</b>	See the following options: <ul style="list-style-type: none"> <li>• <b>Low:</b> Favors preserving image data over deleting lines.</li> <li>• <b>Medium:</b> Attempts to balance preserving image data against deleting lines.</li> <li>• <b>High:</b> Favors deleting lines over preserving image data.</li> </ul>

### 7.6.6.3 Set DPI

If an image has a DPI metadata value that is not set or set below the specified DPI threshold, then this filter resets that value.

For images that are captured using a mobile device, the DPI metadata value is usually not set or set incorrectly; therefore, you must reset the DPI metadata value to optimize OCR.

You can achieve the best results when you know the DPI of the incoming images; then you can simply set **New DPI** to the known DPI. However, if you do not know the image DPI, then you can achieve the next best results from specifying a known paper width; for example, if you know that all the images are 8.5 x 11-inch paper (in a portrait orientation), then set **Specify Paper Width** to 8.5. Depending on how accurately the edges of the paper fit to the borders of the image, you might need to compensate by specifying a larger or smaller value than the actual paper width.



Lastly, using **Auto-Detect** might give acceptable results; for example, if all of the images are different paper sizes (payroll stubs, tax forms, and so forth), then you might not use **Specify Paper Width**.



**Tip:** In general, it is recommended that 200 DPI be the minimum setting for the DPI threshold. Although you should always consider your particular business needs, in many situations, 200 DPI provides a good balance between image size and quality for OCR.

**Table 7-28: Set DPI filter properties**

<b>DPI Threshold</b>	If the image's DPI metadata value is less than the value specified in this field, then the image's DPI metadata value is changed to the <b>New DPI</b> value or an internally calculated value (when <b>Auto-Detect</b> is selected); otherwise the image's current DPI metadata is used.
<b>Mode</b>	See the following options: <ul style="list-style-type: none"> <li>• <b>Auto-Detect:</b> The DPI to use is calculated using an internal algorithm.</li> <li>• <b>Specify Paper Width:</b> Use the DPI calculated by using the value specified in <b>Paper Width</b>.</li> <li>• <b>Fixed:</b> You must specify a fixed DPI value in <b>New DPI</b>.</li> </ul>

<p><b>Paper Width</b></p>	<p>Specify the width of the paper. This width is used to calculate the DPI to use. Before applying the filter with this mode, crop and rotate the image; otherwise, this mode does not work correctly.</p> <p> <b>Note:</b> This field is enabled only if <b>Mode</b> is set to <b>Specify Paper Width</b>.</p>
<p><b>New DPI</b></p>	<p>Specify an integer to use for the DPI.</p> <p> <b>Note:</b> This field is enabled only if <b>Mode</b> is set to <b>Fixed</b>.</p>

#### 7.6.6.4 Check Preprocessor - U.S.

This filter adjusts a mobile picture of a standard U.S. personal or business check's DPI and dimensions (including 90-degree corrective rotations) to closely match a scanned image of the check.

The input to this filter must be a binary TIFF image in G4 compression. If an image is not in this format or compression, you can use another filter to convert it.

The output from this filter is a binary TIFF image in G4 compression.

After this filter has been applied, the image can be processed by the Check Reading engine (with the Advanced Recognition license). You should not submit a mobile picture of a check to the Check Reading engine without applying this filter because the Check Reading engine is unlikely to process it correctly.

## Chapter 8

# Designing image conversion

## 8.1 Image Conversion profiles

The topics within this section introduce the **Image Conversion** functionality, describe main features and functions of the Image Conversion profiles editor, and cover information on how to create, manage, and configure Image Conversion profiles.

### 8.1.1 What is image conversion?

Image conversion is a mechanism that lets you convert files from one format to another and transform files from one type to another. It can be used for:

- Changing image properties including file format, color format, and compression.
- Converting non-image files to images.
- Converting electronic textual documents to *PDF* and *PDF/A* formats.
- Converting Microsoft Office, Text, and *HTML* documents to image or PDF formats without having to install Microsoft Office.
- Specifying additional options when generating output files of such file types as PDF, *TIFF*, or *BMP*. For example, include bookmarks, keywords, and dictionary in the PDF output, or define custom tags in the TIFF output.
- Merging single-page files to multi-page documents and splitting multi-page documents into single pages.
- Merging annotations, added to TIFF images by other modules, such as Image Processor or Completion, into the output image.
- Merging and splitting text-based PDF and PDF/A documents.
- Auto-detecting the type of content of the processed PDF page based upon the following types: text, image, mixed content, or unknown type (e.g. blank page).

For the full list of supported image and non-image file formats that the Image Converter module can handle, see *OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)*.

### Related Topics

[“How does image conversion work?” on page 230](#)

## 8.1.2 How does image conversion work?

Image conversion functionality is handled by the Image Converter module.

Users can create various Image Conversion profiles for every specific business scenario. Image Conversion profiles define how input files must be converted into the specified output format. For example, Image Conversion profile can define how to convert several single-page *TIFF* files into a multi-page *PDF* document.

Once Image Conversion profiles have been created, they must be deployed to the capture server. Profiles deployed to the server become available when setting up an Image Converter step in a capture process. For the detailed information on the concept of capture process model, see [“Designing a CaptureFlow” on page 75](#).

The Image Converter step is set up by selecting one of the Image Conversion profiles. The list of available for setup profiles will include only those profiles that are compatible with the Image Converter step trigger level. When setting up an Image Converter step, the **Input Source** value of the profile selected must correspond to a batch creating module in a CaptureFlow. When a batch based on this CaptureFlow is created, the output files are generated according to the profile selected during setup. For more information on setting up an Image Converter step, see *OpenText Intelligent Capture - Image Handling Modules Guide (ECPCORE-CIH)*.

### Related Topics

[“Getting started” on page 75](#)

## 8.1.3 High-level steps

Perform the following high-level steps to configure and set up your Image Conversion profiles.

**Table 8-1: Image Conversion profiles: high-level steps**

Step	Description	See
1	The entry point to get started with Image Conversion profiles is the <b>Image Conversion Profiles</b> list where you can create and manage profiles. To access the list, select the <b>Image Conversion</b> area.	<a href="#">“Managing Image Conversion profiles” on page 246</a>
2	Add a new Image Conversion profile.	<a href="#">“Adding a new profile” on page 248</a>

Step	Description	See
3	Configure the Image Conversion profile as follows: <ol style="list-style-type: none"> <li>1. Specify input files properties: select <b>Input Source</b> according to the level where the Image Converter module takes files to process: document or page level. For non-image input files, define <b>Options for Non-image Documents</b>.</li> <li>2. Specify output files properties. Select <b>Output Destination</b> as a level for the Image Converter module to save output file. Define <b>Output Format</b>. You can define additional options for <i>PDF</i> and <i>TIFF</i> formats if necessary.</li> <li>3. Specify <b>Miscellaneous Options</b> if necessary.</li> </ol>	<a href="#">"Image Conversion profiles editor" on page 232</a>
4	Test the profile.	<a href="#">"Testing profiles" on page 248</a>
5	Save the profile.	<a href="#">"Image Conversion profiles list" on page 246</a>
6	Upload the newly created profile to the capture server.	For more information, see <a href="#">"Deploying a component" on page 433</a> .
7	Open CaptureFlow Designer and create a CaptureFlow including the Image Converter step in it. Set up the step by selecting the Image Conversion profile you have created and uploaded to the server. Note that the list of profiles is automatically filtered out so that it contains only the profiles compatible with the Image Converter step trigger level.	See <a href="#">"Managing CaptureFlows" on page 91</a> and <i>OpenText Intelligent Capture - Image Handling Modules Guide (ECPCORE-CIH)</i> .

Additional steps may be required based on the business problem you are attempting to solve.

## Related Topics

[“Deploying a component” on page 433](#)

[“Managing CaptureFlows” on page 91](#)

## 8.2 Image Conversion profiles editor

This section describes the Image Conversion profiles editor properties.

**Image Conversion Profiles** editor property sheet contains the following tabs:

- **Profile Property:** includes settings that can be configured for an Image Conversion profile.






**Note:** Some of the properties are grouped into categories. For example, the **Options for Non-image Documents** category includes a set of properties for non-image files, and the **Output Format** category includes a set of properties to specify the output files format. The **Options for Non-image Documents** options are disabled if the **Input Source** value is **Single-page Image**.



- **Value:** an actual value of a setting.
- **Tips:** helpful text to describe a setting.



[“Image Conversion profile properties” on page 233](#) provides a description of Image Conversion profile properties.




**Table 8-2: Image Conversion profile properties**


Category	Property	Description
Input Source	Input Source	<p>Specifies the type of input files. The value of this property must correspond to the level at which the Image Converter module take files to process. The options are:</p> <ul style="list-style-type: none"> <li>• <b>Multi-page Document (Document Level):</b> select if your input files are image or non-image multi-page documents at level 1. Use the <code>InputFile IA</code> value at level 1.</li> <li>• <b>Multi-page Document (Page Level):</b> select if your input files are image or non-image files multi-page documents at level 0. Use the <code>InputFile IA</code> value at level 0.</li> <li>• <b>Single-page Image:</b> select if your input files are single-page image files at level 0.</li> </ul> <p> <b>Note:</b> Single-page non-image files are not supported.</p> <p>For the full list of supported image and non-image file formats that the Image Converter module can handle, see <i>OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)</i>.</p>


Category	Property	Description
<p><b>Options for Non-Image Documents</b></p> <p> <b>Note:</b> The setting group is disabled if <b>Single-page Image</b> is selected for <b>Input Source</b>.</p>	<p><b>Resolution (DPI)</b></p>	<p>Specify the target resolution for processing non-image input documents.</p> <p>Because of their format specification restrictions, the specified target resolution is not applied to the following output formats:</p> <ul style="list-style-type: none"> <li>• JBIG (*.JBG)</li> <li>• Compuserve (*.GIF)</li> <li>• CALS (*.CAL)</li> <li>• FAX (*.DCX)</li> </ul> <p> <b>Note:</b> Due to third-party configuration issues, Image Converter may process non-image documents with incorrect resolution on a Windows 7 x64 operating system. For example, the resolution of the resulting image may be 300 <i>DPI</i> instead of 100 DPI as defined in a profile. The following workaround helps to solve the issue: in <b>Control Panel</b>, disable the <b>User Account Control</b>. For additional information, see the Microsoft documentation.</p>



Category	Property	Description
	Page Size	<p>Specify the target page size for processing non-image input documents.</p> <p> <b>Note:</b> Not applicable when input documents are <i>PDF</i> files.</p> <p>If input files are <i>HTML</i> documents and the <b>Use Document Default</b> value is selected, the printer default value is used.</p> <p> <b>Note:</b> Be aware of the following combinations of <b>Page Size</b> and <b>Orientation</b> values that, due to the third-party applications issues, may cause incorrect processing results:</p> <ul style="list-style-type: none"> <li>• <b>Processing Microsoft Word documents:</b> If <b>Page Size</b> is set to <b>Ledger 17x11 in</b> and <b>Orientation</b> is set to <b>Use Document Default</b>, then the resulting image has <b>Portrait Orientation</b> instead of <b>Landscape</b>.</li> <li>• <b>Processing PowerPoint documents:</b> If <b>Page Size</b> is set to <b>Ledger 17x11 in</b> and <b>Orientation</b> is set to <b>Landscape</b>, then the resulting image has <b>Portrait Orientation</b> instead of <b>Landscape</b>. Also, some text from the original <i>PPT</i> file can be lost.</li> <li>• <b>Processing HTML documents:</b></li> </ul>


Category	Property	Description
		<p>If <b>Page Size</b> is set to <b>Ledger 17x11 in</b> and <b>Orientation</b> is set to <b>Use Document Default</b>, then the resulting image has <b>Portrait Orientation</b> instead of <b>Landscape</b>.</p>
	<p><b>Orientation</b></p>	<p>Specify the target orientation for processing non-image input documents.</p> <p> <b>Note:</b> Not applicable when input documents are PDF files.</p> <p>When <b>Use Document Default</b> is specified, page orientation is determined by the properties of the input non-image document. For example, if a source Microsoft Word document is imported with a landscape orientation, the resulting image file assumes a landscape format.</p> <p> <b>Note:</b> Be aware that certain combinations of the <b>Page Size</b> and <b>Orientation</b> values may cause incorrect processing results. See the note for the <b>Page Size</b> property.</p>



Category	Property	Description
	<b>HTML Rendering Engine</b>	<p>Specify the rendering engine to process non-image HTML files. The options are:</p> <ul style="list-style-type: none"> <li>• <b>Embedded Image Converter Engine</b></li> <li>• <b>Internet Explorer</b></li> </ul> <p> <b>Note:</b> Requires the Virtual Printer feature.</p> <ul style="list-style-type: none"> <li>• <b>Microsoft Word</b></li> </ul> <p> <b>Note:</b> Requires the Virtual Printer feature.</p> <p><b>Embedded Image Converter Engine</b> does not require installing any third-party applications for processing documents. When <b>Internet Explorer</b> or <b>Microsoft Word</b> is selected, ensure the application is installed on the machine where the Image Converter module is installed. For more information, see <i>OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)</i>.</p> <p> <b>Note:</b> If the <b>Internet Explorer</b> value is selected and you specified your custom location for the <b>Temporary Folder</b>, ensure this temporary folder meets the requirements to be available for any user in production mode. For more information on requirements for the temporary folder, see <i>OpenText Intelligent Capture - Image Handling Modules Guide (ECPCORE-CIH)</i>.</p>

Category	Property	Description
	<b>Office Documents Rendering Engine</b>	<p>Specify the rendering engine to process Microsoft Office and text files. The options are:</p> <ul style="list-style-type: none"> <li>• <b>Embedded Image Converter Engine</b></li> <li>• <b>Microsoft Office Engine</b></li> </ul> <p> <b>Note:</b> Requires the Virtual Printer feature.</p> <p><b>Embedded Image Converter Engine</b> does not require installing any third-party applications for processing documents. When <b>Microsoft Office Engine</b> is selected, ensure that you have Microsoft Office installed on the machine where the Image Converter module is installed. For more information, <i>OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)</i>.</p>
<b>Output Destination</b>	<b>Output Destination</b>	<p>Specify the type of output files. The value of this property defines the level for the Image Converter module to save the processed files. The options are:</p> <ul style="list-style-type: none"> <li>• <b>Single-page File:</b> select if you want the Image Converter module to save output as single-page images at page level (level 0).</li> <li>• <b>Multi-page Document (Task Level):</b> select if you want the Image Converter module to combine input into multi-page documents (one per task node) and save the output at trigger level.</li> </ul>

Category	Property	Description
Output Format	File Type	<p>Select the format for output file. The supported file formats are specified in <i>OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)</i>.</p> <p> <b>Note:</b> If input file is a single-page <i>TIFF</i> image and output file has the <b>Multi Format TIFF</b> value, the module performs merging, as it is defined in the profile, and does not convert the input files to the specified <b>Color Format</b> and <b>Compression</b>.</p> <p>For the full list of supported image and non-image file formats that the Image Converter module can handle, see <i>OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)</i>.</p>
	Merge Annotations	<p>Select to include annotations created for TIFF images by the Image Processor or Completion modules into the output images. Once annotations are merged to the image, they become a part of the image itself and cannot be modified.</p>
	Color Format	<p>Select the color format for output files. The list of available values depends on the <b>File Type</b> selected. The supported combinations of file format, color format, and compression are specified in <i>OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)</i>.</p>

Category	Property	Description
	<p><b>Compression</b></p>	<p>Select the compression of output files. The list of available values depends on the <b>File Type</b> and <b>Color Format</b> selected. The supported combinations of file format, color format, and compression are specified in <i>OpenText Intelligent Capture - Operating Specifications (EPCORE-RLI)</i>.</p>
	<p><b>PDF Options</b></p> <p> <b>Note:</b> These options are enabled if <b>PDF/A (*.PDF)</b> or <b>PDF/Web (*.PDF)</b> file type is selected.</p>	<p><b>Compatibility Level:</b> specify the compatibility level of the output PDF file.</p> <p><b>Keep Input Textual Data:</b> select to retain textual data as text during splitting, merging, or converting an electronic document to PDF or <i>PDF/A</i>.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• When selected, the <b>Color Format</b> and <b>Compression</b> settings are ignored for input text files.</li> <li>• This option is not supported for input HTML files when Internet Explorer is selected in <b>HTML Rendering Engine</b>. In this case, image PDF or PDF/A output files will be generated.</li> </ul>
		<p><b>Keep Input PDF Bookmarks:</b> select to include bookmarks from the input PDF file into the output PDF file. Ignore the option if the input file is not the PDF format.</p>



Category	Property	Description
		<p><b>Keep Input PDF Catalog and Dictionary:</b> select to retain the Adobe® Catalog® and Info Dictionary values from the input PDF file in the output PDF file. Ignore the option if the input file is not the PDF format.</p> <p><b>Document Information Source:</b> select the source of the document information for the output PDF:</p> <ul style="list-style-type: none"> <li>• Select <b>First Document</b> to get information from the first input document.</li> <li>• Select <b>Last Document</b> to get information from the last input document.</li> <li>• Select <b>IA Values</b> to use information specified in IA Values.</li> </ul> <p> <b>Note:</b> If the option <b>First Document</b> or <b>Last Document</b> is selected for the input file that is not PDF format, the output PDF file will have empty document information.</p> <p><b>Keyword Separator:</b> specify keyword concatenation mode. The option is disabled if the <b>Document Information Source</b> value is <b>IA Values</b>.</p> <p>Leave blank to use only the first keyword from the <b>First Document</b> or <b>Last Document</b> (as specified in <b>Document Information Source</b>).</p> <p>Add the text to delimit all the keywords by this text. In this case, the keywords from all pages will be used regardless of the <b>Document Information Source</b> selected.</p>

Category	Property	Description
		<p><b>PDF Compression:</b> For best results we recommend using 24-bit color.</p> <p> <b>Note:</b> If you use smaller color formats, the resulting compressed file might be larger than the resulting uncompressed file. The color format affects only when you convert an image file to a PDF.</p> <hr/> <p><b>TIFF Options</b></p> <p> <b>Note:</b> These options are enabled if <b>TIFF (*.TIF)</b> or <b>Multi Format TIFF (*.TIF)</b> file type is selected.</p> <p><b>Number of Custom Tags:</b> specify the number of custom TIFF tags to be added to the output TIFF file. For every [N] custom tag added, a set of options is available:</p> <ul style="list-style-type: none"> <li>• <b>Tag [N] Code:</b> specify the TIFF tag code.</li> <li>• <b>Tag [N] Type:</b> specify the TIFF tag type.</li> <li>• <b>Tag [N] Value:</b> specify the TIFF tag value.</li> </ul> <p>Leave blank to use the <b>TiffTag_N_Value IA Value</b> from the task node. To create the required number of <b>TiffTag_N_Value</b> values, use <b>Custom Values</b> in CaptureFlow Designer.</p>

Category	Property	Description
Miscellaneous Options	Conversion Timeout (seconds)	<p>Specify the maximum amount of time in seconds that the module waits for a response from a rendering application or a virtual printer while converting an imported non-image file. The time-out value is reset each time after a response is returned back to Image Converter. In other words, time-out value is started over for every task sent to rendering application or virtual printer.</p> <p>If processing exceeds the specified time-out time, the error is shown, and further task processing is performed according to the error handling settings specified during a CaptureFlow creation. For more information on error handling setup, see <a href="#">“Adding batch processing steps”</a> on page 104.</p> <p>The default value is 600 seconds and can be changed from 60 to 9999 seconds (inclusive).</p>

Category	Property	Description
	<p><b>Process PDF's with Restrictions</b></p>	<p>The <b>Process PDFs with Restrictions</b> profile property permits a secondary engine to process PDFs. When this property is enabled, the system will retry the operation using the secondary engine if the primary engine cannot process a PDF. This helps process PDFs with document securities, which cannot be processed using the primary engine. Enabling the secondary engine adds processing time but results in a higher document success rate. When the <b>Process PDFs with Restrictions</b> profile property is disabled, only the primary engine is used for PDF processing.</p>

Category	Property	Description
	<b>Extract attachments from PDF file</b>	<p>Enable or disable the extraction of attachments from PDF files. The attachments are saved to IA values, that is, input file nodes. The associated IA values are the following:</p> <p>AttachmentsExtractedCount</p> <p>AttachmentsDocumentCount</p> <p>AttachmentFile&lt;[n]&gt;</p> <p>AttachmentFileName&lt;[n]&gt;</p> <p>AttachmentFileType&lt;[n]&gt;</p> <p>&lt;[n]&gt; is an integer that identifies the file according to the order in which it was extracted. Attachments are extracted in the following order:</p> <ol style="list-style-type: none"> <li>Files attached as annotations in the order that they were added to each page.</li> <li>File attachments in ascending alphanumeric order based on their file names.</li> </ol> <p>The number of IA Values specified in the <code>imgconv.mdf</code> file must be equal to or greater than the number of attachments that are expected to be extracted. By default, 10 IA Values are defined in the <code>imgconv.mdf</code> file.</p> <p>For more information about these IA values, see <i>OpenText Intelligent Capture - Image Handling Modules Guide (ECPCORE-CIH)</i>.</p> <p>For more information about processing attachments in</p>

Category	Property	Description
		<p>PDFs, see <a href="#">“Processing PDF attachments”</a> on page 256.</p> <p>The maximum number of attachments that can be extracted are set by <b>Maximum extracted attachments</b>.</p> <p> <b>Note: Output Destination</b> must be set to <b>Single-page file</b>.</p>
	<b>Maximum extracted attachments</b>	<p>Specify the maximum number of attachments to be extracted from the associated PDF. If the PDF contains more attachments than specified, then they are skipped.</p> <p> <b>Note: Output Destination</b> must be set to <b>Single-page file</b> and <b>Extract attachments from PDF file</b> must be enabled.</p>

### Related Topics

[“Image Conversion profiles usage examples”](#) on page 250

## 8.3 Managing Image Conversion profiles

This section introduces the Image Conversion profiles list, describes how to use it and provides the procedure of creating an Image Conversion profile.


### 8.3.1 Image Conversion profiles list

The **Image Conversion Profiles** list displays all the Image Conversion profiles existing in the currently selected capture system. As with other types of profiles, these profiles are global and can be used in any process of this capture system.

You can view the **Image Conversion Profiles** list by selecting the **Image Conversion** area.

The following table describes the layout and options available in the list window.

**Table 8-3: Image Conversion profiles window**

Element	Description
Section bar	<p>Displays buttons for each open profile as well as the <b>Image Conversion Profiles</b> list. Click a section button to navigate to a different profile or to return to the <b>Image Conversion Profiles</b> list. Close individual profiles by clicking the <b>X</b> on the section bar for the profile.</p> <p>If the number of open profiles is such that some buttons are hidden from view, paging controls and a drop-down button appear on the right side of the section bar. Click &lt;&lt; or &gt;&gt; to scroll left or right, respectively. Or, click the drop-down button to select from a list of open profiles or the <b>Image Conversion Profiles</b> list.</p>
<b>Image Conversion Profile</b>	The column displays names of all Image Conversion profiles existing in the capture system. To open a profile, double click it in the list.
<b>Description</b>	For the selected Image Conversion profile, the column contains its description or usage instructions. To edit a description of the profile, click the existing description.
<b>Last Saved</b>	For the selected Image Conversion profile, the column contains the date when this profile was last saved on the local machine.
<b>Delete</b>	<p>Use the button to delete the selected Image Conversion profile from the list.</p> <p> <b>Note:</b> If you delete a profile that is set up for an Image Converter step in a process, the system will not be able to run this step and process batches; the corresponding error is shown.</p>
<b>New</b>	Use the button to create an Image Conversion profile. For more information on Image Conversion profiles naming restrictions, see <a href="#">“Adding a new profile” on page 248</a> section.
<b>Open</b>	Use the button to open the selected Image Conversion profile for viewing or editing its settings.

The number of selected profiles is displayed in the lower-left part of the profiles list while the number of available profiles – in the lower-right corner. Use the arrow keys to navigate between the profiles in the **Image Conversion Profiles** list.

## Related Topics

[“Image Conversion profiles editor” on page 232](#)

[“Defining capture systems” on page 29](#)

### 8.3.2 Adding a new profile

#### To add a profile:

1. Click **Image Conversion**.
2. In the **Image Conversion Profiles** list window, click **New**. The **Image Conversion Profile - New** window opens.
3. In the window, specify a profile name following the [naming convention](#).
4. Save the profile. The new profile is added to the **Image Conversion Profiles** list. The profile tab appears on the section bar and the profile window opens for editing.



**Note:** Image Conversion profiles are stored on the Designer host machine. Profiles are saved as configuration files in the following folder:

```
C:\Users\\Documents\\<Capture System name>\GlobalData\ImageConversion
```

## Related Topics

[“Image Conversion profiles list” on page 246](#)

### 8.3.3 Testing profiles

#### To test a profile:

1. In the profile window, click **Test Conversion**.
2. In the window that opens, select the files to convert by clicking **Add Files**, and then click **Convert**.



#### Notes

For more information, see the following:

- By default, the converted output files are saved to the current application directory. To change the directory, specify its path in **Save Result To**.
- For password-protected Microsoft Office files, you are prompted to enter the password twice.

The **Status** column displays whether conversion of a file succeeded or failed.

The **Error** column displays an error description.

The converted output files are named as outlined in the following table.

One single-page file	<p><code>&lt;source_file_name&gt;.&lt;ext&gt;</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>&lt;source_file_name&gt;</code> is the name of the source file.</li> <li>• <code>&lt;ext&gt;</code> is the extension specified by the <b>Output Format File Type</b> option.</li> </ul>
One multi-page file	<p><code>&lt;last_source_file_name&gt;_merged_docs&lt;n&gt;.&lt;ext&gt;</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>&lt;last_source_file_name&gt;</code> is the name of the last file merged into the single output file.</li> <li>• <code>&lt;n&gt;</code> is the number of input documents that were merged.</li> <li>• <code>&lt;ext&gt;</code> is the extension specified by the <b>Output Format File Type</b> option.</li> </ul>
Multiple single-page files	<p><code>&lt;source_file_name&gt;_page &lt;n&gt;.&lt;ext&gt;</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>&lt;source_file_name&gt;</code> is the name of the source file.</li> <li>• <code>&lt;n&gt;</code> is the page's number in the multi-page document (from which the single-page files were split).</li> <li>• <code>&lt;ext&gt;</code> is the extension specified by the <b>Output Format File Type</b> option.</li> </ul>
Attachments extracted from PDF files	<p><code>&lt;source_file_name&gt; - AttachmentFile&lt;n&gt; - &lt;attachment_file_name&gt;.&lt;ext&gt;&gt;</code></p> <p>where:</p> <ul style="list-style-type: none"> <li>• <code>&lt;source_file_name&gt;</code> is the name of the source file.</li> <li>• <code>&lt;n&gt;</code> is the number of the attachment.</li> <li>• <code>&lt;attachment_file_name&gt;</code> is the file name of the attachment.</li> <li>• <code>&lt;ext&gt;</code> is the extension specified by the <b>Output Format File Type</b> option.</li> </ul>

## 8.4 Image Conversion profiles usage examples

The section provides some practical examples that can help you understand how to configure Image Conversion profiles for different business scenarios:

### 8.4.1 Converting Microsoft Office multi-page documents into images

#### Prerequisites:

Microsoft Office documents are imported into a capture system using Standard Import. The customer has Microsoft Office installed on the machine.

#### Goal:

The customer needs to convert multi-page Microsoft Word documents into image files. The preferred format for output files is **JPEG**. Customer wants to keep the processed files compressed.

#### Solution:

After Standard Import, the Image Converter module will take input files for processing at document level (level 1) which corresponds to **Multi-page Document (Document Level)** selected as **Input Source**. The **Output Destination** for images is **Single-page File**. The profile with such input and output values is compatible with Image Converter triggered at level 1.

The following table contains the settings that must be configured for the Image Conversion profile.

**Table 8-4: Image Conversion profile settings**

Property	Value
Input Source	Multi-page Document (Document Level)
Resolution ( <i>DPI</i> )	300
Page Size	Use Document Default
Orientation	Use Document Default
HTML Rendering Engine	Internet Explorer
Office Documents Rendering Engine	Microsoft Office Engine
Output Destination	Single-page File
File Type	JPEG (*.JPG)
Merge Annotations	On (Select the check box)
Color Format	8 bit Gray
Compression	Sequential JPEG

Property	Value
Number of Custom tags	0
Conversion Timeout (seconds)	600 (Leave as default)

**Summary:**

In this scenario, virtual printing technology is used for processing non-image documents. The value of **HTML Rendering Engine** does not make any difference in this example as it only impacts converting in case when the format of input files is *HTML*.

## 8.4.2 Scanned images merged into a multi-page PDF document

**Prerequisites:**

Images are imported into a capture system by scanning.

**Goal:**

The customer needs to create an archive by merging separate image files into a single *PDF* document. The output PDF file must be compatible with all the Adobe® Acrobat® versions older than 6.0.

**Solution:**

After ScanPlus, the Image Converter module will take input files for processing at page level (level 0) which corresponds to **Single-page Image** selected as **Input Source**. The **Output Destination** for a multi-page PDF document is **Multi-page Document (Task Level)**. The profile with such input and output values is compatible with Image Converter triggered at level 1 or higher.

The following table contains the settings that must be configured for the Image Conversion profile.

**Table 8-5: Image Conversion profile settings**

Property	Value
Input Source	<b>Single-page Image</b>
Output Destination	<b>Multi-page Document (Task Level)</b>
File Type	<b>PDF/A (*.PDF)</b>
Color Format	<b>Binary</b>
Compression	<b>CCITT Group 4</b>
Compatibility Level	<b>Acrobat 6.0</b>
Keep Input Textual Data	Off (Clear the check box)

Property	Value
Keep Input PDF Bookmarks	Off (Leave as default)
Keep Input PDF Catalog and Dictionary	Off (Leave as default)
Document Information Source	<b>First Document</b>
Keyword Separator	Leave blank to use only the first keyword from the First Document
Conversion Timeout (seconds)	<b>600</b> (Leave as default)

### 8.4.3 Converting Microsoft Office multi-page documents into single-page PDF documents

#### Prerequisites:

Microsoft Office documents are imported into a capture system using Standard Import. The customer has Microsoft Office installed on the machine.

#### Goal:

The customer needs multi-page Microsoft Word documents to be split into single-page *PDF* files. The textual data from the input documents must be retained in the output PDF file. The compatibility with the oldest available Adobe® Acrobat® version is required.

#### Solution:

After Standard Import, the Image Converter module will take input files for processing at document level (level 1) which corresponds to **Multi-page Document (Document Level)** selected as **Input Source**. The **Output Destination** for a single-page PDF is **Single-page File**. The profile with such input and output values is compatible with Image Converter triggered at level 1.

The following table contains the settings that must be configured for the Image Conversion profile.

**Table 8-6: Image Conversion profile settings**

Property	Value
Input Source	<b>Multi-page Document (Document Level)</b>
Resolution ( <i>DPI</i> )	<b>300</b>
Page Size	<b>Use Document Default</b>
Orientation	<b>Use Document Default</b>
HTML Rendering Engine	<b>Microsoft Word</b>
Office Documents Rendering Engine	<b>Microsoft Office Engine</b>

Property	Value
Output Destination	<b>Single-page File</b>
File Type	<b>ADOBE (*.PDF)</b>
Merge Annotations	Off (Clear the check box)
Color Format	<b>Binary</b>
Compression	<b>CCITT Group 4</b>
Compatibility Level	<b>Acrobat 4.0</b>
Keep Input Textual Data	On (Leave as default)
Keep Input PDF Bookmarks	Off (Leave as default)
Keep Input PDF Catalog and Dictionary	Off (Leave as default)
Document Information Source	<b>First Document</b>
Keyword Separator	Leave blank
Conversion Timeout (seconds)	<b>600</b>

**Summary:**

In this scenario, textual data from the input documents is maintained as text while splitting. When **Keep Input Textual Data** is selected, the **Color Format** and **Compression** settings are ignored for input text files so these settings can be left as default.

#### 8.4.4 Converting Microsoft Office multi-page documents into multi-page PDF documents

**Prerequisites:**

The documents are imported into a capture system using Standard Import. The customer does not have Microsoft Office installed on the machine.

**Goal:**

The customer needs multi-page Microsoft Word documents to be converted to multi-page *PDF* files. The textual data from the input documents must be retained in the output PDF file. The compatibility with the oldest available Adobe® Acrobat® version is required.

**Solution:**

After Standard Import, the Image Converter module will take input files for processing at document level (level 1) which corresponds to **Multi-page Document (Document Level)** selected as **Input Source**. The **Output Destination** for a multi-page PDF is **Multi-page Document (Task Level)**. The profile with such input and output values is compatible with Image Converter triggered at level 1 or higher.

The following table contains the settings that must be configured for the Image Conversion profile.

**Table 8-7: Image Conversion profile settings**

Property	Value
Input Source	<b>Multi-page Document (Document Level)</b>
Resolution ( <i>DPI</i> )	300
Page Size	<b>Use Document Default</b>
Orientation	<b>Use Document Default</b>
HTML Rendering Engine	<b>Embedded Image Converter Engine</b>
Office Documents Rendering Engine	<b>Embedded Image Converter Engine</b>
Output Destination	<b>Multi-page Document (Task Level)</b>
File Type	<b>ADOBE (*.PDF)</b>
Merge Annotations	Off (Clear the check box)
Color Format	<b>Binary</b>
Compression	<b>CCITT Group 4</b>
Compatibility Level	<b>Acrobat 4.0</b>
Keep Input Textual Data	On (Leave as default)
Keep Input PDF Bookmarks	Off (Leave as default)
Keep Input PDF Catalog and Dictionary	Off (Leave as default)
Document Information Source	<b>First Document</b>
Keyword Separator	Leave blank
Conversion Timeout (seconds)	<b>600</b>

**Summary:**

In this scenario, embedded Image Converter engine is used for processing non-image documents. Textual data from the input documents is maintained as text while converting. When **Keep Input Textual Data** is selected, the **Color Format** and **Compression** settings are ignored for input text files so these settings can be left as default.

## 8.4.5 Converting emails into multi-page PDF documents

### Prerequisites:

The documents are imported into a capture system as attachments using Standard Import (Email Type).

### Goal:

The customer needs both HTML pages (email body) and attachments to be converted to multi-page *PDF* files. The textual data from the input documents must be retained in the output PDF file. The compatibility with the oldest available Adobe® Acrobat® version is required.

### Solution:

After Standard Import, the Image Converter module will take input files for processing at page level (level 0) which corresponds to **Multi-page Document (Page Level)** selected as **Input Source**. The **Output Destination** for a multi-page PDF is **Multi-page Document (Task Level)**. The profile with such input and output values is compatible with Image Converter triggered at level 1.

The following table contains the settings that must be configured for the Image Conversion profile.

**Table 8-8: Image Conversion profile settings**

Property	Value
Input Source	<b>Multi-page Document (Page Level)</b>
Resolution ( <i>DPI</i> )	300
Page Size	<b>Use Document Default</b>
Orientation	<b>Use Document Default</b>
HTML Rendering Engine	<b>Microsoft Word</b>
Office Documents Rendering Engine	<b>Microsoft Office Engine</b>
Output Destination	<b>Multi-page Document (Task Level)</b>
File Type	<b>ADOBE (*.PDF)</b>
Merge Annotations	Off (Clear the check box)
Color Format	<b>Binary</b>
Compression	<b>CCITT Group 4</b>
Compatibility Level	<b>Acrobat 4.0</b>
Keep Input Textual Data	On (Leave as default)
Keep Input PDF Bookmarks	Off (Leave as default)
Keep Input PDF Catalog and Dictionary	Off (Leave as default)

Property	Value
Document Information Source	First Document
Keyword Separator	Leave blank
Conversion Timeout (seconds)	600

**Summary:**

In this scenario, virtual printing technology is used for processing non-image documents. Textual data from the input documents is retained as text while converting.

### 8.4.6 Processing PDF attachments

When Image Converter extracts attachments, it places extracted files into IA Values. To further process these files for review, conversion or enhancement, you must add additional IA Values and place the attachments in them. For example, you could create the following sample process, which would use the .NET Code module and the sample code in [Example 8-1, “.NET Code module sample code” on page 256](#) to place the attachments in the IA Values.

**Table 8-9: Sample process for processing PDF attachments: included steps**

Step name	Module name (trigger level)	Description
StandardImport	Standard Import (7)	Imports the files, including PDFs and their associated attachments.
ExtractAttachments	Image Converter (1)	Splits the input files, extracts the attachments from the PDFs, and then places the attachments into IA Values.
CreateAdditionalNodes	.NET Code (7)	Creates an additional page node for each attachment and places the extracted files into these nodes.
ConvertAttachments	Image Converter (1)	Converts the extracted attachments.
Completion	Completion (1)	Operator reviews the attachment images.
StandardExport	Standard Export (7)	Exports attachment images to the appropriate backend system.

 **Example 8-1: .NET Code module sample code**

```
namespace CreateNodes4Attachments
{
```

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Text;
using Emc.InputAccel.CaptureClient;
using Emc.InputAccel.UimScript;
using System.IO;

public class CreateNodes : CustomCodeModule
{
    private static string imageConverterStepName = "ExtractAttachments";

    public override void ExecuteTask(IClientTask task, IBatchContext batchContext)
    {
        IBatchNodeCollection documentNodes =
batchContext.GetStepNode(task.BatchNode, imageConverterStepName).GetDescendantNodes(1);
        IBatchNodeCollection pageNodes = batchContext.GetStepNode(task.BatchNode,
imageConverterStepName).GetDescendantNodes(0);

        foreach (IBatchNode document in documentNodes)
        {
            int extractedCount =
document.NodeData.ValueSet.ReadInt("AttachmentsExtractedCount");
            for (int i = 0; i < extractedCount; i++)
            {
                IBatchNode newPage = document.AddNewChild(-1);
                string fileextension =
document.NodeData.ValueSet.ReadString("AttachmentFileType" + (i+1));
                newPage.NodeData.ValueSet.WriteString("Level0_OutputFileExtension",
fileextension);
                newPage.NodeData.ValueSet.WriteBoolean("IsAttachment", true);

                var fileData = document.NodeData.ValueSet.ReadFile("AttachmentFile"
+ (i+1));
                newPage.NodeData.ValueSet.WriteFileData("Level0_OutputImage",
fileData.ReadBytes(), fileextension);
            }
            task.CompleteTask();
        }

        public override void StartModule(ICodeModuleStartInfo startInfo)
        {
        }
    }
}
```





## Chapter 9

# Designing a profile for optical character recognition processing

The topics within this section introduce **Standard OCR** functionality, describe the main features and functions of the **Standard OCR** profile editor, and cover information on how to create, manage, configure, and test **Standard OCR** profiles.

### 9.1 What is Standard OCR?

Standard OCR lets you process different types of electronic documents and images by applying an appropriate OCR engine processing mode to each of them.

It can be used for:

- Extracting textual data from textual documents.
- Extracting textual data from the documents that contain mixed content (for example, text, tables, graphical elements, etc.) by running the full page OCR reading and then refining the results by extracted electronic text.
- Extracting textual data from images by running the full page OCR reading.
- Generating OCR data cache that can be used by other modules in a capture process. The OCR data cache is required for the Click to Extract feature in the Identification and Completion modules.
- Producing searchable *PDF* and text files with the textual data extracted from images.

For the full list of supported image and non-image file formats that the Standard OCR module can handle, see *OpenText Intelligent Capture - Operating Specifications (ECPCORE-RLI)*.

#### Related Topics

[“How does Standard OCR functionality work?” on page 260](#)

## 9.2 How does Standard OCR functionality work?

Standard OCR functionality is handled by the Standard OCR module which is described in *OpenText Intelligent Capture - Recognition and Advanced Recognition Modules Guide (ECPCORE-CMR)*.

You can create various Standard OCR profiles for every specific business scenario. In a Standard OCR profile, you select the most effective OCR engine processing mode for handling a particular input file type of content. Also, the profile specifies the output result options, processing options, and the error handling settings.

Once a Standard OCR profile has been created, it must be deployed to the capture server. In a capture process, the **StandardOCR** step is set up by selecting one of the deployed Standard OCR profiles.

### Related Topics

[“Getting started” on page 75](#)

[“Deploying a component” on page 433](#)

## 9.3 High-level steps

Perform the following high-level steps to configure and set up your Standard OCR profiles.

**Table 9-1: Standard OCR profiles: high-level steps**

Step	Description	See
1	The entry point to get started with Standard OCR profiles is the <b>Standard OCR Profiles</b> list where you can create and manage profiles. To access the list, select the <b>Standard OCR</b> area.	<a href="#">“Managing Standard OCR profiles” on page 262</a>
2	Add a new Standard OCR profile.	<a href="#">“Adding a new profile” on page 263</a>

Step	Description	See
3	Configure the Standard OCR profile as follows: <ol style="list-style-type: none"> <li>1. Define <b>Processing Mode</b> applicable for handling your type of documents.</li> <li>2. Specify <b>Processing Options</b> for those cases when the full page OCR reading will be applied for data extraction.</li> <li>3. Specify the output which you need to get as a result of processing:               <ul style="list-style-type: none"> <li>• select whether you need to create the file with OCR data cache to be used further during the Classification, Extraction, Completion, or Identification steps;</li> <li>• the output file format (<i>PDF</i>, Text, or both).</li> </ul> </li> <li>4. Specify <b>Error Handling</b> options.</li> </ol>	<a href="#">“Standard OCR profile editor” on page 264</a>
4	Save the profile.	<a href="#">“Standard OCR profiles list” on page 262</a>
5	Test your profile.	<a href="#">“Testing a Standard OCR profile” on page 275</a>
6	Upload the newly created profile to the capture server.	<a href="#">“Deploying a component” on page 433</a>
7	Open CaptureFlow Designer and create a CaptureFlow including the <b>StandardOCR</b> step in it. Set up the step by selecting the Standard OCR profile which you have created and uploaded to the server before.	<a href="#">“Managing CaptureFlows” on page 91</a>

## 9.4 Managing Standard OCR profiles

This section introduces the Standard OCR profile list, describes how to use it and provides the procedure of creating a Standard OCR profile.

### 9.4.1 Standard OCR profiles list


To open the panel with the **Standard OCR Profiles** list, select the **Standard OCR** area.

The **Standard OCR Profiles** list displays all the Standard OCR profiles existing in the currently selected capture system. As with other types of profiles, these profiles are global and can be used in any process of this capture system.

The following table describes the layout and options available in the list window.

**Table 9-2: Standard OCR profiles window**

Element	Description
Tabs	The <b>Standard OCR Profiles</b> tab displays the list of all Standard OCR profiles that are stored locally in the currently selected capture system.  Other tab buttons are displayed for opened Standard OCR profiles. Close individual profiles by clicking <b>X</b> on a profile tab.
<b>Standard OCR Profile</b> column	The column displays names of Standard OCR profiles sorted alphabetically. To open a profile, double click it.
<b>Description</b> column	For each profile, the column contains its description or usage instructions. To edit a description of the profile, click the existing description.
<b>File Time</b> column	For each profile, the column contains the date and time when this profile was last saved on the local machine.
<b>Open</b> button	Opens the selected Standard OCR profile for viewing or editing.
<b>New</b> button	Creates a new Standard OCR profile as described in <i>“Adding a new profile”</i> on page 263.

Element	Description
Delete button	<p>Deletes the selected Standard OCR profile from the capture system and physically from your machine.</p> <p> <b>Note:</b> If you delete a profile that is set up for a <b>StandardOCR</b> step in a CaptureFlow, the process logic is not affected unless the profile exists on the server.</p>

### Related Topics

[“Standard OCR profile editor” on page 264](#)

## 9.4.2 Adding a new profile

### To add a profile:

1. Click **Standard OCR**.
2. In the **Standard OCR Profiles** list window, click **New**. The **Standard OCR Profile - New** window opens.
3. In the window, specify a profile name following the [naming convention](#).
4. Save the profile. The new profile is added to the **Standard OCR Profiles** list. The Standard OCR profile editor opens this new profile for editing.



**Note:** Standard OCR profiles are stored on the Designer host machine. Profiles are saved as configuration files in the following folder:

```
C:\Users\\Documents\

```

### Related Topics

[“Standard OCR profiles list” on page 262](#)

## 9.5 Standard OCR profile editor

This section describes the Standard OCR profile editor properties and provides information on how to test the created profile with the current settings.



Standard OCR profile editor property sheet contains the following tabs:

- **Profile Property:** includes settings that can be configured for a Standard OCR profile, grouped into categories.
- **Value:** an actual value of the setting.
- **Tips:** helpful text to describe the setting.


“Standard OCR profile properties” on page 264 provides a description of Standard OCR profile properties.

**Table 9-3: Standard OCR profile properties**

Category	Property	Description
Processing Mode	Processing Mode	<p>Specify the OCR engine processing mode to be applied to the input files that the Standard OCR module receives. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>VoteOcrAndEText</b> (default): Select this option if your input documents contain mixed content. The data from file is extracted by running the full page OCR reading. Where possible, electronic text is also extracted and the results are used to refine the OCR results.</li> <li>• <b>OcrFromImage:</b> Select this option if your input documents are images / contain images only. The data from file is extracted by running the full page OCR reading.</li> <li>• <b>ExtractFromEText:</b> Select this option if your input files are <i>PDF</i> files that contain textual data only. Electronic text is extracted natively, as is.</li> </ul>


Category	Property	Description
<p><b>Processing Options</b></p> <p>Specify the options to process input documents (only for those cases when the full page OCR reading will be applied for data extraction).</p> <p> <b>Note:</b> The setting group is disabled if <b>ExtractFromEText</b> is selected as processing mode.</p> <p><b>Output Result</b></p> <p>Specify the options to generate the output results.</p>	<p><b>Country or language</b></p>	<p>Select a country/language to define the country/language peculiarities of the documents being processing. Multiple selections are allowed.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• If no country/language is selected, then <b>USA</b> is used.</li> <li>• For better performance, consider selecting only actually-used countries/languages.</li> <li>• Some of the country/language combinations might be not available. We recommend to <b>test</b> the profile before using to ensure the selected combination is valid.</li> </ul>
	<p><b>Restrict Output to Selected Character Set</b></p>	<p>Define how to process character codes that are not related to the currently chosen countries/languages. If selected, all codes incompatible with the specified country are suppressed. If cleared, the specified country character set will be ignored, and any character set can be a part of the recognition result.</p>


Category	Property	Description
	<p><b>OCR Options</b></p> <p>Specify the OCR engine options to process input documents.</p>	<p><b>Reader:</b> Select one of the following recognition engines:</p> <ul style="list-style-type: none"> <li>• <b>AEGReader:</b> Only available if <b>Processing Mode</b> is set to <b>OcrFromImage</b>. It is not available for some countries/languages as specified in <b>Country or language</b>, including the following: <ul style="list-style-type: none"> <li>– Azerbaijan (Cyrillic and Latin)</li> <li>– Belarus</li> <li>– Bulgaria</li> <li>– Greece</li> </ul> </li> <li>• <b>RecoStar</b></li> <li>• <b>Voter (default):</b> This is the recommended recognition engine selection. Although recognition could take longer, it provides optimal recognition results for most cases because it selects the best results from the character alternatives of both RecoStar and AEGReader. However, Voter results could differ dramatically from either RecoStar or AEGReader by themselves.</li> </ul>

Category	Property	Description
		<p><b>Input Image Resolution:</b> Select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>ReadFromFile:</b> The resolution as specified in the input file is used. The value is recommended for scanners.</li> <li>• <b>Estimate (default):</b> A suitable resolution is estimated. This value is recommended when images are from both scanners and cameras (also see the recommendations for <b>Repair Resolution</b>) and some languages/countries (for example, Chinese (Traditional, Hong Kong)).</li> </ul> <p><b>Repair Resolution:</b> Select to overwrite resolution values that are less than 97 DPI with a calculated optimal resolution value. Selecting this property is recommended for progressive JPEG images; for example, camera images.</p> <p>Clear the check box to preserve original image resolution values, even if they are less than 97 DPI.</p> <p> <b>Note:</b> This property is only available if <b>Input Image Resolution</b> is set to <b>ReadFromFile</b>.</p> <p><b>Auto Rotate Image:</b> Select if automatic image rotation is necessary. The image is rotated for the processing purpose only and will be returned to the original image upon step completion.</p>



Category	Property	Description
		<p><b>Text Orientation:</b> Specify the orientation of text in a page zone. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Clockwise:</b> Clockwise 90 degrees rotation.</li> <li>• <b>CounterClockwise:</b> Counterclockwise 90 degrees rotation.</li> <li>• <b>Normal</b> (default): No rotation.</li> <li>• <b>UpsideDown:</b> 180 degrees rotation.</li> </ul> <p><b>Find Text Blocks:</b> Select to recognize text appeared in a complicated format (tables, columns). The result will contain blocks with the text lines logically grouped. Note that selecting this option might slightly slow down the recognition performance.</p> <p><b>Number of Text Lines per Segment:</b> Specify an approach on how text lines will be read from a page with a complex layout while extracting. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>TextLine Segments:</b> Text information is extracted by areas. In results, each segment of text can be stored as a random number of text lines. (For example, if a document contains columns, text data will be extracted by lines within a column).</li> <li>• <b>TextLines:</b> Text information can be extracted out of layout area. In results, each segment of text is stored as a single text line. (In the same example, text data will be extracted by lines extended through the columns).</li> </ul>


Category	Property	Description
		<b>Machine Type Confidence:</b> Set a threshold range from 0 to 255 for rejecting low-confidence or unrecognized characters.
	<b>Post-processing</b> Define how to post-process the extraction results.	<b>Logical Context Verification:</b> Mainly enforces homogeneous strings (uppercase only, lowercase only, digits only). If selected, trigram is enabled. If cleared, trigram is disabled.
		<b>Trigram Verification:</b> Uses statistical language models. If selected, boundary handling is enabled. If cleared, boundary handling is disabled.

Category	Property	Description
		<p><b>Dictionary File:</b> Specify the path to your custom dictionary file. The custom dictionary is used before the standard dictionaries of the selected countries/languages. Dictionaries are intended to be used for recognition of alphanumeric, machine-printed text.</p> <p>Custom dictionary requirements are as follows:</p> <ul style="list-style-type: none"> <li>• Format           <ul style="list-style-type: none"> <li>– Each dictionary phrase must reside on a single line only.</li> <li>– The end of each line is delimited by a carriage return (0x0D) plus a linefeed (0x0A) or a linefeed by itself.</li> <li>– The words in a phrase are separated by a single space.</li> <li>– All characters must be in upper case.</li> </ul> </li> </ul> <p> <b>Note:</b> Spaces at the end of lines are ignored.</p> <ul style="list-style-type: none"> <li>• File character encoding: ASCII</li> <li>• Valid file extensions:           <ul style="list-style-type: none"> <li>– Central Europe               <ul style="list-style-type: none"> <li>○ ce</li> <li>○ 1250</li> </ul> </li> <li>– Cyrillic               <ul style="list-style-type: none"> <li>○ cy</li> <li>○ 1251</li> </ul> </li> <li>– Latin 1               <ul style="list-style-type: none"> <li>○ we</li> <li>○ 1252</li> </ul> </li> <li>– Greek               <ul style="list-style-type: none"> <li>○ gr</li> <li>○ 1253</li> </ul> </li> </ul> </li> </ul>

Category	Property	Description
		<ul style="list-style-type: none"> <li>– Turkish               <ul style="list-style-type: none"> <li>○ tr</li> <li>○ 1254</li> </ul> </li> <li>– Baltic               <ul style="list-style-type: none"> <li>○ ba</li> <li>○ 1257</li> </ul> </li> <li>• The directory must provide write permission to Standard OCR.</li> </ul> <p><b>Dictionary Mode:</b> Select whether your custom dictionary (as specified in <b>Dictionary File</b>) has a complete list of the words in a specified domain or the list is incomplete as follows:</p> <ul style="list-style-type: none"> <li>• <b>Complete:</b> Replaces all unknown words with their closest match in the custom dictionary. For example, all of the months would be a complete dictionary.</li> <li>• <b>Incomplete (default):</b> Unknown words are not changed. For example, a list of names of all the people in a country is likely incomplete.</li> </ul>
	<b>Generate OCR Data Cache</b>	<p>Specify if you need to generate the file containing the OCR data cache for the page. If selected, the file will be created.</p> <p> <b>Note:</b> For a multi-page document, OCR data cache will be generated for the first page of the document only. Consider to add the <b>ImageConverter</b> step prior to <b>StandardOCR</b> for splitting multi-page documents to pages.</p>

Category	Property	Description
	<b>Resolution (DPI)</b>	<p>Enabled when <b>Generate OCR Data Cache</b> is selected.</p> <p>Specify the resolution (in DPI) for the output file which contains the OCR data cache. This resolution must conform the resolution determined in the recognition project. Available only for PDF input files.</p>
	<b>File Formats</b>	<p>Specify the format of the output file. Multiple selections are allowed, however the order of selection is sensitive and changes dynamically in the <b>File Formats</b> list. The first selected value corresponds to the <b>OutputFile1</b> and <b>OutputFileExt1</b> IA values and the second selected value corresponds to the <b>OutputFile2</b> and <b>OutputFileExt2</b> IA values accordingly. See the following:</p> <ul style="list-style-type: none"> <li>• PDF</li> <li>• Text</li> </ul>
	<b>PDF Output Options</b>	<p><b>PDF Version and Compatibility:</b> Define the version of Adobe Acrobat to create the output PDF file.</p> <p><b>Compression:</b> Set the compression level to apply to the text in the output PDF file.</p>

Category	Property	Description
		<p><b>Save Image:</b> Select the type of the image that will be saved into the output PDF file. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>NoImage</b> (default): Only extracted text will be included into the output file.</li> <li>• <b>OriginalImage:</b> Extracted text will be included and the source image will be set as a background for the page in the output file.</li> <li>• <b>ResultImage:</b> Extracted text will be included and the processed image will be set as a background for the page in the output file.</li> </ul> <p><b>Binary Format:</b> Check to save image to binary format.</p> <p> <b>Note:</b> Only available if <b>SaveImage</b> is set to <b>ResultImage</b>.</p> <p><b>JPEG Compression Level:</b> Set the image's JPEG compression level. The higher the number, the higher the compression level. Increasing compression level decreases image size but lowers image quality.</p> <p> <b>Note:</b> Only available if <b>SaveImage</b> is set to <b>ResultImage</b>.</p>

Category	Property	Description
		<p><b>Resolution Limit (DPI):</b> Set the maximum resolution level for the image (color, grayscale, or binary). If the image resolution is greater than the specified maximum, then the image is scaled down to the maximum; otherwise, the image resolution is not changed.</p> <p> <b>Note:</b> Only available if <b>SaveImage</b> is set to <b>ResultImage</b>.</p>
<p><b>Error Handling</b></p> <p>Define the options for handling production errors.</p>	<p><b>Processing Timeout (in seconds)</b></p>	<p>Set the time-out value from 60 to 3600 for the Standard OCR module to wait while processing of the current task is being performed. If exceeds, an error will be processed as specified in the <b>When Error Occurs</b> option.</p>
	<p><b>When Error Occurs</b></p>	<p>Select the action to perform in case of error. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Abort entire task:</b> Aborts the entire task, sets the status of the task as failed, and accepts the new task. The <b>ErrorText</b> IA value is returned.</li> <li>• <b>Skip error node and continue task:</b> Marks the current node as processed with error, skips the current node, and continues to process the other nodes in the task by order. The status of the task is not set as failed.</li> </ul>

**Related Topics**

“Testing a Standard OCR profile” on page 275

## 9.5.1 Testing a Standard OCR profile

Once you have completed configuring your Standard OCR profile, you can test it to see how the current settings work and trying to find the best settings for most accurate OCR results.



**Note:** Testing is not available if you do not have the Advanced OCR/ICR engine installed. Install it from the **Extraction Engines** section of the installer and restart Designer to apply these changes.

### To test a profile:

1. Open the profile which you want to test.
2. Click **Test** in the top-right corner of the profile window.
3. Select the file(s) that you want to process with the Standard OCR module. For the list of supported file formats, see *Supported File and Image Formats* of the *Supplemental Reference*.
4. The selected files appear in the **Source Files** list. To delete a file from the list, select the line and press **DELETE**.
5. Check that the requested output formats list corresponds to those values that you have set in the **Output Result** options of the profile.
6. Select the location to save the extraction results. By default, the results will be saved to the same folder where the source files are stored.

When **Apply this path for the next test** is selected, the current path will be used to save your results during the next test.

7. Click **Run**.

When testing is performed, the status is shown for the each document in the list. In case of failure, the **Error Reason** column contains the description of the error.

The status bar at the bottom of the window displays the amount of the processed files and the amount of the processed files added to the result location. Click the link to see the results. A result file name contains the source file name including the extension. For example, for the `test.png` source file, the following output files might be created: `test.png.pdf`, `test.png.txt`, and `test.png.ocr`.



## Chapter 10

# Designing a profile for exporting images and data

Export profiles define the images and data to export and the target format (such as text or CSV) and the location to which to export them (such as a file system or OpenText Content Server).

The Standard Export module uses export profiles to execute the actual export.

### 10.1 High-level procedure



1. Create an export profile by clicking **Export** , **New**, and then specifying the profile name.

The export profile tab with the specified name is displayed.



**Note:** Make sure to follow the naming conventions.

2. Define the export profile on its export profile tab.

See [“Defining an export profile” on page 278](#).

#### Related topics

[“Listing all, creating, opening, and deleting export profiles” on page 277](#)

[“Setting profile properties” on page 283](#)

[“Defining filters” on page 283](#)

[“Defining export commands” on page 285](#)

[“Naming Conventions” on page 481](#)

### 10.2 Listing all, creating, opening, and deleting export profiles



Click **Export**  to display the **Export Profiles** tab, which displays all of the export profiles that are available in the current capture system; on this tab, you can also create profiles and delete or open existing ones by clicking the corresponding buttons.



#### Notes

For more information, see the following:

- **File time** is the timestamp of the profile and is based on the date and time settings of your local machine.

- Deleting an export profile removes it from the capture system and your local machine. However, even if the deleted export profile is used in an installed process, the installed process’s logic remains unaffected.

**Related topics**

“Defining an export profile” on page 278

“Setting profile properties” on page 283

“Defining filters” on page 283

“Defining export commands” on page 285

“Naming Conventions” on page 481

**10.3 Defining an export profile**

On an export profile’s tab, you define the export profile by performing the following actions:

1. Set the export profile’s properties by clicking **Properties**.  
See “Setting profile properties” on page 283.
2. Specify the nodes to be selected by clicking **Add Filter**.  
See “Defining filters” on page 283.
3. For each filter, define export commands for the desired output formats and target locations by clicking **Add Export**.  
See “Defining export commands” on page 285.

For information about all of the actions available on the export profile tab, see the following table.



**Note:** An asterisk sign next to the profile name tab indicates that there are unsaved changes in the profile.

**Table 10-1: Export profile tab: actions**

Button	Action and tips
Add Export	Inserts an export command after the selected export command or, if a filter is selected, as the first export command in a filter.
Add Filter	Inserts a filter after the selected filter (or after the filter in which a command is selected).
Delete	Deletes the selected filter (and its export commands) or export command.

Button	Action and tips
<b>Intended CaptureFlow</b>	(Intelligent Capture only) Specifies a CaptureFlow to enable autocompletion of the CaptureFlow's IA values in expressions.  Only CaptureFlows compiled without errors are displayed. Also, if a previously selected CaptureFlow has an error, then it would no longer be displayed nor would autocompletion be enabled.
<b>Minimum Task Level</b>	(Intelligent Capture only) (Read-only) Displays the highest file level of all of the export commands.  The associated step on the CaptureFlow diagram must get the <i>trigger level</i> equal or higher than the minimum task level.
<b>Move Down</b>	Moves the selected filter (and its export commands) or selected export command down a level.
<b>Move Up</b>	Moves the selected filter (and its export commands) or selected export command up a level.
<b>Properties</b>	Specifies the locale and default formats for the date and number data types.

## Related topics

[“Setting profile properties” on page 283](#)

[“Defining filters” on page 283](#)

[“Defining export commands” on page 285](#)

### 10.3.1 Using autocompletion

When you create an expression, you can benefit from the autocompletion feature. This contextual prompting help displays the steps of an intended CaptureFlow and IA values available on those steps, and helps with syntax.

Autocompletion displays the IA values that are known at design time. However, some IA values are created or get known at runtime only. If you know about these IA values, you can use them in expressions as well, even though they are not displayed in the autocompletion help.



**Note:** Autocompletion displays MDF values and document type values only if the associated CaptureFlow or a document type compiled without errors. If you modify a linked CaptureFlow or a document type to introduce an error,

their values disappear from autocompletion help (but not from expressions in which they are added).

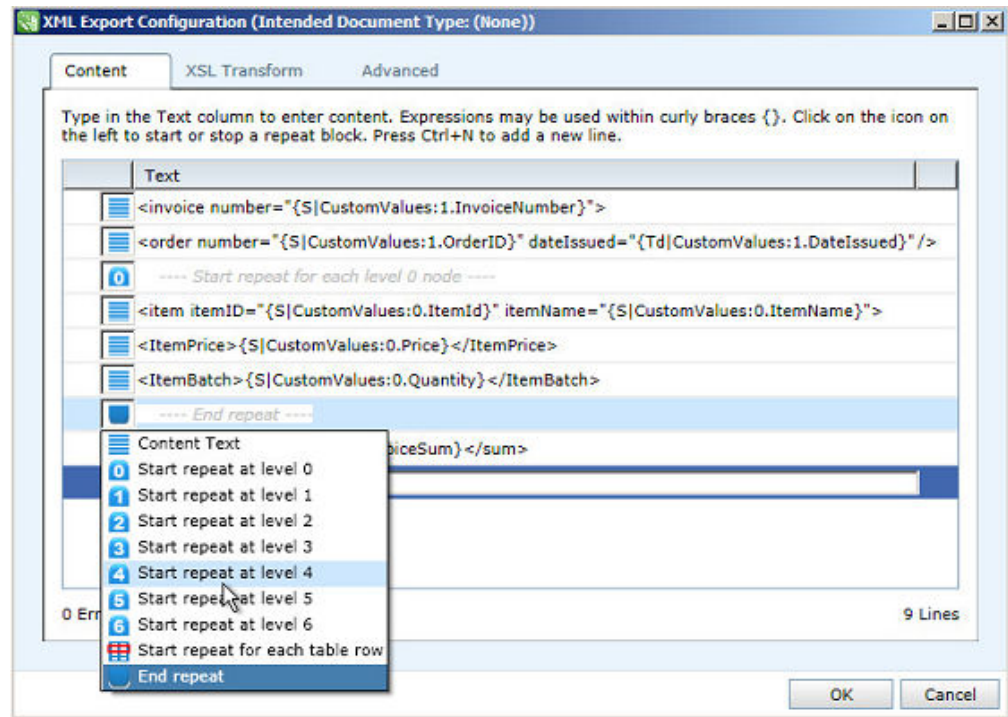
The following table summarizes the information about IA values that can be used in expressions in your export profile.

**Table 10-2: IA values: availability in autocompletion**

IA value source	Description	Autocompletion available?
MDF file	IA values predefined in the MDF file of the module that is used as a step in the linked CaptureFlow.	Yes
Custom MDF file	Custom-specific IA values defined in the CaptureFlow tool for the step in the linked CaptureFlow. These IA values are saved in the custom MDF file.	Yes
Built-in values	Step-specific non-MDF IA values, such as <i>&lt;BatchHold&gt;</i> , <i>&lt;BatchError&gt;</i> , <i>&lt;ErrorStatus&gt;</i> , <i>&lt;Compile_TriggerState&gt;</i> , <i>&lt;Ready&gt;</i> , and <i>&lt;RetriesLeft&gt;</i> .	Only <i>&lt;Ready&gt;</i> is available. Other built-in values are not populated to the autocompletion help.
Dynamically created (\$Runtime) IA values	IA values can be defined in expressions for other steps of the linked CaptureFlow. These IA values are created at runtime.	No
Document type fields	IA values that define the fields of the document type.	Default: no.  An export command can link a specific document type explicitly and have that document type's fields displayed in autocompletion help.  Filters cannot link document types to have their fields displayed in autocompletion help. However, you can add document type fields to filtering expressions manually.

## 10.3.2 Using the embedded content editor

The content editor is the tool embedded in the Export profiles editor. This tool serves for creating the content definition for XML files, Text files, and emails that will be generated by export commands.



**Figure 10-1: XML export configuration**

The content editor allows you to include *content lines* and *repetition blocks*. Content lines include text that will be added to the output file. The text lines may include *expressions* that are resolved to strings and inserted into the generated text at runtime. An expression may be an *IA value data path*, a *format expression*, or a data path to a *document type field*. Expressions are added in curly brackets.

If the inserted IA value references a node below the file level (**Each Level** parameter of the command), the first found node will be handled and other nodes ignored. If the node is not found, the expression returns ?. If you need to iterate all nodes on that level, or all rows in the specified document type table, include a *repetition block*.

A repetition block consists of the **Start repeat** line, the **End repeat** line, and the content lines inside. The content inside the repetition block will be added to the output file or email for each iterated node or table row. The content editor supports nested repetition blocks as well.

The following rules will help you manage lines and text in the content editor:

- **To add lines to the editor:** press **CTRL+N**. A new line is added below the currently selected line. You can also drop the cursor to the typing area of the text line and break it by pressing **ENTER**.
- **To specify the line type:** click the icon that starts the line and select the required option from the drop-down list.
  - **Content Text:** defines the content line.
  - **Start repeat at level [0–6]:** starts the repetition block. This line requires the content lines to be repeated in the output files, and the End repeat line be added at the end.
  - **Start repeat for each table row:** starts the repetition block for each row of the table defined in the document type.

Replace the **<Specify table>** placeholder with the path of the intended table. The table data path format is `<stepname>:<level>.UimData.<tablename>`.

If you link your export command to a document type on the **Advanced** tab, you can select a table name from the pre-populated list of tables defined in the linked document type.
  - **End repeat:** closes the repetition block.
- **To replace or duplicate the text:** You can copy, cut, and paste the text. You cannot cut, copy, and paste lines with text, or 'start repeat' and 'end repeat' lines. You cannot drag and drop lines.
- **To change the type of line:** Click the icon that starts the line and select a different option from the list.



**Note:** You cannot undo this change. If you change the content text line to a different type, the content of this line is lost and cannot be recovered.


- **To delete lines:**
  - **Any type of line:** Select the line out of the typing area and pressing the **DEL** keystroke.
  - **Content text line:** Drop the cursor at the beginning of the typing area and press the **BACKSPACE** keystroke. This action removes the break between the current line and the previous text line. If the previous line is not the content text line, use **DEL**.
  - **Repetition block(all lines):** Select all lines that need to be deleted, and press **DEL**.

## 10.4 Setting profile properties

In the export profile, click **Properties** and specify the properties outlined in “[Export profile: properties](#)” on page 283.

**Table 10-3: Export profile: properties**

Property	Description
<b>Locale</b>	The locale that determines the default formatting of values in the profile.
<b>Default Format for Dates</b>	The default format for dates. Click the arrow or manually enter a format (as described in the <b>Tips</b> column).  For more information about supported format specifiers, see the Microsoft .NET documentation for custom date and time format strings.
<b>Default Format for Numbers</b>	The default format for numbers. Click the arrow or manually enter a format (as described in the <b>Tips</b> column).  For more information about supported format specifiers, see the Microsoft .NET Framework documentation for custom numeric format strings.

 **Note:** The data format samples (in the **Default Format for Dates** and **Default Format for Numbers**) are formatted using the Designer locale; however, at runtime, they are formatted according to the runtime locale.

## 10.5 Defining filters

Filters define the conditions for selecting nodes from a batch. If a node matches the condition, then the node and all of its child nodes are selected. Then, the export commands are executed on those nodes.

### Notes

- The same node can be selected by more than one filter.
- Filters and their export commands are executed as ordered from top to bottom.

Only the lowest level in the conditional expression is evaluated. For example, for level 7 (batch), all nodes can be selected.

To define the nodes to be selected, perform one of the following actions:

- To select all nodes in the batch, keep **All**.
- To select a subset of nodes, click **All** and specify a valid expression.



#### Notes

- Format expressions are not supported.
- Boolean expressions do not require curly braces around value paths and included functions.



#### Notes

For more information, see the following:

- Document type fields are not displayed in autocomplete help. However, you can add them in your filter expression manually using the following syntax:

```
<stepname>:<level 0–7>.UimData.<fieldname>
```

- For example, the following expression selects all documents (that is, level 1 nodes) issued for the MyCompany customer (assuming that *CustName* is a custom value of type String):

```
ScanPlus:1.CustName="MyCompany"
```

- For example, the following filter condition includes values at both levels 1 and 2:

```
CustomValues:2.ID>123 AND CustomValues:1.Age=15
```

All level 1 nodes, including all of their level 0 child nodes, in the batch that match the filtering condition are selected, but the level 2 nodes are not selected.

- Because the Standard Export module is triggered at its associated step's **Level**, only the nodes at the step's level and lower can be selected.

#### Related Topics

[“Using autocomplete” on page 279](#)

[“Defining export commands” on page 285](#)

[“Programming Reference—Expressions” on page 457](#)

## 10.6 Defining export commands

1. To specify an export type, click **Export: (Select Type)**.



### Notes

For more information, see the following:

- To move your export command to a different position in the filter, use **Move Up** and **Move Down**. You can also copy or cut and paste export commands between filters.
  - To delete an export command, use **Delete**.
  - To specify a description for the export command, click **Comment** (directly below **Export: <export\_type>**).
2. To specify the filtered nodes for which the export command is to generate output, specify the level in **Each Level**.
  3. To specify a connection for each export type command destination, click **To: <export\_type>** and select an existing connection or create one as follows:

- **ApplicationXtender**

To create a connection, see [“Creating an ApplicationXtender connection” on page 47](#).

- **CMIS**

To create a connection, see [“Creating a CMIS repository connection” on page 45](#).

- **Content Server**

To create a connection, see [“Creating a Content Server connection” on page 48](#).

- **xECM** (xECM is an OpenText Content Server-based business application)

To create a connection, see [“Creating a Content Server connection” on page 48](#).

- **CSV, File, Text, and XML**

Specify the destination as outlined in the following table.

Action	To:<type>	Description
Export to a batch IA value	<b>Batch</b>	Select the IA value. <b>Note:</b> Does not apply to <b>File</b> export.

Action	To:<type>	Description
Export to a file on disk	<b>File</b>	Specify the path on disk (a regular string or a format expression) and a file system connection. The path must include the file name and extension. IA values and functions must be included in curly braces. Examples:  c:\temp\{S _Node:0.NodeID}.csv,  c:\temp\export\{S _Batch.BatchName}-Image{TMM-dd-ss-ffff Now()}.{S ScanPlus:0.ImageOutputFileExtension}  To learn more about format expressions, see <a href="#">“Programming Reference – Expressions”</a> on page 457.
Export to an FTP or SFTP site	<b>FTP</b>	Specify the FTP or SFTP (SSH File Transfer Protocol) connection and destination path on the FTP/SFTP site to which to export the files and data. See <a href="#">“Creating a file system connection”</a> on page 41.

- **Email**

To create a connection, see [“Creating an email connection”](#) on page 42.

- **ODBC**

To create a connection, see [“Creating an ODBC connection”](#) on page 36.

4. To specify the properties for each export type, click **Export:** <export\_type>.



**Note:** The highest file level among all export commands sets the minimum task level, which is displayed in the **Minimum Task Level** field.

## 10.6.1 Defining ApplicationXtender content


The ApplicationXtender export command specifies the following:

- The files to be exported to the connected ApplicationXtender server.
- The ApplicationXtender application by which the files are to be managed.
- The values that are to be set for the ApplicationXtender application's fields.

### Notes


For more information, see the following:

- Only SSL/TLS-enabled ApplicationXtender connections are supported.
  - For brevity, ApplicationXtender is referred to as *AX*.
  - See your administrator for the AX server's connection information.
1. On the **ApplicationXtender Mapping** tab, specify an AX application by selecting it in the **ApplicationXtender Application** field.

 **Note:** The fields of the AX application are automatically displayed.

2. Specify a value (either a string or expression) in the expression column that maps to the corresponding field in the **ApplicationXtender Field** column.

For more information, see [“AX Export configuration: ApplicationXtender mapping options”](#) on page 288.

 **Tip:** To make a document type's fields available for auto-completion for expressions, select the document type in the **Advanced** tab's **Document Type** field.

3. On the **File** tab, specify the files that you want to be exported and other file options.




For more information, see [“Setting file image options”](#) on page 289.

4. On the **Advanced** tab, specify other options.

For more information, see [“Setting Advanced options”](#) on page 290.

### 10.6.1.1 AX Export configuration: ApplicationXtender mapping options

The following table provides a description of AX mapping options.


Option	Description
<b>ApplicationXtender Application</b>	An AX application.
<b>ApplicationXtender Field</b>	A field of the AX application.
<b>Create index set per node</b>	<p>If this option is selected, then an AX index set is created for each node of the level selected in the <b>Index set node level</b> list. For a UIMData table, only the first row is exported.</p> <p> <b>Note:</b> The AX application must be enabled for multi-indexing.</p> <p>If this option is not selected, then an index set is created for each table row referenced by the mapped table fields. The rows of each table are grouped together. For example, if there are references to two tables, each with two rows, then four index sets are created. The non-table fields are repeated for each row.</p>
<b>Data</b>	Indicates whether the AX field is reference data.
<b>Export by key field</b>	<p>Indicates to use an AX reference key field (single) without also using reference data fields for the initial query.</p> <p> <b>Note:</b> The AX application must have reference key and data fields.</p>
<b>Index set node level</b>	The node level at which an AX index set is to be created.
<b>Intelligent Capture Expression</b>	<p>An expression to set the value for the corresponding AX field.</p> <p>For more information about expressions, see <a href="#">“Programming Reference—Expressions”</a> on page 457.</p> <p> <b>Note:</b> If your profile links an intended CaptureFlow, you can use auto-completion to reference IA values in the expression.</p>
<b>Key</b>	Indicates whether the AX field is a reference key.

Option	Description
<b>Map Automatically</b>	<p>Automatically maps multiple AX fields to their matching value paths (UIMData). The selected <b>Expression</b> field's value path is used as a pattern for the other AX fields' <b>Expression</b> fields. For each AX field name for which there also exists a matching value name, the pattern is copied to the <b>Expression</b> and the value name is substituted. <b>Expression</b> fields that have existing values are not overwritten.</p> <p>The value path could include custom values.</p>
<b>Overwrite mapped data fields</b>	<p>Indicates that reference data fields are overwritten.</p> <p>All existing AX documents that use the same reference key field value are affected.</p>
<b>Required</b>	Indicates whether the AX field is required.
<b>Type</b>	Data type of the AX field.

### 10.6.1.2 Setting file image options

“ApplicationXtender export configuration dialog box: file options” on page 289 describes ApplicationXtender export file options.

**Table 10-4: ApplicationXtender export configuration dialog box: file options**

Parameter	Description
<b>Page file</b>	<p>Select the files to be exported as follows:</p> <ul style="list-style-type: none"> <li>• <b>Page Image</b> Each page is exported as a separate file.</li> <li>• <b>Document PDF File</b> All pages in a single document are exported as a single PDF file.</li> <li>• <b>Batch PDF File</b> All pages in all documents in the batch are exported as a single PDF file.</li> </ul>
<b>Skip first page</b>	<p>If this option is selected, then the first page (regardless of page type) of each document is skipped. Use this option for pages, such as fax cover sheets, where there is possibly index data on the cover sheet, that does not require exporting to the repository.</p> <p> <b>Note:</b> If a document consists of a single page, it is not exported.</p>

Parameter	Description
Submit to full text indexing	The files are submitted for ApplicationXtender Export Index Agent full-text indexing.

### 10.6.1.3 Setting Advanced options

“ApplicationXtender export configuration dialog box: Advanced tab parameters” on page 290 provides a description of ApplicationXtender export advanced tab parameters.

**Table 10-5: ApplicationXtender export configuration dialog box: Advanced tab parameters**

Parameter	Description
Document Type	(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.
Duplicate Index Action	<p>Select error handling for the scenario in which the creation of an AX document results in a duplicate index. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Error:</b> The error is handled as specified in the <b>Error Handling</b> parameter. The task gets an error status.</li> <li>• <b>Overwrite:</b> Replaces all pages in the first document matching all the index values. The original document is not deleted and its internal ID remains the same.</li> <li>• <b>Skip:</b> Does not modify the repository. If there is a document with the same index data, that document is not modified. Index fields are not changed and pages are not added.</li> <li>• <b>Append:</b> Adds the files to the first document matching all the index values. The files are appended as the last pages of the document.</li> <li>• <b>Revision:</b> Continues with the creation of a new document. The original document remains, and a new document is created with all the same index data. The result is multiple documents with the same index data.</li> </ul>

Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	<p>Specifies the number of retries for the <b>Error Handling</b> parameter.</p>
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not needed.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path:</b> Contains the AX REST URL of the document that was created, replaced or merged. Type: String. Level: file level.</li> <li>• <b>&lt;ID&gt;_Status:</b> Specifies the export status. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: node not selected, file not created.</li> <li>– 1: error creating the file, skipped, or: file exists, overwritten with error.</li> <li>– 2: file exists, overwritten.</li> <li>– 3: file exists, skipped.</li> <li>– 4: new file created.</li> <li>– 5: Revision</li> </ul> </li> <li>• <b>&lt;ID&gt;_AXDocID:</b> Contains the AX document ID of the document that was created, replaced, or merged.</li> </ul> <p>The ID must be unique among the profile's commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>

## 10.6.2 Configuring CMIS content

A CMIS export command specifies one of the following:

- The file to be exported to the connected CMIS-compliant server. Each export command would create a single CMIS object and reference the file. The values that are to be set for the CMIS object's attributes are also specified.
- The CMIS folder object to be created. The values that are to be set for the CMIS folder object's attributes are also specified.



**Note:** Only SSL-enabled CMIS connections are supported, except for CMIS connections to Microsoft SharePoint. For CMIS connections to Microsoft SharePoint, only SSL- and TLS-enabled are supported.

1. On the **Export** tab, specify that a standard CMIS folder, document, or a custom type is to be created by selecting the CMIS type in the **Type** field:
  - **Folder**  
A folder is used to organize documents and as such has only attribute values; it does not reference any files.
  - **Document**  
A document has attribute values and references a file.

The attributes of the CMIS type on the CMIS-compliant server are automatically displayed.



### Notes

For more information, see the following:

- Custom types are displayed as children of folder or document. If custom types are not identified as creatable in the repository, then they are grayed out (that is, disabled) in the **Type** field.
  - The exact attributes that are available are determined by the CMIS-compliant server to which the command connects.
  - For more information about the **Export** tab fields, see [“Export options” on page 293](#).
2. When you select a **Secondary Type** supported by the repository, the attributes of the secondary type are displayed. You can select multiple secondary types from the drop-down box.
  3. Specify the file that you want to be referenced by the CMIS **Document** in the **File value** field.
  4. Specify either a string or expression in the **Value** column to generate a value for the corresponding CMIS attribute in the **Attribute** column.





**Tip:** To make a document type's fields available for auto-completion for expressions, select the document type in the **Advanced** tab's **Document Type** field.



- Specify options as described in “Setting Advanced options” on page 294.

### 10.6.2.1 Export options


“CMIS export configuration dialog box: Export options” on page 293 provides a description of CMIS export options.

**Table 10-6: CMIS export configuration dialog box: Export options**

Option	Description
<b>Attribute</b>	The CMIS attribute display name and attribute ID. The attribute ID is displayed in parentheses.
<b>Automatically Map</b>	<p>Automatically maps multiple attributes to their matching value paths (UIMData). The selected <b>Value</b> field's value path is used as a pattern for the other attributes' <b>Value</b> fields. For each attribute ID (that is, not the display name) for which there also exists a matching value name, the pattern is copied to the <b>Value</b> field and the value name is substituted. Existing attribute values are not overwritten.</p> <p>The value path could include custom values.</p> <p> <b>Note:</b> Typically, this feature is used for step/node values (for example, IndexPlus at level 0) as well as for UimData.</p>
<b>File value</b>	<p>(CMIS <b>Document</b> or its subtypes only) Selects a node level (identified by number) to be exported.</p> <p> <b>Note:</b> The choices available depend on the level at which you have chosen to export (in the command's <b>Each Level</b> field).</p>
<b>Required</b>	An asterisk (*) indicates that the attribute is required; otherwise, the attribute is optional.
<b>Type</b>	Data type of the CMIS attribute.

Option	Description
Value	<p>Specify the document field to map to the CMIS attribute. You can specify an expression to be resolved at runtime or a string. For more information about expressions, see <a href="#">“Programming Reference—Expressions”</a> on page 457.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• Format expressions are not supported.</li> <li>• If the data type of the expression’s resulting value does not match the corresponding CMIS data type, then an attempt is made to convert the value to the CMIS data type.</li> </ul> <hr/> <p> <b>Note:</b> If your profile links an intended CaptureFlow, you can use auto-completion to reference IA values in the expression.</p>


### 10.6.2.2 Setting Advanced options

 **Note:** When you update a Core Share metadata profile, the profile ID is changed. Since the metadata profile ID is changed, Core Capture export will not find the metadata profile that was referenced by the old ID of the metadata profile. You must update the Core Capture export profile(s) so as to see the new ID of the Core Share metadata profile.

[“CMIS export configuration dialog box: Advanced tab parameters”](#) on page 294 provides a description of CMIS export advanced tab parameters.

**Table 10-7: CMIS export configuration dialog box: Advanced tab parameters**

Parameter	Description
Document Type	(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.

Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	<p>Specifies the number of retries for the <b>Error Handling</b> parameter.</p>
<b>File Exists Handling</b>	<p>Specifies the error handling scenario in which a file with the same name already exists at the specified destination as follows:</p> <p> <b>Note:</b> To set case sensitivity, see the <b>Use case-sensitive search</b> parameter.</p> <ul style="list-style-type: none"> <li>• <b>Error:</b> (Default) The error is handled as specified in the <b>Error Handling</b> parameter. An error status is logged.</li> <li>• <b>Overwrite:</b> For files, the old file is deleted and replaced by the new one. For folders, the folder objects and all of their children are deleted and replaced by the new folder.</li> <li>• <b>Skip:</b> Nothing is exported. The current export command is skipped and the next command is executed.</li> <li>• <b>Revision:</b> For a folder, attributes are updated. For a document, a new version is created; however, if document versioning is not supported, then a new version is not created; instead, the current document's properties and content file (if specified) are updated.</li> </ul>

Parameter	Description
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command. This ID is used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not needed.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path</b>: Specifies the CMIS object ID of the object that was created, overwritten, versioned, or skipped. Type: String. Level: file level.</li> <li>• <b>&lt;ID&gt;_Status</b>: Specifies the export status. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: Node not selected, file not created.</li> <li>– 1: An error occurred.</li> <li>– 2: File exists, overwritten.</li> <li>– 3: File exists, skipped.</li> <li>– 4: New file created.</li> <li>– 5: Revision</li> </ul> </li> </ul> <p>The ID must be unique among the profile's commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>
<b>Use case-sensitive search</b>	<p>(Optional) For the <b>File Exists Handling</b> parameter, specifies whether to consider case sensitivity when determining if a file already exists in the repository.</p>

### 10.6.3 Defining Content Server content

The Content Server export command specifies the following:

- Parent-child relationship between items and a hierarchy in the Content Server;
- Item definitions which determine how the items should be processed.
- Specify export content as follows:
  - a. On the **Export List** tab, create a list to enable the item definitions you want to export and their hierarchy. For more information, see [“Creating an export list” on page 297](#).
  - b. On the **Item Definitions** tab, create and edit definitions for Content Server objects to specify item location, file that you want to export, categories, and

attribute values. For more information, see [“Configuring item definitions” on page 298](#).

- c. On the **Advanced** tab, specify error handling and additional options. For more information, see [“Setting Advanced options” on page 304](#).

### 10.6.3.1 Creating an export list

The **Export List** tab determines the parent-child relationship between items, defines a hierarchy in the Content Server, and enables the item definitions you want to export.

#### To create an export list:

1. Click **<New Item>**. A blank row is added into the list.
2. Within the **Name in Content Server** area, click **<Specify Value>**.
3. In the **Format String Editor** dialog, enter the name of the new item in the Content Server. Specify an expression in curly brackets to define the name dynamically in runtime. For example: `{S|CustomValues:7.Year}_{S|CustomValues:7.Month}_{S|_Batch.BatchName}`.




**Note:** If your profile links an intended CaptureFlow, autocompletion for this CaptureFlow becomes available while you type the expression. To learn more about format expressions, see [“Programming Reference—Expressions” on page 457](#).

4. Within the **Definition Name** area, select an item definition from the list defined on the **Item Definitions** tab.  
Otherwise, click **<Specify Value>** to define an expression in the **Format String Editor** to map the definition dynamically in runtime. For example: `{S|Classification:1.DocType}`.
5. Repeat steps 1 through 4 to add more items into the list.
6. To define the parent-child relationships between items, select an item in the **Export List** and use the arrow buttons. Error icon indicates that the item position in the list is invalid.

To delete an item from the list, select the item and click **Delete** (or press **DELETE**).

### 10.6.3.2 Configuring item definitions

An item definition determines how you want to process an item. The **Business Workspace**, **Folder**, **Document**, and **Workflow** item types can be defined.

 **Note:** If exporting a Business Workspace, before you begin, ensure the target system has Business Workspaces enabled.

#### To add an item definition:

1. In Designer, click **Export**, and select an export profile.
2. Log into the Content Server.
3. On the **Item Definitions** tab, click the green plus (**Add Definition**) button and select the **Business Workspace**, **Folder**, **Document**, or **Workflow** to add.
4. The new item of the selected type is added. To rename the item, select it in the list and type a new name.
5. Define the item settings depending on its type as follows:
  - For **Business Workspace**, specify the settings on the **Location**, **Categories**, and **Permissions** tabs
  - for **Document**, specify the settings on the **Location**, **Categories**, **File**, and **Permissions** tabs;
  - for **Folder**, specify the settings on the **Location**, **Categories**, and **Permissions** tabs;
  - for **Workflow**, specify the settings on the **Workflow Map** and **Attributes** tabs.
6. If exporting a Business Workspace, you must associate it with a workspace template.
7. Click **OK**.

To delete an item definition from the list, select the item definition and click the red cross (**Remove Definition**) button or press **DELETE**.

#### 10.6.3.2.1 Specifying an item location

On the **Item Definitions > Location** tab, specify the parameters outlined in the following table.

Parameter	Description
<b>Select Location</b>	<p>Select an export location:</p> <ul style="list-style-type: none"> <li>• <b>Specific Folder:</b> select this option for the item in the root position to specify a location on the Content Server. In the <b>Path to Content Server</b> field, do one of the following: <ul style="list-style-type: none"> <li>– enter a path or format expressions in curly brackets. For example: <code>EnterpriseWorkspace\General Invoice Folder, {S CustomValues:7.PathInContentServer}</code>. To learn more about format expressions, see <a href="#">“Programming Reference—Expressions” on page 457</a>.</li> <li>– click the browse button to open the <b>Select Path</b> dialog and browse to a folder on the Content Server.</li> </ul> </li> <li>• <b>Relative Position in Export List:</b> select this option to make the export location of the item relative to the export location of its parent in the export list. An item in the root position cannot have a relative path.</li> </ul>
<b>If the Document exists</b> (the option is available for Document only)	<p>Specify how the module handles documents that exist on the Content Server:</p> <ul style="list-style-type: none"> <li>• <b>Add new version:</b> creates a new version of the document while retaining previous versions.</li> <li>• <b>Replace with new item:</b> deletes all previous versions of the document and replaces it with the new one.</li> <li>• <b>Skip exported item:</b> The current document is skipped.</li> <li>• <b>Generate error:</b> Generates an error if the document exists and uses error handling settings defined on the <b>Advanced</b> tab.</li> </ul>
<b>Generate error if Folder exists</b> (the option is available for Folder only)	<p>Select to generate an error if the folder exists. Error handling settings defined on the <b>Advanced</b> tab are used.</p>
<b>Set item description</b>	<p>Select to include a description for the item exported to the Content Server. You can also specify IA Value in curly brackets to use in describing the item to export. For example: <code>Capture from {S ScanPlus.ScanOperator}</code>.</p>

### 10.6.3.2.2 Specifying categories and attributes

The **Item Definitions > Categories** tab specifies the Content Server categories and their attribute values.

A category is a classification or grouping of attributes. An attribute is a field storing information such as name and social security number that help you identify an item uniquely. For example, the accounting department can include a category labeled **Travel Voucher**. The **Travel Voucher** category of documents can include such attributes as **Last Name** and **P.O. Number**.

#### To specify a category and assign values to its attributes:

1. Click **Add**.
2. In the **Select Category** dialog, browse for the required category on the Content Server and click **OK**. The new category and its attributes are added to the **Categories** tab.



**Note:** The following attribute types are currently not supported by OpenText Content Server Export: *User:Field* and *Set*. Therefore, attributes of these types are not displayed and cannot be configured for the selected category.

3. To modify a value for an attribute, click **<Specify Value>** on the attribute you want to specify.

If an attribute supports multiple values, then you can add values as multiple rows for each attribute by clicking **<Add Row>** or **<Specify Value>**. Required attributes are pre-populated with the maximum number of rows. The attribute's maximum number of values is also displayed.

4. In the **Format String Editor** dialog, enter a value. For example: `{S|Classification:1.DocType}`.



**Note:** If your profile links an intended CaptureFlow, autocompletion for this CaptureFlow becomes available while you type the expression. To learn more about format expressions, see [“Programming Reference—Expressions” on page 457](#).

5. (Optional) Clear rows for those categories or attributes that you do not want to export.


To delete a category, select its check box and click **Delete** (or press **DELETE**).

To update the already added categories according to the latest Content Server status, click **Refresh**.

### 10.6.3.2.3 Specifying file options


The **Item Definitions > File** tab is available for the Document type of item definition only and specifies a name, value, and mime type of the content file exported to the Content Server.

The following table provides a description of the file option parameters.

Parameter	Description
<b>File Name</b>	<p>Provide a file name and extension of the content file. This name is not the stage file name of the exported file.</p> <p>You can add text and format expressions in curly brackets. For example: <code>InvoiceDoc.pdf</code>, <code>Invoice {S CustomValues:0.InvoiceNumber}.pdf</code>. To learn more about format expressions, see <a href="#">“Programming Reference—Expressions” on page 457</a>.</p>
<b>File Value</b>	<p>Enter a path to the file IA value that keeps the file to export. For example: <code>Classification:0.InputImage</code>. To learn more about format expressions, see <a href="#">“Programming Reference—Expressions” on page 457</a>.</p> <p> <b>Note:</b> If your profile links an intended CaptureFlow, the <b>File Value</b> list is populated with the file IA values available on the file level (<b>Each Level</b> parameter) of the command and lower. You can select the required value from the list.</p>
<b>Mime Type</b>	<p>OpenText uses this option when displaying the file on the Content Server.</p> <p>You can enter an expression to specify an IA Value to use in selecting the mime type of the items to export. For example: <code>{CustomValues:0.DocumentContentType}</code>.</p>

### 10.6.3.2.4 Specifying permissions

For each **Document** or **Folder**, the **Item Definitions > Permissions** tab lets you grant permissions for users and groups as outlined in the following table.

 **Note:** Permissions can be set for the newly created items only. You cannot set permissions for a folder or document that already exists on the Content Server.

Parameter	Description
<b>Login/Group Name</b>	The panel includes the following lists: <ul style="list-style-type: none"> <li>• <b>Default:</b> The list includes the following groups that correspond to the groups existing on the Content Server: <b>Owner</b>, <b>Owner Group</b>, and <b>Public</b>. By default, the permissions for these entries are empty and they inherit permissions of their parents during runtime. User can reassign permissions for the default groups and change ownership as necessary. The default entries cannot be deleted.</li> <li>• <b>Others:</b> A user can modify the list by adding or removing users and groups to/from this list and specify permissions for them as necessary.</li> </ul>
<b>Permission level</b>	The permission list order, selection logic, and names cannot be changed. The Designer permission list completely matches the Content Server permission list.

**To reassign permissions for a default group:**

1. On the **Login/Group Name** panel, select the check box for the default group for which you need to add permissions.
2. On the **Permission level** panel, select the required check boxes.

**To change ownership for Owner and Owner Group:**

1. On the **Login/Group Name** panel, next to **Owner** or **Owner Group**, click **Change**.

The **Choose Users or Groups** dialog box opens.

2. In the **Find** field, type the text that might correspond to a user login or group name as it is defined on the Content Server and click the **Find** icon.

All found entries that match the typed text are displayed and are by default sorted alphabetically by **Login/Group Name**. Note that search result displayed in this dialog box has a limited number of instances so make sure to narrow your search criteria.

3. In the search list, select the user or group that you want to add and click **Add**.  
The selected user or group is displayed in the next to **Owner** or **Owner Group** on the **Login/Group Name** panel.

**To add a group/user and define permissions for them:**

1. On the **Login/Group Name** panel, click **Add** (or press CTRL+N).  
The **Choose Users or Groups** dialog box opens.
2. In the **Find** field, type the text that might correspond to a user login or group name as it is defined on the Content Server and click the **Find** icon.  
All found entries that match the typed text are displayed and are by default sorted alphabetically by **Login/Group Name**. Note that search result displayed in this dialog box has a limited number of instances so make sure to narrow your search criteria.
3. In the search list, select the entry/entries that you want to add and click **Add**.  
All selected users and groups appear in the **Others** list on the **Login/Group Name** panel.
4. To define permissions, select a group/user check box and select the required permission check boxes for them.

**To delete a group/user:**



**Note:** Only users/groups from the **Others** list can be deleted.

1. On the **Login/Group Name** panel, select the check box for the group/user which you need to remove.
2. Click **Delete**.

### 10.6.3.2.5 Specifying a Workflow Map and Attributes


The **Item Definitions > Workflow Map** and **Attributes** tabs specify a Content Server workflow map and its attribute values.

A workflow map defines a business process to be executed on the Content Server. For example, a business process for expense reports could route an expense report to a supervisor for approval and then to the company's accounting department for payment. A workflow map attribute is a field that is used to store and track information throughout the life time of a business process.

**To specify a workflow map and assign values to its attributes:**

1. In the **Select workflow map for this workflow** field, click ... (the browse button) and select a workflow map.

The **Attributes** tab is populated with the selected workflow map's attributes.

 **Note:** To update the workflow map to the most recent version in Content Server, click **Refresh**.


2. (Optional) Add a description of the workflow map by clicking **Set/Modify comments** and entering the appropriate text.
3. On the **Attributes** tab, modify a value for an attribute by clicking **<Specify Value>** in the row you want to specify.

If an attribute supports multiple values, then you can add values as multiple rows for each attribute by clicking **<Add Row>** or **<Specify Value>**. Required attributes are pre-populated with the maximum number of rows. The attribute's maximum number of values is also displayed.

4. In the **Format String Editor** dialog box, enter a value.

You can also perform the following actions:

- To clear attributes that you do not want to export, clear the attribute's check box.
- To delete a value, select the row and click **Delete** (or press **DELETE**).
- To update the attributes to the most recent versions in Content Server, click **Refresh**.

 **Note:** If your profile links an intended CaptureFlow, auto-completion for this CaptureFlow becomes available while you type the expression. To learn more about format expressions, see *“Programming Reference—Expressions”* on page 457.

### 10.6.3.3 Setting Advanced options

On the **Advanced** tab, specify the options outlined in the following table.

**Table 10-8: Content Server export configuration dialog box: Advanced tab parameters**


Parameter	Description
Document Type	(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.

Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged. The administrator can manually restart the task.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	<p>Specifies the number of retries for the <b>Error Handling</b> parameter.</p>
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not needed.</p> <p>Dynamic output values are the following:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path:</b> Contains the path of the item on the Content Server, specified in the item definition. <ul style="list-style-type: none"> <li>– If there are several items in the export list, the path of the last item is taken.</li> <li>– If the item exists, the rule defined in this item definition is applied.</li> </ul> </li> <li>• <b>&lt;ID&gt;_Status:</b> Specifies the export status of the item. If there are several items in the export list, a status of the last item is taken.</li> </ul> <p>The ID must be unique among the profile's commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>

## 10.6.4 Defining CSV content

Define CSV content on the tabs outlined in the following table.

**Table 10-9: CSV export configuration: Content tab parameters**

Parameter	Description
Column Separator	Expand the list and select the column delimiter. Values: Comma, Tab, Pipe (' ').
Create Row For Each Level	Expand the list and specify the level at which every filtered node will add a separate row of data to the CSV file.  This level must be equal or lower than the file level of the export command specified in the <b>Each Level</b> command parameter.
Column Name	Click the <New Column> placeholder. Specify the column name (plain text, not a format expression) in the edit box.  Repeat for each new column in the CSV file.   <b>Note:</b> To move a column definition up or down in the ordered list of columns, use <b>Move Up</b> and <b>Move Down</b> . To remove a column definition, use <b>Delete</b> .
Column Value	Click the <Specify Value> placeholder and specify the column value in the Format String Editor. You can specify a regular string or a format expression that will be resolved and inserted into the CSV file at runtime. To insert a function or an IA value in a format expression, use curly brackets. For each file-level node that is handled during the command execution, the expressions are resolved using the data stored in that node and in its child nodes. To learn more about syntax and capabilities of format expressions, see <a href="#">“Programming Reference—Expressions” on page 457</a> .

Define CSV content on the tabs outlined in the following table.

**Table 10-10: CSV export configuration dialog box: Advanced tab parameters**

Parameter	Description
<b>File Encoding</b>	Specify the encoding for the generated file.  <b>System</b> is the default code page of the operating system running the triggered Standard Export module.
<b>File Exists Handling</b>	Expand the list and select the error handling scenario in case a file with the same name already exists at the specified destination. See the following options:  <ul style="list-style-type: none"> <li>• <b>Error:</b> Set by default. The task gets an error status. The error is handled as specified in the <b>Error Handling</b> parameter.</li> <li>• <b>Overwrite:</b> The old file is replaced by the new one.</li> <li>• <b>Rename:</b> The new file is saved with a different name. The new file name contains the initial file name appended with 3 to 20 digits to ensure a unique name. The old file is not affected.</li> <li>• <b>Skip:</b> The new file is not saved. The current node is skipped and the next one is taken.</li> <li>• <b>Append:</b> Appends the new columns to the end of the existing file.</li> </ul>
<b>Write Header</b>	Check the box to include the column names to the CSV file at the first line. Default: checked.

Parameter	Description
Identifier	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not needed.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path:</b> Specifies the full path of the exported file. Type: String. Level: file level.</li> <li>• <b>&lt;ID&gt;_Status:</b> Specifies the export status. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: node not selected, file not created.</li> <li>– 1: error creating the file, skipped, or: file exists, overwritten with error.</li> <li>– 2: file exists, overwritten.</li> <li>– 3: file exists, skipped.</li> <li>– 4: new file created.</li> </ul> </li> </ul> <p>The ID must be unique among the profile's commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>

## 10.6.5 Defining xECM content

You define the xECM content to export by specifying the following properties. For each export command, specify one xECM document type.



**Note:** If a CaptureFlow is specified in **Intended CaptureFlow**, then the **Document Definition's File Value** list is populated with the file IA values available on the file level (**Each Level** parameter) of the command and lower.

See the properties outlined in the following table.


**Table 10-11: xECM export configuration dialog box: Workspace tab**


Property	Description
Business Application	An existing xECM business application to which to export.
Business Object Type	The business application's object type to which to export.

Property	Description
Properties to match a required workspace at runtime	<p>Specifies the mappings between an xECM business object's properties (in the <b>Business Object Property</b> column) and expressions (in the <b>Value</b> column) to enable matching the specified xECM business object at runtime. The properties must uniquely identify the business object; otherwise, the export is not executed. Furthermore, if a business object property is empty or consists of spaces only, then an error occurs.</p> <p>To learn more about expressions, see <a href="#">"Programming Reference—Expressions"</a> on page 457.</p>

Define xECM content on the tabs outlined in ["xECM export configuration dialog box: Document Definition tab"](#) on page 309.

**Table 10-12: xECM export configuration dialog box: Document Definition tab**

Property	Description
<b>Business Properties</b>	
xECM Document Type	An xECM document type
xECM business properties	<p>Specify a format string for each business property.</p> <p>To exclude a property, clear its check box.</p> <p> <b>Note:</b> The following category attribute types are not supported. If business properties with any of the unsupported category attribute types are added, an error occurs on export:</p> <ul style="list-style-type: none"> <li>• Text:Table Key Lookup</li> <li>• User:Field</li> <li>• Text:Reference</li> <li>• Set</li> <li>• Partner Set</li> </ul>
<b>Required</b>	* indicates that the property is required; otherwise, the property is optional.
<b>File</b>	

Property	Description
<p><b>File Name</b></p>	<p>Provide a file name and extension of the content file. This name is not the stage file name of the exported file.</p> <p>You can add text and format expressions in curly brackets. To learn more about format expressions, see <i>“Programming Reference—Expressions”</i> on page 457.</p>
<p><b>File Value</b></p>	<p>Selects the files to be exported.</p> <ul style="list-style-type: none"> <li>• <b>Page Image</b> Each page is exported as a separate file.</li> <li>• <b>Document PDF File</b> All pages in a single document are exported as a single PDF file.</li> <li>• <b>Batch PDF File</b> All pages in all documents in the batch are exported as a single PDF file.</li> </ul> <p> <b>Note:</b> The choices available depend on the level at which you have chosen to export (in the command’s <b>Each Level</b> field); for example, if you choose <b>Page</b> in <b>Each Level</b>, then only <b>Page Image</b> is available for <b>File value</b>.</p>
<p><b>Mime Type</b></p>	<p>This option is used when displaying the file in the xECM business application.</p> <p>You can use an expression to select the mime type of the items to export.</p>
<p><b>Action on detection of existing file</b></p>	<p>From the drop-down , select an action for an existing file with the same file name.</p> <ul style="list-style-type: none"> <li>• <b>Put the batch into an error state</b></li> <li>• <b>Create a copy</b></li> <li>• <b>Create a version</b></li> </ul>

On the **Advanced** tab, specify the options outlined in the *“xECM export configuration dialog box: Advanced tab”* on page 311.

**Table 10-13: xECM export configuration dialog box: Advanced tab**

Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	Specifies the number of retries for the <b>Error Handling</b> parameter.
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not needed. Dynamic output values are the following:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path:</b> Contains the path of the file on the xECM business application.</li> <li>• <b>&lt;ID&gt;_Status:</b> Specifies the export status of the file.</li> </ul> <p>The ID must be unique among the profile's commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>
<b>Document Type</b>	(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.

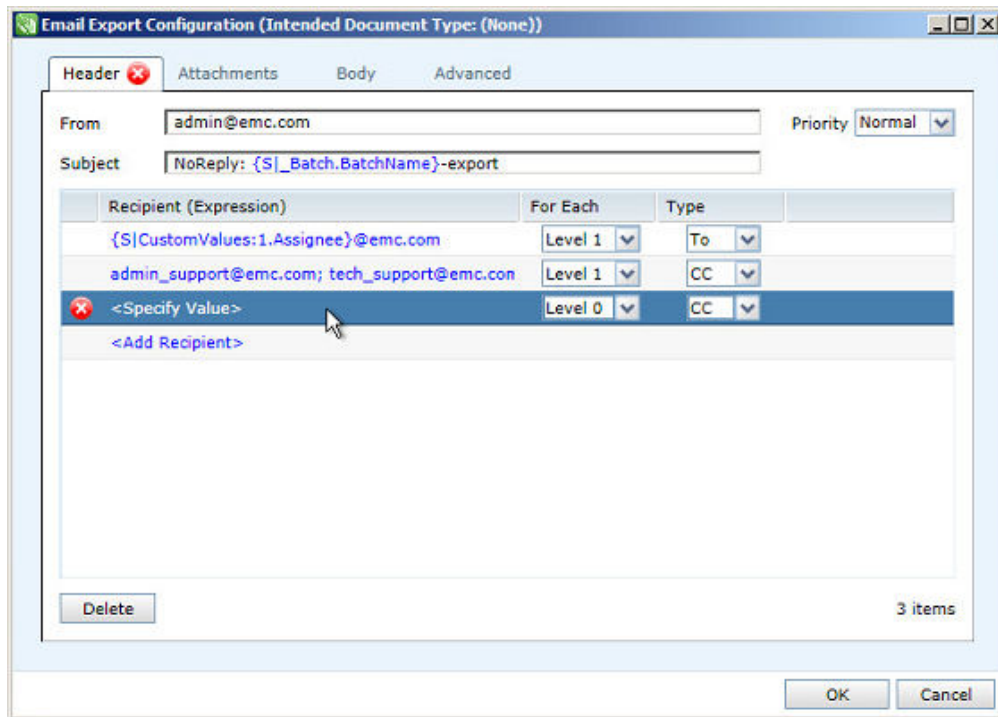
## 10.6.6 Defining email content

An email export command defines an email message. For each file-level node that is handled during the command execution, expressions are resolved using the data stored in that node and in its child nodes. Define the email message on the following tabs:

- **Header:** Defines the email header.
- **Attachments (optional):** Defines the attachments.
- **Body:** Defines the structure and contents of the email body.
- **Advanced:** Defines advanced email generation parameters.

### 10.6.6.1 Defining the email header

On the **Header** tab of the **Email Export Configuration** box, specify the header parameters as follows.



**Figure 10-2: Configuring email export**

#### To define email header parameters:

1. Specify the email address of the sender in the **From** field. This parameter is required. You can add text and format expressions in curly brackets. To learn


more about syntax and capabilities of format expressions, see [Programming Reference](#).



**Note:** The format of the email address is not validated at design time. If the entered address is invalid, an error is generated at runtime only.

2. Specify the mail subject in the **Subject** field. You can add text and format expressions in curly brackets.
3. Expand the **Priority** list and select the priority of the mail message: **High**, **Normal** (default), or **Low**.
4. Define the list of mail recipients as follows:
  - a. Click **<Add Recipient>** to add a new line to the list of recipients.
  - b. Specify the parameters of a new recipient as described in the following table.

**Table 10-14: Email export configuration dialog box: Header tab parameters**

Parameter	Description
Recipient (Expression)	<p>Click <b>&lt;Specify Value&gt;</b> and enter the email address of the recipient. You can add text and format expressions in curly brackets. You can also enter a list of addresses delimited by a semicolon. Each address in this list will be added as a separate recipient to the email header.</p> <p> <b>Note:</b> The format of the entered email address(es) is not validated at design time. If an invalid address is entered, an error is generated at runtime only.</p>
For Each	<p>(Optional) If the <b>Recipient</b> parameter is includes IA values, expand the list and select the <i>recipient level</i> to specify how multiple resolved email addresses will be iterated.</p> <p>If this level is <i>equal</i> to the <i>file level</i> of the command, the email message is sent to <i>all</i> resolved email addresses. If the recipient's level is lower than the file level, the email message is sent to the first resolved email address. For each file-level node that is handled during the command execution, the expressions are resolved using the data stored in that node and in its child nodes.</p> <p>Values: 0 through 7. Default: 0.</p>

Parameter	Description
Type	Expand the list and specify the type of recipient. Options: <b>To</b> , <b>Cc</b> , <b>Bcc</b> . Default: <b>To</b> .

- c. Repeat the above steps to add more recipients.



**Note:** The email header requires at least one recipient of type **To**. If all recipients are **Cc** and **Bcc** types, or if no recipients are defined, a design error is detected.

- d. To delete a recipient, select it from the **Recipient (Expression)** list and press **Delete**.

### Related Topics

[“Programming Reference—Expressions” on page 457](#)

### 10.6.6.2 Defining attachments


On the **Attachments** tab of the **Email Export Configuration** box, define the list of attachments as follows.

#### To define email attachments:

1. Click the **<Add Attachment>** placeholder to add a new line to the list of attachments.
2. Specify the parameters as described in the following table.

**Table 10-15: Email export configuration dialog box: Attachments tab parameters**

Parameter	Description
Display (Expression)	Click <b>&lt;Specify Value&gt;</b> and enter the name of the attached file in the <b>Format String Editor</b> dialog box. You can add text and format expressions in curly brackets.  To learn more about syntax and capabilities of format expressions, see <a href="#">Programming Reference</a> .

Parameter	Description
<b>File (IA Value)</b>	<p>Enter the path to the File IA value that keeps the file to be attached.</p> <p> <b>Note:</b> If your profile links an intended CaptureFlow, the <b>File (IA Value)</b> list is populated with File IA values available on the file level (<b>Each Level</b> parameter) of the command and lower. You can select the required value from the list.</p>
<b>For Each</b>	<p>Expand the list and select the level to specify how multiple attachments resolved from <b>File (IA Value)</b> will be iterated.</p> <p>If this level is equal to the file level, the email is sent with all of the resolved attachments. If this level is lower than the file level, the email is sent with the first resolved attachment.</p> <p>Values: 0 through 7. Default: 0.</p>

- Repeat the above steps for each new attachment.
- To delete an attachment from the list, select the line and press the **Delete** button.

## Related Topics

[“Programming Reference—Expressions” on page 457](#)

### 10.6.6.3 Defining the email body

On the **Body** tab of the **Email Export Configuration** box, you need to define the content of the email message. The common rules for creating the content definition are described in [“Using the embedded content editor” on page 281](#).

The content editor displayed on the **Body** tab allows you to create definitions of HTML-formatted and text-formatted email messages. To choose the type of message, expand the **Type** combo box and select the required option:

- Text:** If you have selected the text format for your email messages, creating the content definition is similar to creating the text file content described in [“Defining text file structure” on page 322](#).
- HTML:** If you have chosen the HTML type, use HTML markup to format your message structure. The content between tags may include plain text and format expressions in curly braces. To learn about the expressions syntax and capabilities, see [Programming Reference](#) section.

## Related Topics

“Setting advanced options” on page 316

“Programming Reference—Expressions” on page 457

### 10.6.6.4 Setting advanced options

On the **Advanced** tab of the **Email Export Configuration** box, specify the parameters outlined in the following table.

**Table 10-16: Email export configuration dialog box: Advanced tab parameters**

Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	Specifies the number of retries for the <b>Error Handling</b> parameter.

Parameter	Description
Identifier	<p>(Optional) Specify the ID to be used as a prefix in the names of dynamic output values generated by the task. Leave this parameter empty if those dynamic values are not needed.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path</b>: not applicable to email export. Always contains an empty string.</li> <li>• <b>&lt;ID&gt;_Status</b>: specifies the status of export. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: node not selected, email not created.</li> <li>– 1: error creating the email, skipped.</li> <li>– 2: not applicable.</li> <li>– 3: not applicable.</li> <li>– 4: new email created.</li> </ul> </li> </ul> <p>The ID must be unique among other profile commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>
Document Type	<p>(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.</p>

## 10.6.7 Defining file export

Specify the file to export by selecting it in **From** and then specify the properties on the **Advanced** tab as outlined in the following table.



**Note:** You must select a CaptureFlow in **Intended CaptureFlow**.

**Table 10-17: File export configuration: Advanced tab parameters**




Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	<p>Specifies the number of retries for the <b>Error Handling</b> parameter.</p>
<b>File Exists Handling</b>	<p>Expand the list and select the error handling scenario in case a file with the same name already exists at the specified destination. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Error:</b> Set by default. The task gets an error status. The error is handled as specified in the <b>Error Handling</b> parameter.</li> <li>• <b>Overwrite:</b> The old file is replaced by the new one.</li> <li>• <b>Rename:</b> The new file is saved with a different name. The new file name contains the initial file name appended with 3 to 20 digits to ensure a unique name. The old file is not affected.</li> <li>• <b>Skip:</b> The new file is not saved. The current node is skipped and the next one is taken.</li> </ul>


Parameter	Description
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command that will be used in the dynamic output values generated by the task. Leave this field empty if those values are not used.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path</b>: Specifies the full path of the exported file. Type: String. Level: file level.</li> <li>• <b>&lt;ID&gt;_Status</b>: Specifies the export status. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: node not selected, file not created.</li> <li>– 1: error creating the file, skipped, or: file exists, overwritten with error.</li> <li>– 2: file exists, overwritten.</li> <li>– 3: file exists, skipped.</li> <li>– 4: new file created.</li> </ul> </li> </ul> <p>The ID must be unique among other profile commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>
<b>Document Type</b>	<p>(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.</p>

### 10.6.8 Defining ODBC content

ODBC export executes SQL statements for the selected ODBC connection. You specify the properties on the **ODBC Export Configuration** and **Advanced** tabs as outlined in the following table.

**Table 10-18: ODBC export Configuration tab parameters and actions**

Parameter/Action	Description
<b>Transaction behavior</b>	Select one of the following: <ul style="list-style-type: none"> <li>• <b>Unsupported:</b> Does not support transactions.</li> <li>• <b>Commit after each statement execution:</b> Commits transaction after each statement has been executed (if supported by the ODBC driver).</li> <li>• <b>Commit after all statements execution:</b> Commits transaction after all statements have been executed (if supported by the ODBC driver) for each batch node.</li> </ul>
<b>Add...</b>	Add the SQL statements to the list of queries to be available for execution. Only statements from the selected ODBC connection can be selected. You can add the same statement multiple times. <p>To include a statement in execution, check it.</p> <p>To exclude a statement from execution, uncheck it. Excluded statements are not validated.</p> <p> <b>Note:</b> INSERT, UPDATE and FETCH queries are supported. For more information on defining queries, see “Defining queries” on page 68.</p>
	Remove the selected SQL statement from the list of statements available for execution.
	Specify the sequence in which the SQL statements are executed. The statements are executed starting with top and ending with the bottom.
<b>Hide/Show statement</b>	Display or hide the text of the selected SQL statement.

Parameter/Action	Description
SQL statement parameters table	<p>For each SQL statement parameter, specify its value by clicking in the <b>Value Expression</b> column. You can specify a regular string or a format expression that will be resolved and inserted into the statement at runtime. To insert a function or an IA value in a format expression, use curly brackets. To learn more about syntax and capabilities of format expressions, see <i>“Programming Reference—Expressions”</i> on page 457.</p> <p>For DbNULL to be written to the table for the following, check <b>Empty Is NULL</b>:</p> <ul style="list-style-type: none"> <li>• Empty String IA value</li> <li>• File IA value with a length of 0</li> </ul> <p> <b>Note:</b> If your profile links an intended CaptureFlow, you can use autocompletion to reference IA values in the format expression.</p> <p>If the given command links a document type in its <b>advanced options</b>, you can use autocompletion to reference fields of that document type in the format expression as well.</p>
<p><b>If 0 rows changed</b> (Insert or Update)</p> <p><b>If 0 rows found</b> (Select).</p>	<p>To specify the behavior of a statement when it does not update the table, select one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Skip and Continue:</b> Continue task processing and execute the next statement.</li> <li>• <b>Treat as error:</b> Return an error and stop the export for the current task.</li> <li>• <b>Execute next statement:</b> (UPDATE only) Continue task processing and execute the next statement (which could be any SQL statement). A common situation that you can use this option to solve is when the row to be updated does not exist and you want it to be inserted by the next statement.</li> </ul>
<b>Error if more than one row found</b>	Specifies that an error is returned if more than one row is found for the selected statement.

On the **Advanced** tab, specify the parameters outlined in the following table.

**Table 10-19: ODBC export configuration: Advanced tab parameters**

Parameter	Description
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not needed.</p> <p>The ID must be unique among the profile's commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>

### 10.6.8.1 Related Topics

[“Using autocompletion” on page 279](#)

[“CSV export configuration dialog box: Advanced tab parameters” on page 307](#)

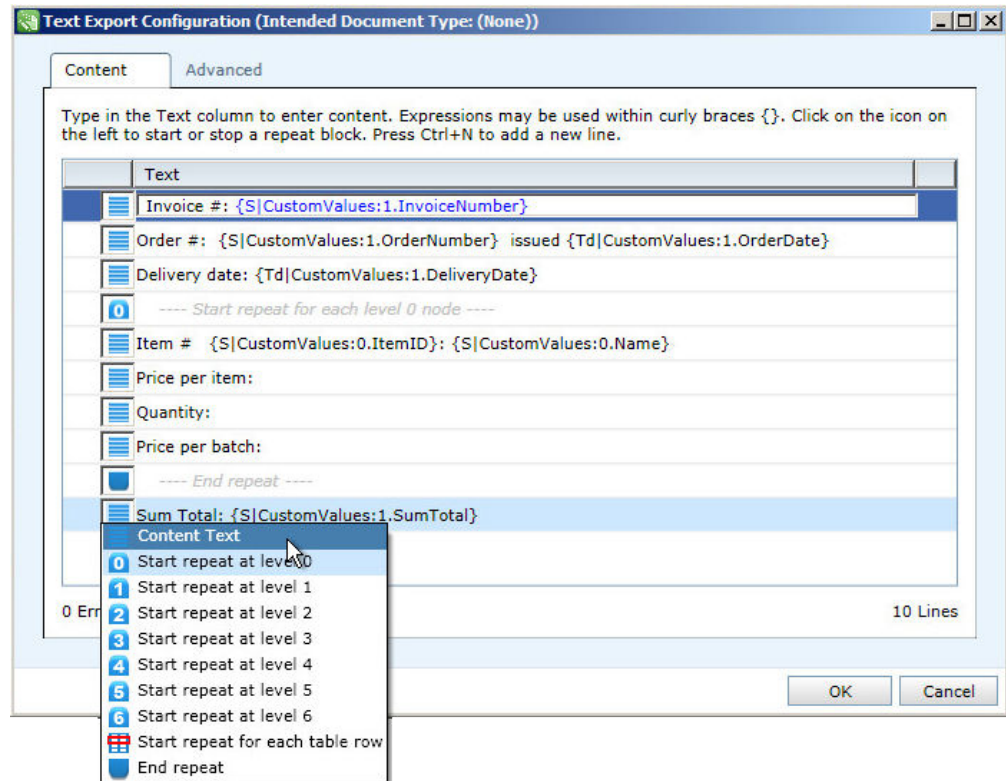
## 10.6.9 Defining text content

Define the text file structure on the **Content** tab and the file generation settings, error handling, and other advanced settings on the **Advanced** tab.

### 10.6.9.1 Defining text file structure

On the **Content** tab of the **Text Export Configuration** box, you define the content of the text file.

You can enter the text file structure line by line, or you can copy the existing file into the content editor and edit its structure. To reuse a text file in the content editor, open it in Microsoft Notepad or any other text viewer and copy the file's content to the Clipboard. Then drop the cursor in the text area of the content line and paste the Clipboard content. The copied content is split into lines automatically.



**Figure 10-3: Text export configuration**

The common directions for defining text file content are as follows:

- Use **Content Text** lines to define text that will be added to the output file. The text lines may IA values, format expressions, and document type fields.  
For more information about format expressions, their syntax, capabilities, and formatting rules, see *“Programming Reference—Expressions”* on page 457.
- If the inserted IA value or expression references a node below the file level (**Each Level** parameter of the command), the first found node will be handled and other nodes ignored. For each file-level node that is handled during the command execution, the expressions are resolved using the data stored in that node and in its child nodes. If you need to iterate all nodes on that level, or all rows in the specified document type table, include a *repetition block*. The content inside the repetition block will be added to the output XML file for each iterated node or table row.
- If you use nested repetition blocks, the level of the internal repetition block must be lower than the level of the wrapper block.

### 10.6.9.2 Setting advanced options

On the **Advanced** tab, specify the parameters outlined in “Text export configuration box: Advanced tab parameters” on page 324.

**Table 10-20: Text export configuration box: Advanced tab parameters**

Parameter	Description
<b>Error Handling</b>	<p>Specifies error handling for file generation errors as follows:</p> <ul style="list-style-type: none"> <li>• <b>Abort:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip:</b> After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	Specifies the number of retries for the <b>Error Handling</b> parameter.
<b>File Encoding</b>	<p>Specify the encoding for the generated file.</p> <p><b>System:</b> The default code page of the operating system running the Standard Export module.</p>
<b>File Length Limit</b>	<p>(Optional) Specify the maximum line length if it exceeds 20 characters. A new line will be automatically inserted every time you exceed this value when typing in the 'content text' line.</p> <p>If this value is less than 20 characters, line breaks are not inserted automatically.</p>

Parameter	Description
<b>Identifier</b>	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if those values are not used.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path</b>: Specifies the full path of the exported file. Type: String. Level: file level.</li> <li>• <b>&lt;ID&gt;_Status</b>: Specifies the status of export. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: node not selected, file not created.</li> <li>– 1: error creating the file, skipped, or: file exists, overwritten with error.</li> <li>– 2: file exists, overwritten.</li> <li>– 3: file exists, skipped.</li> <li>– 4: new file created.</li> </ul> </li> </ul> <p>The ID must be unique among other profile commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>
<b>Document Type</b>	<p>(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.</p>

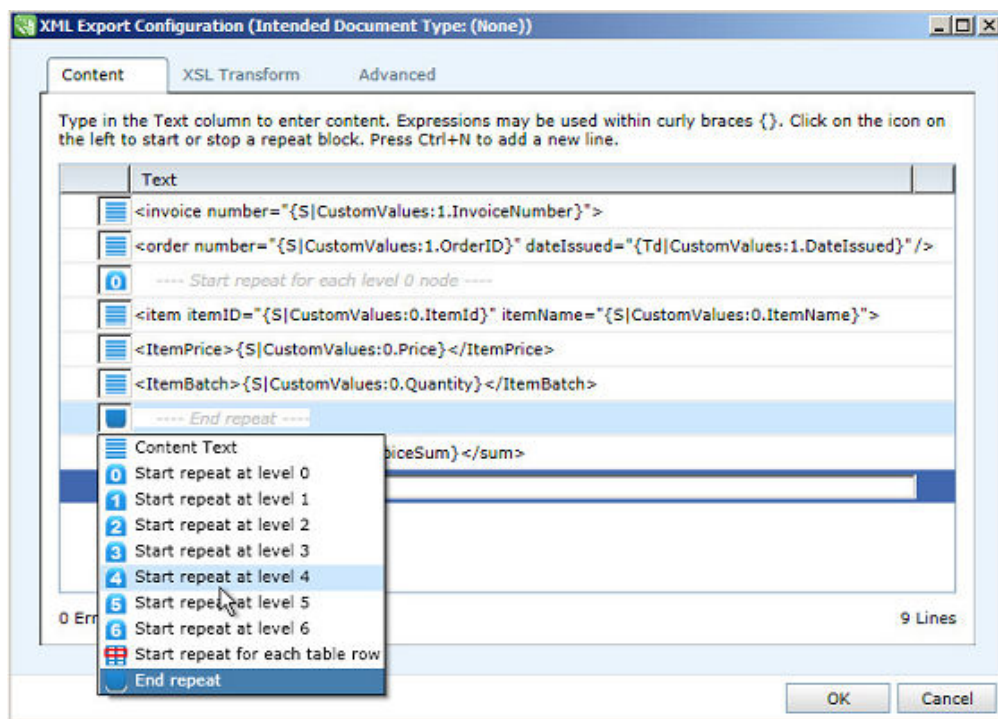
### 10.6.10 Defining XML content

1. To define the XML file structure, in the **XML Export Configuration** dialog box, click the **Content** tab. See [“Defining the XML file structure” on page 326](#).
2. (Optional) To specify the XSL transformation rules for your XML export, click the **XSL Transform** tab. See [“Defining XSL transformation rules” on page 327](#).
3. To define the XML file generation settings, click the **Advanced** tab. See [“Setting advanced options” on page 327](#).

### 10.6.10.1 Defining the XML file structure

On the **Content** tab of the **XML Export Configuration** box, specify the XML content to be generated by your export command.

You can enter the XML file structure line by line, or you can copy the existing XML file into the content editor and edit its structure. To reuse an XML file in the content editor, open it in Microsoft Notepad or any other text viewer and copy the file's content to the Clipboard. Then drop the cursor in the text area of the content line and paste the Clipboard content. The copied XML content is split into lines automatically.



**Figure 10-4: XML export configuration**

The common directions for defining XML file content are as follows:

- Use **Content Text** lines to define XML tags that will be added to the output file. Values between XML tags and XML attribute values may include IA values, format expressions, and document type fields. To learn more about format expressions, their syntax, capabilities, and formatting rules, see [“Programming Reference—Expressions” on page 457](#).
- If the inserted IA value or expression references a node below the file level (**Each Level** parameter of the command), the first found node will be handled and other nodes ignored. For each file-level node that is handled during the command execution, the expressions are resolved using the data stored in that

node and in its child nodes. If you need to iterate all nodes on that level, or all rows in the specified document type table, include a *repetition block*. The content inside the repetition block will be added to the output XML file for each iterated node or table row.

- If you use nested repetition blocks, the level of the internal repetition block must be lower than the level of the wrapper block.
- The generated XML file will not contain anything above the content that you define in the content lines. If you need an XML schema or XML file parameters be included in the generated XML file, add that data specially.

### 10.6.10.2 Defining XSL transformation rules

On the **XSL Transform** tab, you can specify XSL instructions that will be applied to your export XML file.

XSL transformation is an option. You may leave this tab empty as well. To define XSL transformation, you can use standard XSL syntax. Note that only XSLT 1.0 is supported.

```
<xsl:template match="/"> <!--
set of instructions --> <xsl:template>
```

You do not need to put any reference to your XSL instructions in the **Content** tab. If the **XSL Transform** tab is not empty, the exporter will do the binding automatically.

### 10.6.10.3 Setting advanced options

On the **Advanced** tab of the **XML Export Configuration** box, specify the following parameters.

**Table 10-21: XML export configuration dialog box: Advanced tab parameters**

Parameter	Description
<b>Error Handling</b>	Specifies error handling for file generation errors as follows: <ul style="list-style-type: none"> <li>• <b>Abort</b>: After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, file generation is canceled and an error status is logged.</li> <li>• <b>Skip</b>: After the number of retries, which is specified in the <b>Error Retry Count</b> parameter, occurs, the current node is skipped and execution proceeds to the next one. No error status is logged.</li> </ul>
<b>Error Retry Count</b>	Specifies the number of retries for the <b>Error Handling</b> parameter.

Parameter	Description
<p><b>Identifier</b></p>	<p>(Optional) Specify the ID of the export command. This ID will be used in the dynamic output values generated by the task. Leave this field empty if dynamic output values are not used.</p> <p>Dynamic output values are as follows:</p> <ul style="list-style-type: none"> <li>• <b>&lt;ID&gt;_Path:</b> Specifies the full path of the exported file. Type: String. Level: file level.</li> <li>• <b>&lt;ID&gt;_Status:</b> Specifies the status of export. Type: String. Level: file level. Values: <ul style="list-style-type: none"> <li>– Empty string: node not selected, file not created.</li> <li>– 1: error creating the file, skipped, or: file exists, overwritten with error.</li> <li>– 2: file exists, overwritten.</li> <li>– 3: file exists, skipped.</li> <li>– 4: new file created.</li> </ul> </li> </ul> <p>The ID must be unique among other profile commands. It must start with an ASCII alphabetic character and contain ASCII alphanumeric characters. The maximum allowed length for the output value name is 64 characters.</p>
<p><b>Document Type</b></p>	<p>(Optional) Select the document type for its fields to be available in auto-completion for expressions. Only document types compiled without errors are available.</p>

## Chapter 11

# Designing a document classification or data extraction project

## 11.1 Information extraction projects

The information extraction project contains profiles that define classification and extraction. For each profile, machine-learning is used to learn to recognize classes of documents (such as invoices)—and thus the correct data to extract—from the documents as they are processed. Because machine-learning begins with the first document, data extraction might not be performed on the first document. To enable machine-learning, set the **Distributed Capture** profile's **Use auto-learning with** property. Once classification and extraction is accurate enough, then you can turn off auto-learning to improve performance.

 **Note:** When auto-learning is part of an Intelligent Capture module, the setting is part of the configuration of the Collector step.


Profiles can be based on out-of-the-box business scenarios which contain pre-configured fields for extracted data; for example, invoices with fields for an invoice number and the total amount charged. You can add additional custom fields. You can also add a profile that does not have any pre-configured fields.

### 11.1.1 High-level procedure—Information extraction projects

1. Create a project.
2. On the **Recognition Projects** tab, open the **Information Extraction** project and add profiles. All scenarios, except for **Custom**, have pre-configured fields. You might also add custom fields and specify each profile's preferred cultures.

Profiles correspond to document types and profile fields correspond to document type fields. A document type with the same name and fields is automatically created for each profile.

3. In **Document Types**, configure the corresponding document type.  
For more information, see [“Designing a document type” on page 341](#).

 **Note:** Some document type field properties are read-only for information extraction profiles.



#### Caution

If you downloaded the Information Extraction project from the server without also downloading an associated document type and that document

type does not exist on your machine, then a new default document type—without any of your field property, rules, and form customizations—is automatically generated. If this situation occurs and the original document type still resides on the server or on another machine, then you should publish it to the server (if necessary) and download it to your local machine.

## 11.1.2 Creating information extraction profiles

### To create an information extraction profile:

1. In the information extraction project, click **Add Profile**, specify a name, and select a scenario in **Scenario for the default field set**.



**Note:** To delete a profile, select the profile and click **Remove Profile**.

The scenarios are as follows:



**Note:** For information only, the pre-configured fields are displayed as read-only on the **Scenario Fields** tab.

- **Custom:** Empty profile without any pre-configured fields.
  - **Invoice:** Standard invoice.
  - **Sales order:** Standard sales order.
  - **Delivery note:** Goods receipt issued upon a receipt of a shipment.
  - **Order confirmation:** Confirmation of the sales order, including the delivery details.
  - **Remittance advice:** Payment details.
  - **Incoming quotation:** Request for quote (RFQ) details.
2. From the Classification culture list, select the language that will be used to classify the documents.
  3. To define custom fields, click **Fields > Custom fields > Add Field**.


Information extraction profile fields are mapped to document type field properties as outlined in the following table.



**Note:** To delete a field, select the field and click **Remove Field** (or right-click the field and click **Remove**).


To rename a field, right-click the field and click **Rename**.

Field type	Document type		Description
	Data type	Validation type	
Amount	Number	Amount	An amount to two decimal places.
Barcode	String/Text	Not applicable	A barcode text field. Information extraction will learn the barcode types to identify at extraction time.
Business Entity	String/Text	Not applicable	Name of a company or other organization.
Currency	String/Text	Not applicable	Currency name or symbol.
Date	Date	Not applicable	A date.
Decimal	Number	Decimal	A decimal number.
Sentiment	String	Not applicable	<p>The tone of the document: positive, neutral, or negative.</p> <p>If a file of the type <b>Sentiment</b> is used in an information extraction profile, place the config file for Magellan under <code>..\binnt\IEE2\Config</code>. Name the config file <code>MagellanConfig.xml</code>. The xml file must contain the following content:</p> <pre>&lt;MagellanConfig&gt; &lt;url&gt;&lt;/url&gt; &lt;username&gt;&lt;/username&gt; &lt;password&gt;&lt;/password&gt; &lt;/MagellanConfig&gt;</pre>

Field type	Document type		Description
	Data type	Validation type	
String	String/Text	Not applicable	A string.   <b>Note:</b> If a document type field accepts multiple lines (for example, an address block), then it is not machine-learned. An alternative would be a one-to-one correspondence between each line and a field.
String List	String/Text	Not applicable	A comma-delimited list of strings.  For the document type, the strings are displayed in a multiline <b>Text Box</b> with one line per string.
Table	Table	Not applicable	A table.

- To define columns for tables, select the table, click **Add Column**, and specify the name and column type.





Information extraction profile fields are mapped to document type field properties as outlined in the following table.

 **Note:** To delete a column, select the column and click **Remove Field** (or right-click the column and click **Remove**).

To rename a field, right-click the field and click **Rename**.

Column type	Document type		Description
	Data type	Validation type	
Amount	Number	Amount	An amount to two decimal places.
Date	Date	Not applicable	A date.

Column type	Document type		Description
	Data type	Validation type	
Decimal	Number	Decimal	A decimal number.
Description	String/Text	Not applicable	A custom description.
String	String/Text	Not applicable	A string.
Item Qualifier	String/Text	Not applicable	The type of item specified for a row; for example, "Item delivered" or "Additional transportation fee".
Unit	String/Text	Not applicable	The name or abbreviation of the unit of measurement.

5. To specify the cultures to apply to a profile's incoming documents, in the **Preferred Cultures** list, perform the following actions:
  - Add a culture to the list by clicking **Add**  and selecting a culture.
  - Cultures are applied to incoming documents in ascending order. Order the cultures by selecting the culture and clicking **Move up**  or **Move down** .
  - Delete a culture from the list by selecting the culture and clicking **Delete** .
6. Click **Save**.

### 11.1.3 Converting Advanced Recognition document types to Information Extraction document types

New in 21.4, users can convert their Advanced Recognition document types to Information Extraction document types. This saves you the effort of recreating Information Extraction document types if you already have Advanced Recognition document types set up in Designer.

1. In Intelligent Capture Designer, ensure that you first create a recognition project for Information Extraction:
  1. From the left pane, click **Recognition**.
  2. Click **New**.
  3. In the New Recognition Project Name field, type the name for your Information Extraction project.
  4. From the Project Type list, select **Information Extraction**.

2. From the left pane, click **Document Types**.
3. Select your Advanced Recognition Document Type and click **Copy to Other Project**.
4. In the Save As dialog, do the following:
  1. Select either **Prefix** or **Suffix**.
  2. Depending on the choice you made, using a prefix or suffix, type a name for the new Document Type. For example, IE Invoices.
  3. From the Recognition Project list, select the AR project.
  4. Click **Save**.



**Tip:** You can also convert Information Extraction document types to Advanced Recognition using the same procedure.

## 11.2 Recognition projects

This section introduces recognition projects. It describes the purpose and use of recognition projects and discusses how the projects interact with the advanced recognition modules. It also provides high-level steps for creating recognition projects.



To fill an empty recognition project with content, click **Recognition** to open the Recognition Designer tool. For more information, see *OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)*.

The topics within this section explain the role of recognition projects in batch processing and provide the detailed instructions on how to add and manage recognition projects.

### 11.2.1 What is a recognition project?

A recognition project is a solution that encapsulates resources required by the advanced recognition logic of Intelligent Capture. These resources include templates, index families, free form rules, settings for advanced recognition modules, PAL settings, various configuration settings, scripts, and other data.

You need to create or import a recognition project in your capture system in the following cases:

- You design a CaptureFlow that includes at least one of these advanced recognition modules as a step: Classification, Identification, Extraction, or Collector. These steps link a recognition project during setup. In production, the advanced recognition modules use the settings and resources of the recognition project to identify documents, extract image data, and collect images for PAL.
- You design a CaptureFlow that includes the Completion module as a step. This module requires that you create a document type and define data fields to be

displayed to the Completion operator in production. When you create a document type, you need to point a recognition project that already exists in your capture system. When you save the document type, the data fields appear in the linked recognition project as an index family.

## 11.2.2 High-level steps to use a project in production

Perform the high-level steps outlined in the following table to create a recognition project and use it in production.

**Table 11-1: Recognition projects: high-level steps**

Step	Job	Refer To
1	Create a recognition project or import a ready one into your capture system.	<i>"Adding a new recognition project" on page 337</i>
2	Create a document type and link it to your recognition project. When you click to save a new document type, an index family is added in your project.  You can repeat this step to have multiple index families within one project.	<i>"High-level procedure" on page 345</i>
3	Create templates for your recognition project in Recognition Designer.	<i>OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)</i>
4	Set up project options in Recognition Designer.	<i>OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)</i>
5	Compile your recognition project in Recognition Designer.	<i>OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)</i>
6	Prepare the production environment for the recognition project:	
	(a) Create a shared production folder that can be accessed by the advanced recognition modules in production.	<i>OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)</i>

Step	Job	Refer To
	(b) Specify the production folder path in the <b>RecognitionProjectSharedDirectory</b> parameter in <b>System Configuration -&gt; Global Options -&gt; Deployment Files</b> .	"Recognition project path" on page 67
	(c) Deploy the global options on the production server.	"Uploading or deploying service components including profiles and CaptureFlows" on page 439
7	Send your project to production using Recognition Designer, or copy it to the production folder manually.	<i>OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)</i>
8	After the CaptureFlow is installed, set up its advanced recognition steps to use the recognition project located in the production folder.	"Setting up CaptureFlow steps" on page 448

### 11.2.3 Managing recognition projects

This section describes how to manage recognition projects.

#### 11.2.3.1 Recognition projects list


The **Recognition Projects** list displays all recognition projects available in the current capture system. Use the **Recognition Projects** list to create projects, import external projects in the current capture system, delete the projects, and open any existing project for editing in Recognition Designer.



To open the **Recognition Projects** list, click **Recognition**. The following table describes the **Recognition Projects** list elements.

**Table 11-2: Recognition projects list: user interface elements**

Element	Description
Project Name column	Displays the name of the recognition project.  When opened for the first time, the <b>Recognition Projects</b> list shows the <b>Default</b> project only. This project can be used by default when creating document types.

Element	Description
<b>Description</b> column	Displays the description for the recognition project.  To edit this parameter, open the project in Recognition Designer and use the <b>Project Options &gt; Notes</b> menu command.
<b>File Time</b> column	Displays the timestamp of the file. Is based on the date and time settings on the local machine.
<b>Open</b> button	Opens the selected project for editing in Recognition Designer.
<b>New</b> button	Creates a new recognition project in the current capture system.
<b>Delete</b> button	Deletes the selected recognition project from the capture system and from the disk physically.
<b>Import DPP</b> button	Imports an existing recognition project into the current capture system.   <b>Note:</b> This button is available if Recognition Designer is used with the Advanced Recognition licence.

### Related Topics:

[“Adding a new recognition project” on page 337](#)

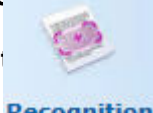
[“Opening a recognition project” on page 338](#)

[“Importing a recognition project” on page 338](#)

[“Deleting a recognition project” on page 340](#)

### 11.2.3.2 Adding a new recognition project

To add a new recognition project:

1. Click **Recognition** .
2. In the **Recognition Projects** list, click the **New** button.
3. In the **Recognition Project – New** window, do the following:
  1. Type the project name.
  2. Select the type of project (AR or IE).
4. Click **Save**.

The new project is created in your capture system. The project name appears in the list of recognition projects.

The project files are located in the `\Default\GlobalData\Recognition\<Project Name>` folder that is created within the Designer working directory. If not **configured**, the working directory is `\Documents\<product_name_and_version>`. The following content is generated inside the *<Project Name>* folder:

- *<Project Name>*.dpp: the project file
- *<Project Name>*.dps: the project settings file
- Subfolders Cache, IdxClasses, Resources\OCR, Resources\Scripts (contains a default script file)

### Related Topics:


[“Opening a recognition project” on page 338](#)

[“Deleting a recognition project” on page 340](#)

#### 11.2.3.3 Opening a recognition project

To open a recognition



1. Click **Recognition** .
2. In the **Recognition Projects** list, select the project and click **Open**.
3. The project opens in Recognition Designer.

#### 11.2.3.4 Importing a recognition project

To include an existing recognition project in your capture system, import it. Use import to add a project from a different machine, from a different capture system, or a project developed in an earlier version of Recognition Designer or in Dispatcher Manager.

When importing Dispatcher projects, the following limitations apply:

- Projects created in Dispatcher 6.0 SP3 can be imported from a directory with read-write permission for a current user.
- Projects created in Dispatcher 6.5, 6.5 SP1, and 6.5 SP2 can be imported from a directory with read-only permissions for current user.
- Before importing projects created in Dispatcher 5.x, upgrade them to version 6.0 or 6.5.

To import a recognitic




1. Click **Recognition** .
2. In the **Recognition Projects** list, click **Import DPP**.

3. In the file lookup window, navigate to the project file and confirm your choice. Import starts automatically. The progress bar in the **Project Import** popup window displays the status of the operation. Project import includes the following operations:
  - The recognition project files are imported to folder \<User Name> \GlobalData\Recognition\<Project Name> inside the Designer working directory. If not **configured**, the working directory is \Documents \<product\_name\_and\_version>.
  - The imported project appears in the **Recognition Projects** list.
  - A document type is generated for each index family found in the imported project. The document types appear in the **Document Types** list.
4. (Optional) Click **Yes** in the **Project Imported with Warnings** message box to view the migration report.
5. The imported project opens in Recognition Designer.
6. Open the generated document type(s) to make sure that they are available for edit. Make sure that all fields were added from each index family to the corresponding document type. If **naming conventions** for document types or index fields were violated, you may find the following modifications in the document type names and field names:
  1. The prohibited symbols found in the field name are removed. The prohibited symbols include spaces, underscores, not alphanumeric characters.
  2. If the first symbol is a digit, the "A" character is inserted before it.
  3. If the name length is longer than 64 symbols, the name is shortened.
  4. If the file name is empty (for example, after removing all prohibited symbols), it is named starting with the "A" character.
  5. If there is duplicate field name in the document type, a number is added.
7. (Optional) If any incorrect fields or index families are found during import, a warning appears at the end of the import process. To investigate the issues, view the migration log file and messages in the **Problems** panel. The migration report can be found in the <Project Name>\_Migration.txt file inside the project folder.

### 11.2.3.5 Deleting a recognition project

When you delete a recognition project, it is permanently removed from the capture system and from the file system. The **Default** project cannot be deleted.

#### To delete a recognition project

1. Click **Recognition**  **Recognition**.
2. In the **Recognition Projects** list, select a project (or projects) and click **Delete**.
3. In the prompt dialog box, click to delete all source files.

If a deleted project was associated with any document types, the document types are marked as unreadable and cannot be opened. To open an unreadable document type, either recreate a project with the same name or import the deleted project (if available).

#### Related Topics:

*“Maintaining integrity between advanced recognition projects and document types” on page 380*

## Chapter 12

# Designing a document type

### Overview

A document type defines the data entry form on which operators validate and fix the extracted data for a class of documents (for example, invoices, purchase orders, or questionnaires). Specifically, document types define operator assistance, validation and auto-population rules and patterns, field data types, and the format and arrangement of the fields in the client. For example:

- The number of digits and their grouping in a phone number could be automatically validated by a pattern.
- If not all numbers in a field for which a phone number was expected were recognized by optical character recognition, then an operator could manually correct the phone number by looking at the page image in the client application's preview window.

Furthermore, each document type is associated with either a machine-learned information extraction project or manually-created recognition project, both of which perform the actual classification of documents and the extraction of the document data.

## 12.1 Document types and Intelligent Capture runtime components

### 12.1.1 Which Intelligent Capture runtime components use document types?

Document types are used by Intelligent Capture runtime components that work with document data (fields), such as Classification, Identification, Extraction, Completion, Collection, and Web Client.

You need document types in the following cases:

- Your CaptureFlow includes a step or steps that execute any of these modules: Classification, Identification, Extraction, Collection. All these modules keep their configuration settings and definitions in a recognition project to which they are linked during step setup. The recognition project includes at least one index family – a set of fields to be extracted from the image by the *OCR* logic. To add an index family to a particular recognition project, you need to create a document type associated with that project and define data fields in the document type.
- Your CaptureFlow includes steps that execute any of these operator modules: Identification, Completion, Standard Import with the distributed capture profile

(runs Intelligent Capture Web Client). When running, these runtime components display a data entry form in which the operator views and validates values in the fields. The fields, their layout, properties, and population and validation behavior are defined in the document type. The required document types are associated with a CaptureFlow step during setup either directly, or through the recognition project.



**Note:** Only Completion and Web Client use controls and the layout of the data entry form.

Population rules are used to extend the recognition logic of Identification, Extraction, Completion, or Web Client. Where the data recognition may fail or cannot be applied, use a rule to populate the field(s).

Validation rules are used to enhance the recognition logic of Identification, Extraction, Completion, or Web Client. Use a rule to validate the values in the extracted and populated fields.

When you create a document type, you map it to a particular recognition project, irrespective of further use of that document type as a standalone profile or as part of a recognition project. You can create one recognition project for your CaptureFlow, associate it with multiple document types, and use that project to set up all advanced recognition steps of your CaptureFlow.

You can also map document types to different recognition projects and use them in different steps in the same or different CaptureFlows, or dynamically link a particular project to a step at runtime using scripting. For example, you can set up multiple Completion steps with the same document type, or multiple Identification and Standard Import steps with the same recognition project.



**Note:** You cannot associate the same document type with multiple recognition projects and in this way create similar index families in different projects. As a workaround, create a copy of this document type with a different name for each recognition project.

### **Related Topics:**

[“Identification restrictions” on page 343](#)

[“Web Client restrictions” on page 343](#)

## 12.1.2 Identification restrictions

Identification does not support all of the document type features and settings. The limitations apply to the displayed fields, rules, document settings, and scripting. In particular:

- Identification does not display the data entry form as defined in the document type. When the operator works with a classified page, only *pre-index* fields particular to the page template are displayed; the layout of the data entry form is not followed. Fields for pre-indexing can be defined for each index family (an internal representation of the document type) in the recognition project. Only editable fields can be selected; controls, including buttons, are never displayed to the operator.



**Note:** Pre-indexing is not supported for information extraction projects.

- Not all document settings apply to Identification. For details, see [“Document properties reference” on page 381](#).

### Related Topics:

[“Which Intelligent Capture runtime components use document types?” on page 341](#)

## 12.1.3 Web Client restrictions

Because of its browser-based user interface, Web Client does not support all of the document type features that are supported for the Identification and Completion modules.

The following table describes the features that Web Client does not support. If the feature is not listed here, then it is fully supported.

User interface category	Document type feature	Web Client support
Field	Combo Box	Not Supported.
Control	Text Block	Not supported.
	Group Box	Not supported.
	Image	Not supported.
	Table	Supported, but the number of rows to display is fixed; a scroll bar is used when the number of rows exceeds this fixed number.
Field Properties	Custom Value	Not supported.
	Label Style	Not supported.
	Show in Summary	Not supported.

User interface category	Document type feature	Web Client support
	Style	Not supported.
	Auto Pan Group	Not supported.
	Autosuggest	The behavior is browser-based.
	Autoexpand	The behavior is browser-based.
Document Properties	Locale	<p>The locale is supported for the default formatting of date/time and numeric values and is determined according to the following priority:</p> <ol style="list-style-type: none"> <li>1. The selected <i>&lt;country&gt;</i>.</li> <li>2. <b>Operator's Locale:</b> The locale is determined according to the following priority: <ol style="list-style-type: none"> <li>a. Web Client's <b>culture</b> argument.</li> <li>b. The first entry in the browser's preferred language list.</li> </ol> </li> <li>3. If an error occurs, then the <b>English (United States)</b> locale (the default) is used.</li> </ol>
	Help URL	Not supported.
	Default Form Position	Not supported.
	Document label	Not supported.
	Folder label	Not supported.
	Structured Document	Not supported.
	Custom Value	Not supported.

 **Notes**

For more information, see the following:

- When switching to another document type, Web Client discards all values, except when a field **Name** (as specified in the field's **Data Definition** properties) in the target document type matches the original field name—even if the retained value is invalid in the new field.
- To avoid the unintentional matching of fields, you might want to use reasonably unique names instead of generic names; for example, `invoiceDate` instead of `Date`.

It is recommended that you use a custom date/time format. However, if you choose to use a standard format (in **Field Properties** > **Manual Validation** > **Date/Time Format** or **Number Format**), then they are converted as described in [“Standard date/time format conversions for Web Client”](#) on page 401.

### Related Topics:

[“Which Intelligent Capture runtime components use document types?”](#) on page 341

## 12.2 High-level procedure

1. Create document type by performing one of the following actions:
  - Recognition project  
Click **Document Type** > **New**.
  - Information extraction project  
Select a document type associated with the information extraction project and click **Open**.



**Note:** When the information extraction project is created, a corresponding document type is automatically created.



2. Define the fields as described in [“Defining fields”](#) on page 347.
3. Click **Save As**.


You can also perform the tasks outlined in the following table.





### Notes

For more information, see the following:

- The **Project** column displays each document type’s associated recognition project.
- The **File Time** column displays the timestamp for when the document type file was saved and is based on the date and time setting on your local machine.
- The status of a document type is indicated by a symbol to the right of its name as follows:
  - \* (asterisk): The document type has unsaved changes.
  - : The document type is unreadable. This may be caused by an attempt to edit the document type file (encoded *XML*) in a third-party text editor.
  - : The document type is read-only.

 **Note:** For document type scripting, the document type name determines the .NET class to contain the scripting.


Icon/tooltip/keyboard shortcut	Action and tips
<p><b>Document Types &gt; New</b></p> <p><b>CTRL+N</b></p>	<p>Document types are associated with one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Information Extraction</b> project For each profile in the <b>Information Extraction</b> project, a corresponding document type is automatically created. For more information, see <a href="#">“Information extraction projects” on page 329</a>.</li> <li>• <b>Advanced Recognition</b> project Specify the following property:                             <ul style="list-style-type: none"> <li>– <b>New Document Type Name</b> A descriptive document type name. For classification and data validation and correction purposes, the document type name is displayed in the client.  A new index family with the same name as the document type is added to the associated recognition project.</li> </ul> </li> </ul>
<p><b>Document Types &gt; Open</b></p> <p><b>CTRL+O</b></p>	<p>Edits a document type.</p>
<p></p> <p><b>Save</b></p> <p><b>Save &gt; Save</b></p> <p><b>CTRL+S</b></p> <p><b>Save &gt; Save As</b></p> <p><b>Save &gt; Save All</b></p> <p><b>CTRL+SHIFT+S</b></p>	<p>Saves a document type; duplicates the document type; saves all document types.</p>

Icon/tooltip/keyboard shortcut	Action and tips
Delete DELETE	Deletes a document type.   <b>Notes</b>  For more information, see the following: <ul style="list-style-type: none"> <li>• To delete information extraction-based document types, you must use the <b>Information Extraction Project Editor</b>.</li> <li>• For recognition projects, the corresponding index family is also deleted from the associated recognition project.</li> </ul>
<Enter document type to filter>	Enter a string pattern to filter the list of document types in the <b>Document Type</b> and <b>Project</b> columns.

## 12.3 Defining fields

Perform the steps outlined in “High-level steps for defining document type fields” on page 347 to define document type fields.

**Table 12-1: High-level steps for defining document type fields**

Open this tab:	Do the following:	For details, see:
<b>Fields</b>	Create fields, tables, and controls by clicking the plus sign  , selecting one from the list, and specifying the properties.	“Managing fields, controls, segments, and tables” on page 348
<b>Form</b>	(Optional) Design the form that is displayed to the operator in a client module.  Design the form by creating the fields, tables, and controls and arranging them on the form.	“Designing the layout of the form” on page 350

Open this tab:	Do the following:	For details, see:
<b>Population Rules</b>	<p>(Optional) You might populate fields with data based on a set of rules. For example, you might populate a field using a database query.</p> <p>Create population rules by clicking <b>Population Rules</b> &gt; <b>&lt;New Rule&gt;</b> and specifying the parameters.</p>	<p>“Creating population rules” on page 356</p> <p>“Creating population and validation rules” on page 353</p>
<b>Validation Rules</b>	<p>(Optional) You might validate the data in a field based on a set of rules. For example, you might make sure that the vendor ID is less than or equal to a certain number of digits.</p> <p>Create validation rules by clicking <b>Validation Rules</b> &gt; <b>&lt;New Rule&gt;</b> and specifying the parameters.</p>	<p>“Creating validation rules” on page 361</p>
<b>Document Properties</b>	<p>Specify various document-type-level properties, such as a description to be displayed in the client.</p> <p>Specify document properties by clicking <b>Document Properties</b> and specifying the properties.</p>	<p>“Document properties reference” on page 381</p>

## 12.4 Managing fields, controls, segments, and tables

This section describes the actions that can be performed on fields, controls, segments, and tables on the **Form** or **Fields** tab.



**Note:** For brevity in this section, the term, *components*, is used to see fields, controls, segments, and tables.

Segments are mainly a design time tool used to visually group fields, controls, and tables together on the **Form** tab. The child components in a single segment are displayed all by themselves on the **Form** tab. At runtime, the segments themselves are not displayed and every segment’s child components are displayed on the same form in the order shown in the **Form** or **Fields** tab’s list of components. To view the runtime form, click **Preview**.







 **Notes**


For more information, see the following:

- Deleting a segment also deletes all of its child components.
- A new segment with a child **Text Box** is always inserted at the end of the list of components.
- The segment name can be used for document type scripting; for example, to handle child fields, controls, and tables as a single group.

“Fields, controls, and tables: actions” on page 349 provides a description of the action components and icon/tooltip/keyboard shortcuts.

**Table 12-2: Fields, controls, and tables: actions**

Action	Icon/tooltip/keyboard shortcut	Description
Add components	 <b>Add</b>	<p>In the list of components on the left, select a component, click  and then choose a component.</p> <p>The component is inserted after the selected component.</p>
Delete components	 <b>Edit &gt; Delete</b> <b>DELETE</b>  <b>Remove</b> <b>DELETE</b>	<p>In the list of components on the left, select a component and click .</p> <p>Deleting a table also deletes all of its fields.</p>
Cut, copy and paste components	 <b>Edit &gt; Cut</b> <b>CTRL+X</b> <b>Edit &gt; Copy</b> <b>CTRL+C</b> <b>Edit &gt; Paste</b> <b>CTRL+V</b> <b>Edit &gt; Delete</b> <b>DELETE</b>	<p>In the list of components on the left, select a component and then use the cut, copy, and paste operations to copy and paste it.</p> <p>The component is pasted after the selected component.</p>

Action	Icon/tooltip/keyboard shortcut	Description
Change the <b>TAB</b> order of components	▲ ▼ <b>Move Up</b> <b>Move Down</b>	In the list of components on the left, select a component and click ▲ ▼ .
Rename components	<b>F2</b>	Double-click a component in the list of components on the left and enter a name.
Make components visible on the form or hide them	<input checked="" type="checkbox"/> <b>Field1</b> <b>SPACE</b>	Select or clear the check box next to a component to make the component visible on the form or hide it.  Hidden components might be used to store auxiliary data that is extracted and processed as document data.   <b>Note:</b> The corresponding property in document type scripting has no effect on components.

## 12.5 Designing the layout of the form

To design the layout of the form, perform the following tasks outlined in the table below.

### Notes





For more information, see the following:



- To change visual properties, such as label position and alignment, go to the field's **Visual** properties.
- The Z-order specifies the **TAB** order when fields, controls, and labels overlap each other.

The virtual Z-order layers are as follows:

- **Fields:** All fields are placed on the top-most layer, and their position within this layer is defined by their position in the fields panel list.
- **Controls:** All controls are placed on the background decorative layer, and their position within this layer is defined by their position in the fields panel list.

**Table 12-3: Form tab: actions**

Action	Icon/tooltip/keyboard shortcut	Description
Move fields and controls	Arrow keys	Using the pointer, drag fields and controls.
Resize fields and controls	CTRL+an arrow key	Using the pointer, resize fields and controls.
Select multiple fields and controls	None	Right-click and drag around a group of fields and controls.
	 <b>Select All Below</b>	Selects the current field or control and all the rest below it to the last field or control in the segment or in the table.
Align fields and controls	 <b>Align Left</b>	Select a group of fields and controls and click tool in the toolbar. All selected fields/controls are set to the left, right, or all on the same row on the form, relative to the alignment direction of the first selected item in the fields panel.  This alignment cannot be undone.
	 <b>Align Right</b>	
	 <b>All in a Row</b>	

Action	Icon/tooltip/keyboard shortcut	Description
Set the <b>TAB</b> order	 <b>Re-Order</b>	<p>Use the <b>Re-Order</b> mode to set the <b>TAB</b> order for the fields on the form.</p> <ol style="list-style-type: none"> <li>1. Begin by clicking the field in the canvas that you want to start with, then click the <b>Re-Order</b> button in the toolbar.</li> <li>2. Next, click each remaining field in the canvas in the order in which users should tab to them. As you select each successive field in the canvas, its position in the fields panel changes so that it immediately follows the previously selected field.</li> </ol> <p> <b>Note:</b> If the displayed segment includes a table and fields and controls following that table, the <b>TAB</b> order is defined separately in the pre-table area, in the table, and in the post-table area.</p> <ol style="list-style-type: none"> <li>3. To exit <b>Re-Order</b> mode, press <b>ESC</b>, or click the <b>Re-Order</b> button again, or click in an empty part of the canvas.</li> </ol> <p>Reordered fields cannot be undone. To specify a different tabbing order, enter <b>Re-Order</b> mode again and repeat the steps above.</p>
Preview the form as it would be displayed to the operator	<b>Preview</b>	Click <b>Preview</b> .

## 12.6 Creating population and validation rules

A document type can include the following types of rules:

- **Population rules:** Specify how the given fields will be populated automatically. By default, fields are populated with data that is extracted from the image by the *OCR* logic. However, some fields displayed to the operator in the data entry form do not get data from the image. To auto-populate such a field, create a population rule for it. Population rules can also modify (normalize, format) data in the fields.
- **Validation rules:** Automatically check the validity of data in the specified field or in a group of related fields. Validation rules detect and highlight invalid fields, preventing the operator from forwarding an incompletely or incorrectly processed document to the next step.

For validation of simple number, date, and string/text formats, it is recommended that you use each field's **Validation and Formatting** properties because they might execute faster.

Rules are evaluated in the order in which they appear on the **Population Rules** and **Validation Rules** tabs. Population rules fill out document fields and then validation rules verify the entered data and set invalid fields in error. Population rules can specify dependent fields, which trigger the rule to be executed when any one of the dependent fields is updated. It is recommended that the rules be ordered such that they run from top to bottom and that all dependent fields of the next rule are populated before that rule is triggered. Rules with undefined dependent fields are skipped.

Operators are likely to process document fields based on the field order, which enables the operator to work from top to bottom through the document. During the operator session, an updated field triggers all dependent population and validation rules in a cascade. Cascade execution is based on the effect when an updated field triggers the rule that updates another field that triggers another rule, and so on. Rules in a cascade are executed in the top-to-bottom direction. Every rule in a sequence is triggered only once.

### **Example 12-1:**

The rules are defined in a sequence as follows:

- Rule A (trigger field: X, updated field: Y)
- Rule B (trigger field: Y, updated field: X and Z)

In the example, if the rule **A** is triggered, it updates the field **Y**. This triggers the rule **B**, and it updates the fields **X** and **Z**. The field **X** must trigger the rule **A**, but this rule has already been executed in this cascade; besides, it is located above the last executed rule. So it is not triggered.



## 12.6.1 Identification: order of triggering

Pre-index fields are displayed in Identification's **Index Fields** panel. A page's associated template specifies the pre-index fields to be extracted. The rules are triggered as follows:

- If all of a rule's dependent fields are pre-index fields for the page, then the rule is triggered; otherwise, the rule is skipped.
- The rule can populate or validate only pre-index fields on the same page.

If the template is changed, then the pre-index fields for the new template are extracted and the rules are triggered again.

## 12.6.2 Extraction: order of triggering

During extraction, population and validation rules are triggered in the following order:

1. Non-empty target fields are overwritten. If a population rule returns an empty string, the field is cleared. If the returned value does not match the field's data type, the field is not updated. If any of one of the dependent fields is empty, then the rule is skipped and the next rule is triggered.
2. Each validation rule checks the validity of data entered in one or several index fields. If the field's data is invalid, that field is set in error. If any of the dependent fields is invalid or blank, that rule is skipped and the next rule is triggered.



**Note:** The purpose of the rule-based validation consists in identifying the status of the document. If any field in the document is in error, or a blank field requires a value, or a field requires manual confirmation, the document status is "not completed". Based on its status, a document can be routed to Completion for manual validation, or this step can be bypassed.

For document type scripting, the `DocumentExtracted()` event handler is fired. Population rules (except for one-time population rules) for which the dependent fields were updated by the script are re-triggered. Then validation rules are also re-triggered for all fields that were updated by the script.

### 12.6.3 Completion: order of triggering

The Completion module can use the document type rules when triggered at the document level or higher. In page mode in Completion, rules are not triggered.

If a document has passed through the Extraction step before, Completion receives this document with the document type (*XML*) and extracted fields (*<UIMdata>*). Otherwise, Completion receives the document type only.

1. For each document in the task, Completion validates every field's value against its properties. All invalid fields are set in error.
2. Population and validation rules are executed as described for **Extraction**.



**Note:** One-time population rules are executed for each document once during the batch processing. Therefore, one-time population rules that were triggered in a preceding step are skipped.

3. If defined in document type scripting, the `DocumentLoad()` event handler is fired. Population rules (except for one-time population rules) for which the dependent fields were updated by the script are re-triggered. Then validation rules are also re-triggered for all fields that were updated by the script.
4. The operator receives the task for manual validation.

#### Operator work

The operator views the data entry form in which:

- Auto-populated and validated fields display valid data. If some fields require manual confirmation, such a field is highlighted as a *work item* (in terms of Completion, this is the field that requires operator attention).
- Unpopulated fields (that is, blank fields) might be highlighted as work items, depending on the field properties.
- Fields for which a validation rule failed are highlighted as work items.
- Fields for which a validation rule was skipped are not considered work items and are not highlighted. A rule is skipped if any of its dependent fields is blank or set in error.

The operator needs to process all work items by entering valid data and confirming the results. If no Extraction step was run, all fields are blank; the operator needs to populate all fields with valid data manually and confirm the results.

Every time the field is updated (by the operator or by a population rule), the following logic executes:

- **Field-level validation:** The value in the updated field is checked to comply with the field properties defined in the document type. If this validation fails, the field is in error and highlighted accordingly. Otherwise, all population and validation rules depending on the updated field are executed in a **cascade**.

- **Population rules:** All population rules that include the updated field as a dependent field are re-triggered. The following limitations apply:
  - If the updated field is in error, rules depending on this field are not triggered.
  - If the updated field tries to trigger a rule that has any other dependent field in error, such a rule is skipped and the next rule in a sequence executes.
  - One-time population rules are not re-triggered.
- **Validation rules:** All validation rules that depend on the updated field are re-triggered. The following limitations apply:
  - If the updated field is in error, rules depending on this field are not triggered.
  - If the updated field tries to trigger a rule that has any other dependent field in error, such a rule is skipped and the next rule in a sequence executes.
  - If the rule fails, the field is highlighted in error. All other dependent fields are highlighted as well. Not validated fields are not highlighted.




**Note:** You can use document type scripting to define which dependent fields should be marked in error if the rule fails.

#### 12.6.4 Creating population rules

Click **Population Rules** > <**New Rule**> and specify the parameters outlined in the following table.

**Table 12-4: Population rule parameters**

Parameter	Description
Definition	<p>Defines the method as follows:</p> <p> <b>Note:</b> Edit the definition by clicking it.</p> <ul style="list-style-type: none"> <li>• <b>Expression</b> Create an expression by clicking &lt;Edit expression&gt;. For more information, see “Using expressions in population and validation rules” on page 364.</li> <li>• <b>Query</b> Create a query by clicking &lt;Select query&gt;, selecting the named query, and defining the field mapping. For more information, see “Creating queries” on page 367.</li> <li>• <b>Script</b> Not required. The script to execute is determined by the scope. For more information, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>. To deploy scripts, see “Non-system files for deployment” on page 66.</li> <li>• <b>SnapMatch</b> Create a SnapMatch query by clicking &lt;Edit SnapMatch&gt;, selecting a connection, and defining the field mapping. For more information, see “Creating SnapMatch definitions” on page 360.</li> </ul>

Parameter	Description
<p><b>Dependent Fields</b></p>	<p>Specifies the fields that trigger the rule to be executed when any one of these fields is updated. These fields are known as dependent fields and are enabled when the <b>One-time</b> property is cleared. Also, if any one of the dependent fields is empty, then the rule is not triggered. See the following:</p> <ul style="list-style-type: none"> <li>• <b>Expression</b> Dependent fields are specified in the expression statement.</li> <li>• <b>Query</b> Dependent fields are specified in the query statement.</li> <li>• <b>Script</b> Specify the dependent fields by clicking <b>&lt;Edit fields&gt;</b> and entering the comma-delimited list of fields.</li> <li>• <b>SnapMatch</b> Dependent fields are specified in the query statement.</li> </ul>
<p><b>Method</b></p>	<p>Specifies the method used to populate the target fields as follows:</p> <ul style="list-style-type: none"> <li>• <b>Expression:</b> Result of the regular expression specified in <b>Definition</b>.</li> <li>• <b>Query:</b> Result of a named query.</li> <li>• <b>Script:</b> Result of a method implemented as part of the document type scripting.</li> <li>• <b>SnapMatch:</b> Result of a query to a SnapMatch data file. You generate a SnapMatch data file by creating a <b>SnapMatch</b> connection.</li> </ul>
<p><b>Name</b></p>	<p>Specifies the name of the rule and must follow these conventions:</p> <ul style="list-style-type: none"> <li>• Must be unique in the Document Type</li> <li>• The maximum allowed length is 64 characters.</li> <li>• Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters. Underscores and spaces are not allowed.</li> </ul> <p>If you are defining a rule with scripting, the rule's name will be included in the .NET method:  ExecutePopulationRule&lt;rulename&gt; or  ExecuteValidationRule&lt;rulename&gt;.</p>

Parameter	Description
<b>One-time</b>	<p>Specifies whether to execute the rule once or multiple times during batch execution as follows:</p> <ul style="list-style-type: none"> <li>• If this property is selected, then the rule is executed only once during batch execution. For example, if the rule is executed during extraction, then it would not be executed again later for dependent fields.</li> </ul> <p>For table scope, the rule is executed once for each blank row that an operator inserts; however, if the row is added by either entering text or rubberbanding in the placeholder row, then although this rule is still executed, the entered value is not used in the rule.</p> <ul style="list-style-type: none"> <li>• If this property is cleared, then in addition to being executed once as noted before, it is also executed again for dependent fields.</li> </ul>
<b>Scope</b>	<p>Specifies the scope for target fields as follows:</p> <ul style="list-style-type: none"> <li>• <b>Document:</b> Document-level fields only. Fields in tables are out of scope.</li> <li>• <i>&lt;Table name&gt;</i>: The selected table and all document-level fields. For tables, the rule is triggered for each row. Fields in other tables are out of scope.</li> </ul>
<b>Target Field</b>	<p>Specifies the target fields to be populated with the results of the rule as follows. The assigned value must comply with the data type of the target field. If the data types do not match, then the target field is cleared and set to a default value. See the following:</p> <ul style="list-style-type: none"> <li>• <b>Expression</b> Select one target field to be populated with the expression result (value).</li> <li>• <b>Query</b> Target fields are mapped in the definition.</li> <li>• <b>Script</b> Target fields are specified in the script.</li> <li>• <b>SnapMatch</b> Target fields are <b>mapped</b> in the definition.</li> </ul>

### 12.6.4.1 Creating SnapMatch definitions

A population rule can use a SnapMatch data file.

#### To create a SnapMatch definition:

1. Click the **Definition** column.
2. In the **SnapMatch connection** list, select the SnapMatch connection to use.
3. Perform the following actions:
  - **Output Fields Mapping** tab: Set the mapping between the SnapMatch data file columns and the populated fields as follows:

- **Column:** Select the name of the column.
- **Copy To:** Select the corresponding document type field.

- **Confidence:**

A row must pass both of the following minimum confidence value criteria. First, a set of columns is evaluated as specified in **Minimum columns confidence** and then the total confidence of that set is calculated and evaluated against the minimum confidence value as specified in **Minimum confidence for the output**.

- **Minimum confidence for the output:** Specify the minimum confidence value used to match rows. You can use this parameter to ignore rows that have low confidence. The lower you set this value, the higher the probability that a matching row would be incorrect. However, only the single row with the highest confidence is returned.
- **Minimum columns confidence:** Specify the minimum confidence level that is required for a candidate match to be accepted for a set of individual columns, that is a row. This minimum confidence level for a row is also known as a *filter*.

To add a confidence level to a set of columns, click **Add Filter** and then click the highlighted area under each column.

To delete the confidence levels for a set of columns, delete the entire filter by selecting the filter and then clicking **Delete Filter**.

#### Related Topics:


[“Defining queries” on page 68](#)


[“Defining connections” on page 34](#)


## 12.6.5 Creating validation rules

Click **Validation Rules** > **<New Rule>** and specify the parameters outlined in the following table.

**Table 12-5: Validation rule parameters**

Parameter	Description
<b>Definition</b>	<p>Defines the method as follows:</p> <p> <b>Note:</b> Edit the definition by clicking it.</p> <ul style="list-style-type: none"> <li>• <b>Expression</b> Create an expression by clicking <b>&lt;Edit expression&gt;</b>. For more information, see <i>“Using expressions in population and validation rules”</i> on page 364.</li> <li>• <b>Query</b> Create a query by clicking <b>&lt;Select query&gt;</b>, selecting the named query, and defining the field mapping. For more information, see <i>“Creating queries”</i> on page 367.</li> <li>• <b>Script</b> Not required. The script to execute is determined by the scope. For more information, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>. To deploy scripts, see <i>“Non-system files for deployment”</i> on page 66.</li> </ul>

Parameter	Description
<p><b>Dependent Fields</b></p>	<p>Specifies the dependent fields that trigger the rule to be executed when updated by operator input, a population rule, or OCR result. Also, if any one of the dependent fields is empty, then the rule is not triggered. When a validation rule fails, all of its dependent fields are also set in error. See the following:</p> <ul style="list-style-type: none"> <li>• <b>Expression</b> Dependent fields are specified in the expression statement.</li> <li>• <b>Query</b> Dependent fields are specified in the <b>Input Parameters</b> list defined in the query statement.</li> <li>• <b>Script</b> Specify the dependent fields by clicking <b>&lt;Edit fields&gt;</b> and entering the comma-delimited list of fields.</li> </ul> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>– Each time you add a dependent field, you must also add the field to scripting and vice-versa. Otherwise, inconsistent rule behavior could occur.</li> <li>– Instead of having all dependent fields set in error upon validation rule failure, you could use <code>IUimDataContext.UnmarkFieldForValidationRule()</code> to specify the particular dependent fields to unmark and thus prevent them from being shown in error. This is useful when the value of a particular dependent field is always going to be correct. For more information, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</li> </ul>

Parameter	Description
<b>Error Message</b>	<p>(Optional) Specifies the error message text that displays if the rule fails. See the following:</p> <ul style="list-style-type: none"> <li>• To insert a new line, press <b>CTRL +ENTER</b>.</li> <li>• To include a URL in the error message, use the following syntax:  <code>[ &lt;label&gt;   &lt;URL&gt; ]</code></li> </ul> <p>For example:</p> <pre>[Click here for help.   file:// 10.13.164.22/user/mycompany/ mycompanysite.htm]</pre> <pre>[Click here for help.   http:// mycompany.com]</pre> <p> <b>Note:</b> In Identification, the message appears in the small area above the <b>Index Fields</b> panel. It is recommended to make the error message short enough so that the Identification operator could see all index fields without scrolling or resizing the <b>Index Fields</b> panel. You could also adjust the font size in the <b>MessageBar system style</b>, which applies to both the Identification message area and the Completion message bar.</p>
<b>Method</b>	<p>Specifies the method used to validate the target field as follows:</p> <ul style="list-style-type: none"> <li>• <b>Expression:</b> Validates a field using the expression in the <b>Definition</b> column.</li> <li>• <b>Query:</b> Validates that the field's value exists in an external data source.</li> <li>• <b>Script:</b> Validates a field or multiple fields using a method in the document type scripting. Use scripting to define a complex business rule such that cannot be defined using the other methods. For example, calculations that include or exclude one field value based on another cannot be defined using an expression. Row-specific validation must also be performed using a script.</li> </ul>

Parameter	Description
Name	<p>Specifies the name of the rule and must follow these conventions:</p> <ul style="list-style-type: none"> <li>• Must be unique in the Document Type</li> <li>• The maximum allowed length is 64 characters.</li> <li>• Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters. Underscores and spaces are not allowed.</li> </ul> <p>If you are defining a rule with scripting, the rule's name will be included in the .NET method:  ExecutePopulationRule&lt;rulename&gt; or  ExecuteValidationRule&lt;rulename&gt;.</p>
Scope	<p>Specifies the scope for target fields as follows:</p> <ul style="list-style-type: none"> <li>• <b>Document:</b> Document-level fields only. Fields in tables are out of scope.</li> <li>• &lt;Table name&gt;: The selected table and all document-level fields. For tables, the rule is triggered for each row. Fields in other tables are out of scope.</li> </ul>

### Related Topics:

[“Customizing system styles” on page 52](#)

## 12.6.6 Using expressions in population and validation rules

### Overview

Expressions can be used in both population and validation rules as follows:

- To populate a single target field, a population rule can use a value expression; that is, an expression resulting in a value of a String, DateTime, Integer, Double, or Boolean data type.
- To validate fields, a validation rule can use a conditional expression; that is, an expression resulting in a value of type Boolean. If the expression returns True, the validation rule succeeds; if the expression returns *False*, then the validation rule fails.

### Limitations

An expression used in the document type context cannot use **data paths** to reference *UimData* values (document type fields). To read/write data in the document type fields, use the field name itself.

When used in the document type context, a value expression cannot access IA values in the batch nodes using **data paths**. Node values are out of the rule's scope. If you need to populate document type fields from IA values, you might use scripting.

Another way to populate document type fields from IA values is to copy data from the required IA values into the *InUimData* array of values. You can do this in CaptureFlow scripting or by adding an IA value assignment on the CaptureFlow chart before the step linking the document type. This scenario is described in [“Transferring document data” on page 116](#).

### 12.6.6.1 Details

#### Expression syntax

The expression statement that you enter in the **Expression Editor** window must comply with the value expression syntax described in the [“Programming Reference – Expressions” on page 457](#). The summary is as follows:

- Curly braces are not used.
- An expression can include only fields within the rule's scope.
- An expression can include supported **functions** and **operators** for data processing. Nested constructs are allowed.

For population rules, expressions can also be used for string manipulations (Regex, replacing characters, trimming, ToLower or ToUpper conversion, and others), for data conversion (string to number, number to string), string formatting, and resetting date/time. For example, the following expression uses the `RegexReplace` function to normalize the value in the *ItemDescription* field. The function takes the first 30 characters of the item description and trims extra characters:

```
RegexReplace(ItemDescription, "{0,30}.*", "$1", "false", "false")
```



**Note:** For more examples of using the universal `RegexReplace` function in expressions, see [“RegexReplace” on page 473](#).

For example, the following expression for a validation rule uses the `Length` function to verify that the **VendorID** field includes the identifier of the valid length:

```
Length(VendorID)>5
```

#### Expression result (population rules only)

The expression result is a value to be set to the target field. The data types of the expression result and of the target field must match. An attempt to populate a field with a value of a mismatching data type clears the target field or sets a default value specified in the field properties. See [“Matching data types” on page 366](#) for matching data types.

**Table 12-6: Matching data types**

Document type field	Expression value
String/Text	String
Date (extended with date and time format)	DateTime
Number (extended with number format)	Integer Double
Boolean (checkboxes only)	Boolean

For example, the following expression, which calculates the *Sum* table field, all of the fields are of type Number:

```
UnitsOfMeasure*Price*Quantity*Discount
```

**Related Topics:**

- [“Creating population rules” on page 356](#)
- [“Creating validation rules” on page 361](#)
- [“Programming Reference—Expressions” on page 457](#)
- [“Data paths” on page 459](#)
- [“Functions” on page 463](#)
- [“Operators” on page 459](#)

**12.6.6.2 ArraySum function****Syntax:**

```
double ArraySum(<expression>)
```

Evaluates the expression against each row and then returns the sum of all the table’s row results. An *expression* consists of the following:

- At least one column.
- All columns must be from the same table.
- Table columns, non-table fields, numeric constants, and functions.
- Any of the following operators:
  - +
    - 
    - \*
    - /

The function can only be used in the validation and population rules as follows:

- In the expression and query methods
- For document scope

For validation rules, the following applies:

- The validation rule result is part of server-side validation.
- For the expression method, the dependent fields are specified; thus, if the validation rule fails, they are all marked as invalid.
- For the query method, you can use the function to evaluate the query input parameters.

For population rules, the following applies:

- You can use a calculation button inside the target field to execute the rule.
- For the expression method, the specified target field is populated with the return value.
- For the query method, you can use the function to evaluate the query input parameters.

### Example

The expression, which multiplies two table column values (one of which is divided by 100) and one non-table field, is evaluated for each row and then returns the sum of all the row results.

```
ArraySum(TableField1 * (TableField2 / 100) * NonTableField3)
```

## 12.6.7 Creating queries

Expressions can be used in both population and validation rules as follows:

- To validate the data in one or multiple fields. The query result is a dataset that must include the expected number of rows. If the returned dataset contains an unexpected number of rows, then the rule fails, an error message (if specified) is displayed, and all dependent fields in the client are highlighted in error.
- To populate one or multiple target fields with the query results from an external data source. The query result is a dataset that includes one to many fields (columns) and returns zero, one, or multiple rows. If zero rows are returned, then the target fields are not populated.

A query can populate the following types of fields:

- A set of text fields (document and table fields)
- A list box or drop-down list box

### To create queries for rules:

1. In the row for the rule, click **<Select query>** in the **Definition** column.

2. In **Query Name**, select the query.

**Notes**

For more information, see the following:

- Typing in the **Query Name** box filters the list.
  - The named query must have been uploaded to the server; locally stored queries cannot be accessed.
3. (Optional) For each of the query's input parameters listed in **Input Parameters**, specify a value to replace the input parameter (at runtime) by clicking the **Value** column and specifying an expression. See the following:
    - For validation, an input parameter is typically initialized with a field's data, therefore you would need to specify the field name.
    - For population, you might specify a constant, the result of a function to be assigned to the input parameter, or the value of a field name to be passed in the query as a condition.



**Note:** The fields displayed in the autocomplete box are restricted to the rule's scope.

4. (Validation rules only) In **Expected Returned Rows**, select the number of rows you expect in the returned dataset.
5. (Population rules only) For the **Output** options, set the mapping between the dataset columns and the populated fields as follows:

- **Field Values**

Use this option to populate particular fields. The first row in the returned dataset populates the fields; other rows are ignored. For a table, the query populates one table row. The definition of the rule maps the dataset columns to the fields to be populated. If the scope of the rule is a particular table, only table fields can be populated. For each field, click **New Mapping Field Value** to set the mapping as follows:

- **Column:** Specify the name of the dataset field exactly as specified in the query statement. The entered value is not verified.
- **Copy To:** Select the field to be updated from this dataset column.

**Copy To** lists fields that are in the scope of the rule. If the scope is a *<table name>*, you should map table fields only.

- *<list\_box\_name>*(**Choice List**)

Use this option to populate a list box or a drop-down list box with a set of values. The first two columns in the returned dataset form the **choice list** in the format *<ChoiceName>* (*<ChoiceValue>*); other columns are ignored. The *<ChoiceName>* comes from the first dataset column and displays to the operator. The code-facing *<ChoiceValue>* comes from the second dataset

column. If only one column is returned, both values are populated from the first column. The populated field can be a document field or a table field.

## 12.6.8 Example: Using population rules

The following example demonstrates how to use population rules to populate text boxes with calculated values and values from a text file:

- “Document type” on page 369
- “Population logic” on page 370
- “Required components” on page 370
- “Population rules” on page 371
- “Testing” on page 373

### Document type

The example uses a custom document type that serves for processing scanned purchase orders. You can create a similar document type in your capture system for testing purposes:

The screenshot shows the 'Document Types' configuration window for 'TestDoctype'. The interface is divided into several sections:

- Segment1:** Contains text boxes for OrderID, Date, Company, Address, and ZIP code, and a checkbox for StatusPaid.
- OrderedItems:** A table with columns: Name, Quantity, Units (drop-down), Price, Type of goods (drop-down), Discount (%), and Sum.
- Section1:** A Summary box containing text boxes for Positions and Total.

The left sidebar shows a list of fields with checkboxes, including OrderID, Date, Company, Address, ZIPcode, StatusPaid, OrderedItems (with sub-items ItemId, Name, Quantity, UnitsOfMeasure, Price, TypeOfGoods, Rate, Sum), SummaryGrpBox, OrderedItemNumber, and Total.

**Figure 12-1: Document type**

This document type implies that each purchase order includes the list of ordered items. The items will be extracted into the **OrderedItems** table with the following array fields: **ItemId** (text box, hidden), **Name** (text box), **Quantity** (text box), **UnitsOfMeasure** (drop-down list box), **Price** (text box, price per unit), **TypeOfGoods** (drop-down list box), **Rate** (text box, shows a discount rate for the selected type of goods), and **Sum** (text box, the sum total calculated per row).

**!** **Important**

Set the **Record View** mode. The **Grid View** mode will not allow you to expand drop-down list boxes in a table at runtime.

**Population logic**

The **TypeOfGoods** list box is automatically populated from the choice list. The **Rate** text box (labelled as **Discount (%)**) is automatically populated from the **Rates.txt** file every time the **TypeOfGoods** list box is updated by the operator. The **Sum** text box is automatically populated with the expression result.

To implement automatic population, we need to create two population rules. One of these rules is based on a text file query that uses a file system connection.

**Required components**

Before creating rules, prepare the required components outlined in [""](#) on page 370.

**Table 12-7:**

Required component	Details
Rates.txt file	<p>Create the <b>Rates.txt</b> file as follows and save it in the local folder:</p> <pre>GoodsId,Rate Children,10 Common,0 Food,10 Medical,10 Social,15</pre> <p>The first line is the file header. Other lines store the types of goods and discount rates for each type.</p>
RatesFileConnect connection	<p>Create a connection to the <b>Rates.txt</b> file with the <b>RatesFileConnect</b> name and the following properties:</p> <ul style="list-style-type: none"> <li>• <b>File Directory:</b> Specify the full path to the <b>Rates.txt</b> file.</li> <li>• <b>File Name:</b> Specify <b>Rates.txt</b>.</li> </ul>
TypeOfGoods choice list	<p>Populate the choice list of the <b>TypeOfGoods</b> drop-down list. The items added to the choice list must be spelled exactly as in the <b>Rates.txt</b> file.</p>

Required component	Details
GetRate query	<p>To populate the <b>Rate</b> field, create a query with the <b>GetRate</b> name and the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Query Type:</b> Specify <b>TextFile</b>.</li> <li>• <b>Connection Name:</b> Expand the list and select <b>RatesFileConnect</b>.</li> <li>• <b>Description:</b> Specify the description of the query. For instance, "Get the discount rate for the selected type of goods from the Rates.txt file".</li> <li>• <b>Text File Query:</b> Specify <b>GoodsId= : Goods Id</b>. The first name is the name of the field in the text file header. The second one is the input parameter which name must match the header field name. The value selected in the <b>TypeOfGoods</b> list will be passed to the input parameter, and the query will be executed with this filter.</li> </ul> <p>This query will extract one file record with the type of goods matching the value in the <b>TypeOfGoods</b> list.</p>

Test the connection and the query. Upload them to the server (**System > Deployment > check components > click Upload**).

### Population rules

Go to the **Population Rules** tab and create the population logic outlined in the following table.

**Table 12-8:**

Goal	Details
Populating the <b>Rate</b> field	<p>Create a population rule with the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Name:</b> Specify <b>PopulateRate</b></li> <li>• <b>Scope:</b> Expand the list and select <b>OrderedItems</b> (the table name)</li> <li>• <b>Method:</b> Expand the list and select <b>Query</b></li> <li>• <b>Definition:</b> Click &lt;Select query&gt; and specify the following query parameters: <ul style="list-style-type: none"> <li>– <b>Query Name:</b> Expand the list of queries and select <b>GetRate</b></li> <li>– <b>Input Parameters:</b> Double click in the <b>Value</b> column and specify <b>TypeOfGoods</b> – the array field name whose value will be passed to the query through the <b>GoodsId</b> input parameter</li> <li>– <b>Output:</b> Expand the list and select <b>Field Values</b>.</li> </ul> </li> </ul> <p>Create the following mapping:</p> <ul style="list-style-type: none"> <li>○ <b>Column:</b> Specify <b>Rate</b> (the field in the file header which value will be populated)</li> <li>○ <b>Copy To:</b> Specify <b>Rate</b> (the array field name)</li> </ul> <p>The <b>TypeOfGoods</b> field name (input parameter) appears in the <b>Dependent Fields</b> column automatically.</p> <ul style="list-style-type: none"> <li>• <b>One-time:</b> Must be unchecked</li> <li>• <b>Target Field:</b> Not displayed (defined in the definition's mapping)</li> </ul>

Goal	Details
Populating the <b>Sum</b> field	<p>Create a population rule with the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Name:</b> Specify <b>PopulateSum</b></li> <li>• <b>Scope:</b> Expand the list and select <b>OrderedItems</b> (the table name)</li> <li>• <b>Method:</b> Expand the list and select <b>Expression</b></li> <li>• <b>Definition:</b> Click <b>&lt;Edit expression&gt;</b> and specify the following expression in the Expression Editor window:   <math display="block">\text{Quantity} * \text{Price} * (100 - \text{Rate}) / 100</math> <p>Make sure that all fields involved in the calculation are assigned the <b>Number</b> data type in the <b>Fields</b> tab. Otherwise, the <b>Sum</b> field will not be populated.</p> <p>The text of the expression appears in the <b>Definition</b> column. The <b>Dependent Fields</b> column is updated to show the rule's trigger fields.</p> </li> <li>• <b>One-time:</b> Must be unchecked</li> <li>• <b>Target Field:</b> Expand the list and select <b>Sum</b> (the target field to be populated)</li> </ul>

The rules in the **Population Rules** tab must be ordered as shown above. Both rules have dependent fields that must be populated to let these rules be triggered. When triggered, these rules work in the updated table row.

## Testing


Save the document type and upload it to the server. To test the population rules, create a CaptureFlow with two steps – ScanPlus (level 7) and Completion (level 1). Install a new process to the server from this CaptureFlow. To set up the **Completion** step, double click it. Optionally, enter the name of your test document type in the **Allowed document types** parameter, or you can save the default setup configuration.

Run ScanPlus, create a batch, scan or import any image, and finish the batch. Then run Completion and select the name of your batch to work on. The Completion tool loads the task and displays one document with scanned pages in the **Documents** panel. Since the process does not include any steps for classifying a document, the document type is unknown and the fields are not displayed. Click the **Choose Type** link in the document bar and select your document type. The data entry form appears.

**Figure 12-2: Testing document types**

Start entering values in the table. Use **TAB** to jump to the next field. The **Type of goods** list is populated from the choice list. When you expand it and select a value, the **Discount (%)** field is populated automatically. When you jump into the **Sum** field, it is automatically populated with the result of the expression.

Go back to a completed table row and select a different value in the **Type of goods** field. The **Discount (%)** and **Sum** fields are recalculated and updated automatically. Change the value in the **Quantity** or **Price** field and press **TAB** to confirm the update. The **Sum** field is recalculated and updated accordingly.

 **Note:** If any rule needs to be fixed, you can make updates and re-test them immediately on the same batch. Exit from Completion without changes to send the Completion task back to the pool. Then make your updates in Designer, save the updated components, and upload them on the server. Then go to Completion, choose the same batch, and click **Get Work** to pick up the same task from the pool.

### 12.6.9 Example: Using validation rules

The following example demonstrates how to use validation rules to verify data in the extracted fields using expressions and document type scripting:

- “Document type” on page 375
- “Complementary population and validation logic” on page 375
- “Validation rules” on page 375
- “Required components” on page 376
- “Document type scripting” on page 377

- “Testing” on page 378

## Document type

This example uses the document type described in “Example: Using population rules” on page 369.

## Complementary population and validation logic

Following up the example provided in “Example: Using population rules” on page 369, along with the population rules we add two validation rules to help operators verifying the **Sum** and **Total** fields.

In this scenario, the population logic serves to automatically calculate and populate fields, and the validation logic serves to verify data manually entered from the image:

- When the operator populates the **OrderedItems** table, the value in the **Sum** column is calculated automatically and populated by the **PopulateSum** rule.
- If the operator sees a different sum in the image, they update the **Sum** field manually. The **ValidateSum** rule checks the validity of entered data and highlights the field if it is invalid.
- When all table data is populated, the operator populates the **Total** field from the image. This field must show the total amount of values shown in the **Sum** table field. The **ValidateTotal** rule checks the validity of entered data and highlights the field if it is invalid.

## Validation rules

Go to the **Validation Rules** tab and create the validation logic outlined in the following table.

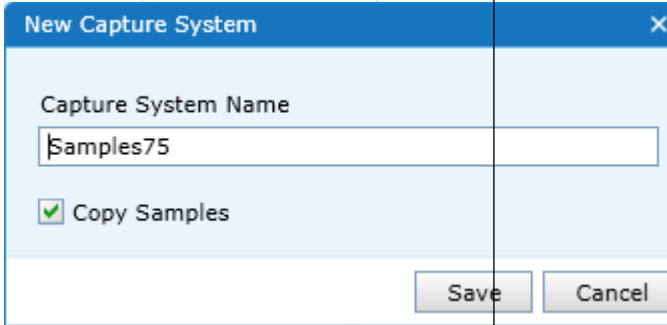
**Table 12-9:**

Goal	Details
Validating the <b>Sum</b> table field	<p>Create a validation rule with the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Name:</b> Specify <b>ValidateSum</b></li> <li>• <b>Error Message:</b> Specify <b>One or several highlighted fields in this row contain invalid data</b></li> <li>• <b>Scope:</b> Expand the list and select <b>OrderedItems</b> (the name of the table section)</li> <li>• <b>Method:</b> Expand the list and select <b>Expression</b></li> <li>• <b>Definition:</b> Click <b>&lt;Edit expression&gt;</b> and specify the following expression in the Expression Editor window:   <math display="block">\text{Sum} = (\text{Quantity} * \text{Price} * (100 - \text{Rate}) / 100)</math> </li> </ul> <p>Make sure that all fields involved in the calculation have the <b>Number</b> data type assigned in the <b>Fields</b> tab of Document Type Editor. Otherwise, the <b>Sum</b> field will not be calculated.</p> <p>The text of the expression appears in the <b>Definition</b> column. The <b>Dependent Fields</b> column is updated automatically to show the rule's trigger fields.</p>
Validating the <b>Total</b> field	<p>Create a validation rule with the following properties:</p> <ul style="list-style-type: none"> <li>• <b>Name:</b> Specify <b>ValidateTotal</b></li> <li>• <b>Error Message:</b> Not required. The error message is defined in scripting.</li> <li>• <b>Scope:</b> Expand the list and select <b>Document</b></li> <li>• <b>Method:</b> Expand the list and select <b>Script</b></li> <li>• <b>Definition:</b> Not required.</li> <li>• <b>Dependent Fields:</b> Click <b>&lt;Edit fields&gt;</b> and specify the following in the appeared Expression Editor window: <b>Sum, Total</b></li> </ul>

**Required components**

One of the validation rules in the example uses document type scripting. Before creating this rule, prepare as outlined in the following table.

**Table 12-10:**

Required component	Details
Programming environment	Any programming environment that supports the latest version of .NET Framework supported by Intelligent Capture. We recommend to use Microsoft Visual Studio. The <i>Release Notes</i> contains the information on supported versions.
Visual Studio project for a C# or Visual Basic class library	
A directory for storing script source code	<p>Once you have created the capture system, the sample script sources are located at:</p> <pre>&lt;drive&gt;:\Users\&lt;User&gt;\Documents \&lt;product_name_and_version&gt; \Samples75\ScriptSource\Profile Scripts\.</pre> <p>Copy this folder to the same location for your system or create the corresponding ScriptSource subfolder if it does not exist:</p> <pre>...\Documents\&lt;product_name_and_ version&gt;\&lt;Your system&gt; \ScriptSource\Profile Scripts\</pre>

## Document type scripting

In your programming tool, open the `ProfileScripts.sln` source file and create the `.cs` (otherwise, `.vb`) class file for your document type.

In case of C#, the text of the file may look as follows:

```
namespace Custom.InputAccel.UimScript{
    using System;
    using System.Windows;
    using Emc.InputAccel.UimScript;

    public class ScriptTestDoctype : UimScriptDocument{

        public void ExecuteValidationRuleValidateTotal(IUimDataContext dataContext){
            int n = dataContext.FindTableSection("OrderedItems").RowCount;
            decimal expectedTotal = decimal.Zero;
            for (int index = 0; index < n; index++)
                expectedTotal += dataContext.FindArrayFieldDataContext("Sum")
```

```
[index].ValueAsDecimal;

        if (expectedTotal !=
dataContext.FindFieldDataContext("Total").ValueAsDecimal) throw new
ApplicationException("Total should be: " + expectedTotal);
    }
}
}
```

Compile the script. Before compiling, close any modules that may lock the script for reading, such as Designer, Completion, Identification, Image Processor, and NET Code Module. Copy all DLL files (primary and secondary DLLs) and data files to %AllUsersProfile%\EMC\InputAccel\Custom\DebugBin. The sample project puts the output DLL files to the DebugBin folder of the current user by default. For more information on these procedures, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

## Testing

Upload the updated document type to the server. When you click **System**, the changes in the document type are saved automatically.

To test the population rules, create a CaptureFlow and setup its steps as described in the *Testing* section of “[Example: Using population rules](#)” on page 369. When setting up Completion, leave its **Work Level** parameter to **Document** to let the expression-based validation rule work in the scope of a whole document rather than of a single field. If you choose the **Field** work level, the **ValidateSum** rule will be executed but the operator will not be able to see the results of multi-field validation – fields in error will not be highlighted. Validation rules are not triggered on the **Manual** or **Character** work level.

Create a batch and run Completion as described in “[Example: Using population rules](#)” on page 369. When the table data is populated, change the value in the **Sum** field in any table row and press **ENTER**. The following fields are highlighted in error: **Quantity, Price, Rate, Sum**. If you drop the cursor to any highlighted field, the error message appears in the message area: “One or several highlighted fields in this row contain invalid data”.

▲ Rule Error: (Field 4 of 4): One or several highlighted fields in this row contain invalid data.

Form View

(1)

Name	Quantity	Units	Price	Type of goods	Discount (%)	Sum
Cola	10	Bottles24	10	Food	10	85

(2)

Name	Quantity	Units	Price	Type of goods	Discount (%)	Sum
Apple juice	10	Bottles12	5	Food	10	45

(3)

Name	Quantity	Units	Price	Type of goods	Discount (%)	Sum

Summary

Positions

Total

**Figure 12-3: Rule error**



**Note:** The color of fields in error and the font of the message area can be configured as described in “Defining global options” on page 63.

Enter the correct sum and press **ENTER**. The highlighting is removed, the error message is not displayed.

Populate the **Total** field with an incorrect value. The error message “**The <xxx> value in the Total field does not match the expected value of <yyy>**” appears in the message area. The **Total** field and all **Sum** fields in the table are highlighted in error. Enter the expected value in the **Total** field.

Now change the **Sum** value in any table row and press **ENTER**. The fields in the table row are highlighted again. The value in the **Total** field does not match the total amount of all **Sum** values, however, the **ValidateTotal** rule is not triggered because one of its dependent fields (the **Sum**) is in error. Correct the **Sum** field to show the valid value. Now the **Total** field and all **Sum** values are NOT highlighted in error.

## 12.7 Maintaining integrity between advanced recognition projects and document types

When creating a document type, you link it to a particular recognition project. Inside the project, the document type is represented by an index family with the document type's name. When you add, remove, rename, or reorder index fields in the document type, the saved changes are immediately synchronized with the index family inside the recognition project.

It is important to maintain the integrity between the recognition projects and document types in your capture system. [“Synchronizing recognition projects and document types” on page 380](#) explains the effects of deleting any of these components and how you can restore their integrity.


**Table 12-11: Synchronizing recognition projects and document types**


If this occurs...	The result and resolution is...
A linked recognition project is deleted	<p>The document type cannot be opened until the recognition project is recovered or a replacement project with the same name is created.</p> <p>The index family is regenerated in the project when you reopen and then save the document type in Document Type Editor.</p>
Document type is deleted	The corresponding index family and all related field placements are deleted from the linked recognition project.
The XML file with the document type definition is deleted.	The corresponding index family is orphaned in the linked recognition project. An orphaned index family cannot be removed from the recognition project using Recognition Designer. The only way to delete an orphaned index family is to recreate a document type with the same name and then delete it in Document Type Editor.
Fields are deleted or renamed in the document type	<p>Deleting a field in a document type removes the corresponding index field and the field placement in the linked recognition project.</p> <p>Renaming a field in the document type does not affect its field placement in the linked recognition project.</p>




## 12.8 Document properties reference




Unless otherwise noted, the properties outlined in “[Document properties](#)” on page 381 apply to Identification, Completion, and Web Client.

**Table 12-12: Document properties**

Property	Description
<Comments> area	A description for your document type. This description is displayed in the Designer <b>Document Types</b> profile’s <b>Description</b> column.
<b>Allow Question Marks</b>	<p>If selected, the values in the index fields can include question marks as valid characters.</p> <p>If cleared (default), question marks in the index fields always indicate low-confidence characters that must be repaired (this is, replaced by a valid character).</p>
<b>Custom Value</b>	<p>Optional. Specify the text (for instance, parameters specific for this document type) that is to be retrieved by scripting. This text is not displayed to operators.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

Property	Description
<p><b>Default Form Position</b></p>	<p>Specify the default position of the <b>Form</b> panel in the main window of Completion only. This setting applies when the operator works with this document type.</p> <p> <b>Note:</b> This setting does not affect Identification or Web Client.</p> <p>See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Bottom:</b> Default. The <b>Form</b> panel is docked to the bottom edge of the main window. The <b>View &gt; Form on Right</b> menu option appears dimmed, the operator cannot change the layout of the <b>Form</b> panel.</li> <li>• <b>Right:</b> The <b>Form</b> panel is docked to the right edge of the main window. The <b>View &gt; Form on Right</b> menu option appears dimmed, the operator cannot change the layout of the <b>Form</b> panel.</li> <li>• <b>User Choice:</b> The <b>Form</b> panel is docked to the bottom edge by default. The operator can re-dock the panel to the right edge or return to the default layout using the <b>View &gt; Form on Right</b> menu option. The selected position is saved in the user preferences and always applies when the operator works with this document type.</li> </ul>

Property	Description
<b>Document Label</b>	<p>Specify the format of the document label to be displayed in the <b>Documents</b> panel of Completion module when the operator works with this document type. This label also appears in the status bar of Completion.</p> <p> <b>Note:</b> This setting does not affect Identification or Web Client.</p> <p>Click &lt;Edit label&gt; and enter a string (constant) or specify the <b>format string</b> in the <b>Format String Editor</b> window.</p> <p><b>Example:</b> Document label format: Doc {ND _documentPosition}: {S _docType} for {S LastName}            Displayed label: <b>Doc 2: Order for Smith</b> – the second document of type <b>Order</b> in the folder issued for customer <b>Smith</b>.</p> <p>If you leave this property blank, all documents are displayed with the default <b>Document</b> name.</p>
<b>Don't allow empty masked fields</b>	<p>If this property is selected, masked fields cannot be left blank, even if allowed.</p> <p> <b>Note:</b> By default, this property is hidden; to display it, press <b>ALT+F9</b>.</p>
<b>Folder Label</b>	<p>Specify the format of the folder label to be displayed in the <b>Documents</b> panel of the Completion module when the operator works with this document type.</p> <p> <b>Note:</b> This setting does not affect Identification or Web Client.</p> <p>Click &lt;Edit label&gt; and enter a string (constant) or specify the <b>format string</b> in the <b>Format String Editor</b> window.</p> <p><b>Example:</b> Document label format: Loan {ND _folderPosition}: {S LastName}            Displayed label: <b>Loan 2: Smith</b> – the second folder in a task with loan applications from customer <b>Smith</b>. The batch is structured to include a folder per loan applicant.</p> <p>If you leave this property blank, all folders are displayed with the default <b>Folder</b> name.</p>

Property	Description
<p><b>Help URL</b></p>	<p>Enter the URL of the help file specifically created for documents of the given document type. This file is opened when the operator processes this document type and presses <b>F1</b>. Allowed formats: <code>http://</code> and <code>file://</code>.</p> <p>If you leave this field blank, pressing <b>F1</b> opens <i>OpenText Intelligent Capture - Desktop Client Guide: Identification and Completion (ECPCORE-CCV)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
<p><b>Locale</b></p>	<p>Select the locale to be used for default formatting of date/time and numeric values.</p> <p>For <b>Operator Locale</b>, the locale is determined at runtime: first by the command line locale, and if not given, then by the UI Culture specified for the operator’s machine.</p> <p> <b>Note:</b> For Web Client, see “<a href="#">Web Client restrictions</a>” on page 343.</p>
<p><b>Structured Document</b></p>	<p>This option applies only to structured documents whose templates in the recognition project define field zones.</p> <p>If selected, an image zone is found and displayed for the selected field, even if the field’s value was not extracted (for instance, because the page was not processed by Extraction). The operator can index non-extracted fields manually by clicking an image snippet that is displayed near the selected field.</p> <p>Default: Not selected. If the selected field was not extracted, then the field’s zone is not displayed.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

Property	Description
<b>Confidence Threshold</b>	<p>A user-set value between 1 and 100 that informs the importance of machine learning returning an accurate value. The higher the value, the more important its accuracy. When set, operators must manually confirm the field values with an extraction confidence below the set threshold.</p> <p>A confidence threshold can be set for a field in a document or for the entire document type. This feature provides an extra layer of protection when analyzing sensitive document data (for example, loan amount).</p> <p>Setting a confidence threshold for an entire document provides convenience. In an invoice, the operator might want to reduce the description confidence threshold because they do not need perfect extraction for that value. In comparison, the operator might increase the confidence threshold for a more critical value, like an invoice number.</p>

## 12.9 Field, control, segment, and table reference



**Note:** Some document type field properties are read-only for information extraction projects. You can change them in the corresponding information extraction project.



### 12.9.1 Text Box (field)

“Data definition properties” on page 385 provides a description of data definition properties.

**Table 12-13: Data definition properties**

Property	Description
<b>Custom Value</b>	<p>Custom text that is stored with the field or control. The text is not visible to the operator but can be retrieved with scripting.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

Property	Description
<b>Data Type</b>	The type of data for the field. The data type determines each field's <b>Validation and Formatting</b> properties. For more information, see each field's <b>Validation and Formatting</b> properties.
<b>Extract Page</b>	The successfully extracted value to use for indexing the field for each document. See the following options: <ul data-bbox="868 604 1339 720" style="list-style-type: none"><li>• <b>First</b> (default): The first extracted field indexes the field.</li><li>• <b>Last</b>: The last extracted field indexes the field.</li></ul>


Property	Description
<b>Index Level</b>	<p>The level at which the field is to be indexed:</p> <p> <b>Note:</b> If you need to index a field above the document level for a bundle of documents of different document types, then add the same field with the same <b>Index Level</b> property to all of the document types.</p> <ul style="list-style-type: none"> <li>• If the index level is 1 (Document level), the field will be indexed for each individual document.</li> <li>• If the index level is 2 (Folder level), all documents in one folder must have the same value in this field.</li> </ul> <p> <b>Note:</b> Not applicable for table fields.</p> <p>For the Extraction module, this field is extracted from all documents in the folder and each document has its own value.</p> <p>For the Completion module, if the extracted values do not match, the last extracted value appears in the folder-level field of each document. If the operator populates this field with a different value and confirms it, this value will be propagated to all the documents in the folder. When the operator merges folders, all included documents get into the first folder and display the value specific for the first folder. Where the displayed value differs from the original one, the field requires manual confirmation.</p> <ul style="list-style-type: none"> <li>• If the index level is higher than 2 (Folder level), the field will be indexed for each individual document, but the first extracted or the last confirmed value will be stored in all documents in the index-level node.</li> </ul>



Property	Description
<p><b>Confidence Threshold</b></p>	<p>A user-set value between 1 and 100 that informs the importance of machine learning returning an accurate value. The higher the value, the more important its accuracy. When set, operators must manually confirm the field values with an extraction confidence below the set threshold.</p> <p>A confidence threshold can be set for a field in a document or for the entire document type. This feature provides an extra layer of protection when analyzing sensitive document data (for example, loan amount).</p> <p>Setting a confidence threshold for an entire document provides convenience. In an invoice, the operator might want to reduce the description confidence threshold because they do not need perfect extraction for that value. In comparison, the operator might increase the confidence threshold for a more critical value, like an invoice number.</p>
<p><b>Input Mode</b></p>	<p>One of the following options for the field:</p> <ul style="list-style-type: none"> <li>• <b>Editable/Not Required</b> (default)</li> <li>• <b>Editable/Required</b></li> <li>• <b>Read-Only</b>: Operator input is not allowed</li> <li>• <b>Set by Expression</b>: Not available for fields that are in a table.</li> </ul>
<p><b>Name</b></p>	<p>A name that identifies the field or control. Field names are not displayed to the operator. You can use a field name to reference a field in an expression or document type scripting. A valid field name is as follows:</p> <ul style="list-style-type: none"> <li>• Contains alphanumeric characters from the ASCII range only.</li> <li>• Starts with a letter.</li> <li>• Does not contain spaces.</li> <li>• Is a valid variable name.</li> <li>• Is unique in the list of fields in the document type.</li> </ul>



Property	Description
Value	<p>An <b>expression</b> that initializes the field when the field's input mode is set to <b>Set by Expression</b>; otherwise, leave this property blank.</p> <p>Type in the property, or click the button and build the expression in the <b>Expressions Editor</b> dialog box.</p>



“Validation and formatting properties” on page 389 describes validation and formatting properties.



**Table 12-14: Validation and formatting properties**



Property	Description
Common properties for all data types	
 <b>Note:</b> The <b>Highlight Suggestions</b> property is only valid for Web Client.	
Control Type	The type of UI control to display on the form.

Property	Description
<p><b>Highlight Suggestions</b></p>	<p>Highlight data indicated for the following data types. In the client, <b>Suggest</b> must also be enabled.</p>  <p>See the following types of data:</p> <ul style="list-style-type: none"> <li>• <b>String/Text</b> <ul style="list-style-type: none"> <li>– <b>Pattern:</b> Valid values are specified by <b>Text Mask</b> and <b>Size Range</b>.</li> <li>– <b>Regular Expression:</b> Valid values are specified by <b>Validation Expression</b> and <b>Size Range</b>.</li> </ul> </li> <li>• <b>Date:</b> Valid values are specified by <b>Date Range</b> and <b>Document Properties &gt; Locale</b>; for example, <i>22/02/2019</i> would be valid for a Russia or Germany locale but invalid for a United States locale).</li> <li>• <b>Number</b> <ul style="list-style-type: none"> <li>– <b>General/Integer/Amount/Percent:</b> Patterns must be specified by <b>Number Range</b> and <b>Input Mode</b> with <b>Editable/Required</b> selected.</li> <li>– <b>Decimal:</b> Patterns must be specified by <b>Number Range</b>, <b>Input Mode</b> with <b>Editable/Required</b> selected, and <b>Digits after Decimal</b>.</li> </ul> </li> </ul> <p> <b>Note:</b> Although this property does not affect fields in a table, all table cells are still highlighted, which assists in identifying missing and incorrect rows within a table.</p>

Property	Description
<b>Manual Confirmation</b>	<p>Operator must confirm the field's value by pressing <b>ENTER</b> or clicking a check mark next to the field.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>– For multi-line text fields, pressing <b>ENTER</b> moves the cursor to the next line instead of confirming the value.</li> <li>– If a calculation can be performed for the field, then pressing <b>ENTER</b> the first time performs the calculation and then pressing <b>ENTER</b> again confirms the calculated value.</li> </ul> <p>See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Do Not Confirm</b> (default): The extracted field is never highlighted as a work item.</li> <li>• <b>Always Confirm</b>: The field is highlighted as a work item until confirmed manually.</li> <li>• <b>Confirm On Blank</b>: If blank, the field is highlighted as a work item until confirmed manually.</li> </ul>
<b>Lines</b>	<p>Whether text displays on a single line or on multiple lines:</p> <ul style="list-style-type: none"> <li>• <b>Single line</b>: Use to remove newline characters.</li> <li>• <b>Multiple lines</b>: Use to retain newline characters and display a scroll bar if needed.</li> </ul>
<b>Date data type</b>	
<b>Date Range</b>	<p>Range of dates as follows:</p> <ul style="list-style-type: none"> <li>• Specific dates</li> <li>• An integer that represents the number of minutes after the time of data entry.</li> </ul> <p> <b>Note:</b> Blank or empty dates are not allowed. An empty date defaults to 1900-01-01.</p>

Property	Description
<b>Date/Time Format</b>	<p>Specifies how the date and time is to be formatted and displayed in the field. Select one of the standard formats from the list. To use a custom date format, enter one or more of the <b>custom format codes</b>. Regardless of whether the displayed date and time is truncated (for example, without the time), the entire date and time is always saved in a UIM datetime property.</p> <p> <b>Note:</b> However, for Web Client, if you use a standard format, then they are converted as described in “<b>Standard date/time format conversions for Web Client</b>” on page 401.</p>
<b>Number data type</b>	
<b>Digits after Decimal</b>	(Only when <b>Validation Type</b> is <b>Decimal</b> ) The number of digits allowed to the right of the decimal point.
<b>Number Format</b>	<p>Specifies how a number is to be formatted and displayed in the field. To override the default precision for a number format, type one of the standard format codes (G, N, F, P) followed by the desired precision. For example, to change from the default 2-decimal <b>Number</b> format (1,234.00) to a 3-decimal format (1,234.000), enter <b>N3</b>. To use a custom number format, enter one or more of the <b>custom format codes</b>. Regardless of whether or not the displayed number’s precision is truncated, the number, including its original precision, is always saved as UIM data.</p> <p> <b>Notes</b></p> <ul style="list-style-type: none"> <li>• For Web Client, if you choose to use a standard format, then they are converted as described in “<b>Standard date/time format conversions for Web Client</b>” on page 401.</li> <li>• Number formatting is used only for display purposes and does not affect the stored value.</li> </ul>

Property	Description
<b>Number Range</b>	<p>The minimum and maximum numbers that are allowed in the field. See the following:</p> <ul style="list-style-type: none"> <li>• For <b>Validation Type</b> set to <b>Amount</b>, the <b>Number Range</b> should include a currency sign; for example, 5 . 12\$.</li> <li>• For <b>Validation Type</b> set to <b>Percent</b>, the <b>Number Range</b> should include a percent sign; for example, 5%.</li> </ul> <p> <b>Note:</b> Blank or empty numbers are not allowed. An empty number defaults to zero.</p>
<b>Validation Type</b>	<p>The type of pattern for validation as follows:</p> <ul style="list-style-type: none"> <li>• <b>General/Integer/Amount/Percent:</b> Patterns are specified by <b>Number Range</b> and <b>Number Format</b>.</li> <li>• <b>Decimal:</b> Patterns are specified by <b>Number Range</b>, <b>Number Format</b>, and <b>Digits after Decimal</b>.</li> </ul>
<b>String/Textdata type</b>	
<b>Case</b>	<p>The required character case.</p> <p>If <b>Uppercase Only</b> is selected, then the field value is always converted to uppercase regardless of whether the value is set manually (typing, rubberbanding) or programmatically (scripting, rules, extraction).</p>
<b>Formatting Expression</b>	<p>A regular expression to use for formatting data. The formatted data is stored as the UIM data.</p> <p> <b>Note:</b> <b>Validation Type</b> must be set to <b>Regular Expression</b>.</p>
<b>Include Placeholder Characters</b>	<p>Whether placeholder characters in a text mask are to be saved/exported with the field data.</p>
<b>Size Range</b>	<p>The minimum and maximum number of characters that are allowed in the field.</p>


Property	Description
<p><b>Text Mask</b></p>	<p>A <b>mask</b> to restrict operator input in this field to a particular format.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• <b>Validation Type</b> must be set to <b>Pattern</b>.</li> <li>• This text mask is also used to determine which data on the image is valid for highlighting with <b>Highlight Suggestions</b>.</li> </ul>
<p><b>Validation Expression</b></p>	<p>A regular expression to use for validation. To highlight and validate data correctly, the validation expression must work for both the original and formatted data (as specified in <b>Formatting Expression</b>).</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>• <b>Validation Type</b> must be set to <b>Regular Expression</b>.</li> <li>• This pattern is also used to determine which data on the image is valid for highlighting with <b>Highlight Suggestions</b>.</li> </ul>
<p><b>Validation Type</b></p>	<p>The type of pattern for validation as follows:</p> <ul style="list-style-type: none"> <li>• <b>Pattern</b>: a text mask in <b>Text Mask</b> with the <b>Include Placeholder Characters</b> option.</li> <li>• <b>Regular Expression</b>: a regular expression, which is based on the .NET Regex class, for <b>Validation Expression</b> and <b>Formatting Expression</b>.</li> </ul>



“Visual properties” on page 395 describes visual properties.

**Table 12-15: Visual properties**

Property	Description
Alignment	<p>The horizontal alignment of text that is displayed in the field. See the following options:</p> <ul style="list-style-type: none"><li>• <b>Left:</b> Aligned on the left, but ragged on the right.</li><li>• <b>Right:</b> Aligned on the right, but ragged on the left.</li><li>• <b>Center:</b> Ragged on both the right and the left.</li><li>• <b>Justify:</b> Aligned on both the left and the right. Space between words is adjusted.</li></ul>

Property	Description
<p><b>Auto Pan Group</b></p>	<p>(Optional) The name of the auto pan group to which this field belongs.</p> <p>Applies to the Completion module only. Use this property to group two or more fields in the data entry form into an auto pan group. To create a group name, enter it in this property and reuse that name when setting this property for other fields.</p> <p>Default: blank (the field does not belong to any auto pan group).</p> <p>When several fields belong to one auto pan group, in production their zones (if defined) are united into one zone group. When the auto pan mode is enabled, field grouping improves the behavior of the Completion module as follows:</p> <ul style="list-style-type: none"> <li>• <b>Auto Pan without field grouping:</b> Completion displays and highlights the current image zone. When the operator selects the next field, the image scrolls to display and highlight the next zone. If the next zone is located in a different image, Completion switches to that image automatically.</li> <li>• <b>Auto Pan with field grouping:</b> When the operator selects a field that belongs to a group, Completion tries to display all image zones of that group simultaneously. The best results are achieved when all zones in a group are located in one image and the image is scaled to the page width. In that case, the image remains steady when the operator selects other fields within a group. Special cases: <ul style="list-style-type: none"> <li>– <b>Table field grouping:</b> If table fields are grouped, Completion tries to display all image zones in the current table row simultaneously.</li> <li>– <b>Zoomed image:</b> If the image is zoomed, Completion tries to display as many zones in the group as possible, including the current zone. When there is not enough space to display the entire group, priority is given in the following order:</li> </ul> </li> </ul>

Property	Description
	<ol style="list-style-type: none"> <li>1. display the highlighted zone of the current field</li> <li>2. display the space to the left and above the field zone for a possible field label</li> </ol> <p>Image zoom will not be changed automatically to display more zones in a group simultaneously.</p> <ul style="list-style-type: none"> <li>– <b>Zones in multiple images:</b> If the current field in a group does not have an image zone, Completion automatically switches to the image in which the group has a zone. If a group has zones in multiple images, the <b>Extract Page</b> property (<b>Data Definition</b> section) indicates the image to display: <ul style="list-style-type: none"> <li>○ <b>First:</b> Display the first image that has a zone in the group</li> <li>○ <b>Last:</b> Display the last image that has a zone in the group</li> </ul> </li> </ul>
<b>Label</b>	The label to be displayed for the field/control.
<b>Label Position</b>	<p>The position of the label relative to the field/control. Option are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Top:</b> Default. The label appears above the field, spanned to the same width.</li> <li>• <b>Left:</b> The label appears to the left of the field, spanned to the same height.</li> <li>• <b>Hidden:</b> The label does not appear in the data entry form.</li> </ul>
<b>Label Style</b>	<p>A style for the field's label from the list of <b>predefined custom styles</b>. A style defines the extended visual properties of the label, such font family, font size, text padding, and others.</p> <p> <b>Note:</b> To set a custom style for a check box label, use the <b>Style</b> property (below).</p>

Property	Description
<b>Label Text Alignment</b>	<p>The horizontal alignment of the label text. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Left:</b> Aligned on the left, but ragged on the right.</li> <li>• <b>Right:</b> Aligned on the right, but ragged on the left.</li> <li>• <b>Center:</b> Ragged on both the right and the left.</li> <li>• <b>Justify:</b> Aligned on both the left and the right. Space between words is adjusted.</li> </ul>
<b>Label Width</b>	<p>The width of the background box that the label resides within. Specify an integer that represents the width of the background box.</p>
<b>Show In Summary</b>	<p>The field in the <b>Fields Summary</b> panel, which is displayed to the operator during production. The fields summary list is used to show the values of important fields all in one location. Each field in the list is a link that operators can click to quickly navigate to the corresponding data entry field immediately.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
<b>Style</b>	<p>A custom style that defines visual characteristics, such as a font family, font size, background and foreground colors, and line thickness. The characteristics available depend on the specific field/control. For more information, see <a href="#">“Adding custom styles” on page 53</a>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
<b>Tool Tip</b>	<p>The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator’s mouse hovers over the field/control.</p>

### 12.9.1.1 Using masking

You can additionally specify a mask for any text box of data type **Text/String** to restrict operator input.

To add a mask, enter the format string in the field's **Text Mask** property in the **Validation and Formatting** section. In particular, you can do the following:

- Set the expected number of characters in the field, that is, the maximum length or size of a field.
- Restrict the type of characters that can appear at a given position. The type can be a digit, a character (uppercase, lowercase, or both), or some predefined set of symbols.
- Display the value in the predefined format, such as a phone number format (\_\_\_\_) \_\_\_\_-\_\_\_\_.

“[Characters allowed in a mask](#)” on page 399 provides the valid characters and their meaning.

**Table 12-16: Characters allowed in a mask**

Character	Meaning	Example
0	A single digit between 0 and 9 is required.	Zip code (5 digits allowed): 00000
L	Letter, required. Restricts input to Unicode defined letters (for example, a - Z for the U.S.).	
&	Character, required. Restricts input to Unicode letters and non-control characters.	
A	Alphanumeric, required. Restricts input to Unicode letters and numbers.	
\	Escape. Escapes a mask character, turning it into a literal. \\ is the escape sequence for a backslash.	

Character	Meaning	Example
() _ -	Format characters: braces, underscore, dash. These characters are displayed in the field and cannot be deleted.	Document number format: (000)000-0000  Input: (408)234-5678 4082345678  Displayed value: (408)234-5678
All other characters	Literals. All non-mask elements will appear as themselves within a masked text box. Literals always occupy a static position in the mask at runtime, and cannot be moved or deleted by the user.	
*	Wildcard that repeats the last mask character. The scenario is for a variable-length number such as an account number. This allows the ending part of the mask to extend. This mask modifier works only when used at the end of the mask.	Document number format with a variable-length part (000)000-0*

### Characters not allowed in a mask

The following characters are reserved and cannot be used in a mask unless they are preceded by a backslash:

9 # ? C a . , : / \$ < > |



**Note:** Clients use a question mark (?) in the field to replace a low-confidence or unrecognized character from *OCR* extraction.

### How a mask affects operator input

If the field has a mask, the operator views format characters (if any used) and underscores beneath each placeholder where input is expected. The underscores can only be replaced by characters that match the format of the placeholder characters. Wrong input highlights the field in error. Format characters are not replaced by operator input. When the operator enters a character, the cursor skips the format character and goes to the next logical position in the field.

### How a mask affects automatic input

If the value is extracted or set programmatically, masking is applied as if the value is typed by the operator. If the mask defines format characters, the field displays the

formatted value. If the extracted value in the field mismatches the mask, the operator views that field highlighted as a work item. If the extracted result exceeds the number of allowed characters in the field, the value is considered valid and any overflow characters are discarded.

### How a mask affects data export

By default, format characters are displayed in the field but not saved as part of the value. To save and export a formatted value, set the **Include Placeholder Characters** option in the **Validation and Formatting** section.

#### 12.9.1.2 Standard date/time format conversions for Web Client

If you choose to use a standard format (in **Field Properties > Manual Validation > Date/Time Format** or **Number Format**), then they are converted as follows:

- Months and days of the week are displayed as numbers only; that is, they are not spelled out.
- Time zones are not displayed.
- 12-hour times are converted to 24-hour times and AM/PM indicators are not displayed.
- The following characters are treated as literal characters:
  - . (period)
  - , (comma)
  - / (forward slash)

For example, for the US English locale, the standard formats are converted as outlined in [“Standard Formats for US English locale” on page 401](#).

**Table 12-17: Standard Formats for US English locale**


Standard format	Converted format
General Date/Time	M/d/yyyy H:mm
Full Date/Time	MM dd, yyyy H:mm
Universal Sortable Date/Time	yyyy-MM-dd HH:mm:ss
Universal Full Date/Time	MM dd, yyyy H:mm:ss
Date-Short	M/d/yyyy
Date-Long	MM dd, yyyy
Time	H:mm
General	#.#####
Number	#.00



Standard format	Converted format
Fixed-Point	#.00
Percent	#.00 %

## 12.9.2 Drop-down List Box (field)

“Data definition properties” on page 402 provides a description of data definition properties for the drop-down list box.

**Table 12-18: Data definition properties**


Property	Description
Custom Value	<p>Custom text that is stored with the field or control. The text is not visible to the operator but can be retrieved with scripting.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
Data Type	<p>The type of data for the field. The data type determines each field’s <b>Validation and Formatting</b> properties. For more information, see each field’s <b>Validation and Formatting</b> properties.</p>
Extract Page	<p>The successfully extracted value to use for indexing the field for each document. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>First</b> (default): The first extracted field indexes the field.</li> <li>• <b>Last:</b> The last extracted field indexes the field.</li> </ul>



Property	Description
<p><b>Index Level</b></p>	<p>The level at which the field is to be indexed:</p> <p> <b>Note:</b> If you need to index a field above the document level for a bundle of documents of different document types, then add the same field with the same <b>Index Level</b> property to all of the document types.</p> <ul style="list-style-type: none"> <li>• If the index level is 1 (Document level), the field will be indexed for each individual document.</li> <li>• If the index level is 2 (Folder level), all documents in one folder must have the same value in this field.</li> </ul> <p> <b>Note:</b> Not applicable for table fields.</p> <p>For the Extraction module, this field is extracted from all documents in the folder and each document has its own value.</p> <p>For the Completion module, if the extracted values do not match, the last extracted value appears in the folder-level field of each document. If the operator populates this field with a different value and confirms it, this value will be propagated to all the documents in the folder. When the operator merges folders, all included documents get into the first folder and display the value specific for the first folder. Where the displayed value differs from the original one, the field requires manual confirmation.</p> <ul style="list-style-type: none"> <li>• If the index level is higher than 2 (Folder level), the field will be indexed for each individual document, but the first extracted or the last confirmed value will be stored in all documents in the index-level node.</li> </ul>
<p><b>Input Mode</b></p>	<p>One of the following options for the field:</p> <ul style="list-style-type: none"> <li>• <b>Editable/Not Required</b> (default)</li> <li>• <b>Editable/Required</b></li> <li>• <b>Read-Only:</b> Operator input is not allowed</li> <li>• <b>Set by Expression:</b> Not available for fields that are in a table.</li> </ul>




Property	Description
<p><b>Name</b></p>	<p>A name that identifies the field or control. Field names are not displayed to the operator. You can use a field name to reference a field in an expression or document type scripting. A valid field name is as follows:</p> <ul style="list-style-type: none"> <li>• Contains alphanumeric characters from the ASCII range only.</li> <li>• Starts with a letter.</li> <li>• Does not contain spaces.</li> <li>• Is a valid variable name.</li> <li>• Is unique in the list of fields in the document type.</li> </ul>
<p><b>Value</b></p>	<p>An <b>expression</b> that initializes the field when the field's input mode is set to <b>Set by Expression</b>; otherwise, leave this property blank.</p> <p>Type in the property, or click the button and build the expression in the <b>Expressions Editor</b> dialog box.</p>


The following table provides a description of validation and formatting properties for the drop-down list box.

**Table 12-19: Validation and formatting properties**

Property	Description
Autosuggest	<p>As a user types in a string, the first value in the drop-down list that matches the string is selected. Matching criteria is as follows:</p> <ul style="list-style-type: none"> <li>• <b>Off</b> A value has a matching string at the beginning. The drop-down list of values does not automatically expand.</li> <li>• <b>Containing</b> The value has a matching string anywhere. The drop-down list automatically expands and only matching values are displayed.</li> <li>• <b>Beginning</b> The value has a matching string at the beginning. The drop-down list automatically expands and only matching values are displayed.</li> <li>• <b>All Choices</b> A value has a matching string at the beginning. The drop-down list of values automatically expands.</li> </ul> <p> <b>Note:</b> The behavior is browser-based for Web Client.</p>

Property	Description
Choice List	<p>The list of items from which the operator can choose a value. Add one value per line, pressing <b>ENTER</b> to start a new line. For example:</p> <pre>PrePaid PostPaid</pre> <p>To specify a default value, use the following syntax:</p> <pre>&lt;defaultchoice1&gt;() &lt;choice2&gt;() ... &lt;choiceN&gt;()</pre> <p>where <code>&lt;defaultchoice1&gt;</code> is the default.</p> <p> <b>Note:</b> The <code>&lt;ChoiceName&gt;</code> (<code>&lt;ChoiceValue&gt;</code>) syntax cannot be used when specifying the default value.</p> <p>Values must comply with the data type specified for the field:</p> <ul style="list-style-type: none"> <li>• <b>Date:</b> Use canonical date time values with hours, minutes, and seconds. For example, specify 12/12/2012 as (2012-12-12 00:00:00).</li> <li>• <b>Boolean:</b> Use only <code>true</code> and <code>false</code> in lowercase</li> </ul> <p>Optionally, you can use the</p> <pre>&lt;ChoiceName&gt; (&lt;ChoiceValue&gt;)</pre> <p>format for all items in the choice list. <code>&lt;ChoiceName&gt;</code> is the user-facing text and <code>&lt;ChoiceValue&gt;</code> is the code-facing value. For example:</p> <pre>PrePaid(1) PostPaid(2)</pre> <p>The choice list items (both values) can be used in expressions and in document type scripting.</p> <hr/> <p> <b>Note:</b> The data captured using <i>OCR</i> or set programmatically must match the values specified in the choice list <i>exactly</i>, including case. Otherwise,</p>

Property	Description
	Completion displays an error when the operator moves focus to the field.
<b>Control Type</b>	The type of UI control to display on the form.
<b>Date Range</b>	<p>Range of dates as follows:</p> <ul style="list-style-type: none"> <li>• Specific dates</li> <li>• An integer that represents the number of minutes after the time of data entry.</li> </ul> <p> <b>Note:</b> Blank or empty dates are not allowed. An empty date defaults to 1900-01-01.</p>
<b>Manual Confirmation</b>	<p>Operator must confirm the field's value by pressing <b>ENTER</b> or clicking a check mark next to the field.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>– For multi-line text fields, pressing <b>ENTER</b> moves the cursor to the next line instead of confirming the value.</li> <li>– If a calculation can be performed for the field, then pressing <b>ENTER</b> the first time performs the calculation and then pressing <b>ENTER</b> again confirms the calculated value.</li> </ul> <p>See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Do Not Confirm</b> (default): The extracted field is never highlighted as a work item.</li> <li>• <b>Always Confirm:</b> The field is highlighted as a work item until confirmed manually.</li> <li>• <b>Confirm On Blank:</b> If blank, the field is highlighted as a work item until confirmed manually.</li> </ul> <p> <b>Note:</b> If the list is expanded, then pressing <b>ENTER</b> the first time selects a value in the list and then pressing <b>ENTER</b> again confirms the selected value.</p>

Property	Description
<p><b>Number Range</b></p>	<p>The minimum and maximum numbers that are allowed in the field. See the following:</p> <ul style="list-style-type: none"> <li>• For <b>Validation Type</b> set to <b>Amount</b>, the <b>Number Range</b> should include a currency sign; for example, 5.12\$.</li> <li>• For <b>Validation Type</b> set to <b>Percent</b>, the <b>Number Range</b> should include a percent sign; for example, 5%.</li> </ul> <p> <b>Note:</b> Blank or empty numbers are not allowed. An empty number defaults to zero.</p>
<p><b>Size Range</b></p>	<p>The minimum and maximum number of characters that are allowed in the field.</p>

The following table provides a description of visual properties for the drop-down list box.

**Table 12-20: Visual properties**



<p><b>Label</b></p>	<p>The label to be displayed for the field/control.</p>
<p><b>Label Position</b></p>	<p>The position of the label relative to the field/control. Option are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Top:</b> Default. The label appears above the field, spanned to the same width.</li> <li>• <b>Left:</b> The label appears to the left of the field, spanned to the same height.</li> <li>• <b>Hidden:</b> The label does not appear in the data entry form.</li> </ul>
<p><b>Tool Tip</b></p>	<p>The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator’s mouse hovers over the field/control.</p>

### 12.9.3 List Box (field)

“Data definition properties” on page 409 provides a description of data definition properties for the list box.

**Table 12-21: Data definition properties**



Property	Description
Data Type	The type of data for the field. The data type determines each field's <b>Validation and Formatting</b> properties. For more information, see each field's <b>Validation and Formatting</b> properties.
Extract Page	The successfully extracted value to use for indexing the field for each document. See the following options: <ul style="list-style-type: none"><li>• <b>First</b> (default): The first extracted field indexes the field.</li><li>• <b>Last</b>: The last extracted field indexes the field.</li></ul>


Property	Description
<p><b>Index Level</b></p>	<p>The level at which the field is to be indexed:</p> <p> <b>Note:</b> If you need to index a field above the document level for a bundle of documents of different document types, then add the same field with the same <b>Index Level</b> property to all of the document types:</p> <ul style="list-style-type: none"> <li>• If the index level is 1 (Document level), the field will be indexed for each individual document.</li> <li>• If the index level is 2 (Folder level), all documents in one folder must have the same value in this field.</li> </ul> <p> <b>Note:</b> Not applicable for table fields.</p> <p>For the Extraction module, this field is extracted from all documents in the folder and each document has its own value.</p> <p>For the Completion module, if the extracted values do not match, the last extracted value appears in the folder-level field of each document. If the operator populates this field with a different value and confirms it, this value will be propagated to all the documents in the folder. When the operator merges folders, all included documents get into the first folder and display the value specific for the first folder. Where the displayed value differs from the original one, the field requires manual confirmation.</p> <ul style="list-style-type: none"> <li>• If the index level is higher than 2 (Folder level), the field will be indexed for each individual document, but the first extracted or the last confirmed value will be stored in all documents in the index-level node.</li> </ul>
<p><b>Input Mode</b></p>	<p>One of the following options for the field:</p> <ul style="list-style-type: none"> <li>• <b>Editable/Not Required</b> (default)</li> <li>• <b>Editable/Required</b></li> <li>• <b>Read-Only:</b> Operator input is not allowed</li> <li>• <b>Set by Expression:</b> Not available for fields that are in a table.</li> </ul>

Property	Description
<b>Name</b>	<p>A name that identifies the field or control. Field names are not displayed to the operator. You can use a field name to reference a field in an expression or document type scripting. A valid field name is as follows:</p> <ul style="list-style-type: none"> <li>• Contains alphanumeric characters from the ASCII range only.</li> <li>• Starts with a letter.</li> <li>• Does not contain spaces.</li> <li>• Is a valid variable name.</li> <li>• Is unique in the list of fields in the document type.</li> </ul>
<b>Value</b>	<p>An <b>expression</b> that initializes the field when the field's input mode is set to <b>Set by Expression</b>; otherwise, leave this property blank.</p> <p>Type in the property, or click the button and build the expression in the <b>Expressions Editor</b> dialog box.</p>

The following table provides a description of validation and formatting properties for the list box.

**Table 12-22: Validation and formatting properties**

Property	Description
Choice List	<p>The list of items from which the operator can choose a value. Add one value per line, pressing <b>ENTER</b> to start a new line. For example:</p> <pre>PrePaid PostPaid</pre> <p>To specify a default value, use the following syntax:</p> <pre>&lt;defaultchoice1&gt;() &lt;choice2&gt;() ... &lt;choiceN&gt;()</pre> <p>where <code>&lt;defaultchoice1&gt;</code> is the default.</p> <p> <b>Note:</b> The <code>&lt;ChoiceName&gt;</code> (<code>&lt;ChoiceValue&gt;</code>) syntax cannot be used when specifying the default value.</p> <p>Values must comply with the data type specified for the field:</p> <ul style="list-style-type: none"> <li>• <b>Date:</b> Use canonical date time values with hours, minutes, and seconds. For example, specify 12/12/2012 as (2012-12-12 00:00:00).</li> <li>• <b>Boolean:</b> Use only <code>true</code> and <code>false</code> in lowercase</li> </ul> <p>Optionally, you can use the</p> <pre>&lt;ChoiceName&gt; (&lt;ChoiceValue&gt;)</pre> <p>format for all items in the choice list. <code>&lt;ChoiceName&gt;</code> is the user-facing text and <code>&lt;ChoiceValue&gt;</code> is the code-facing value. For example:</p> <pre>PrePaid(1) PostPaid(2)</pre> <p>The choice list items (both values) can be used in expressions and in document type scripting.</p> <p> <b>Note:</b> The data captured using <i>OCR</i> or set programmatically must match the values specified in the choice list <i>exactly</i>, including case. Otherwise,</p>

Property	Description
	Completion displays an error when the operator moves focus to the field.
<b>Control Type</b>	The type of UI control to display on the form.
<b>Manual Confirmation</b>	<p>Operator must confirm the field's value by pressing <b>ENTER</b> or clicking a check mark next to the field.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>– For multi-line text fields, pressing <b>ENTER</b> moves the cursor to the next line instead of confirming the value.</li> <li>– If a calculation can be performed for the field, then pressing <b>ENTER</b> the first time performs the calculation and then pressing <b>ENTER</b> again confirms the calculated value.</li> </ul> <p>See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Do Not Confirm</b> (default): The extracted field is never highlighted as a work item.</li> <li>• <b>Always Confirm</b>: The field is highlighted as a work item until confirmed manually.</li> <li>• <b>Confirm On Blank</b>: If blank, the field is highlighted as a work item until confirmed manually.</li> </ul>

The following table provides a description of visual properties for the list box.

**Table 12-23: Visual properties**

<b>Label</b>	The label to be displayed for the field/control.
<b>Label Position</b>	<p>The position of the label relative to the field/control. Option are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Top</b>: Default. The label appears above the field, spanned to the same width.</li> <li>• <b>Left</b>: The label appears to the left of the field, spanned to the same height.</li> <li>• <b>Hidden</b>: The label does not appear in the data entry form.</li> </ul>

<b>Tool Tip</b>	The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator’s mouse hovers over the field/control.
-----------------	---



## 12.9.4 Combo Box (field)

 **Note:** Not supported for Web Client.

“Data Definition Properties” on page 414 provides a description of data definition properties for the combo box.

**Table 12-24: Data Definition Properties**


Property	Description
Data Type	The type of data for the field. The data type determines each field’s <b>Validation and Formatting</b> properties. For more information, see each field’s <b>Validation and Formatting</b> properties.
Extract Page	The successfully extracted value to use for indexing the field for each document. See the following options: <ul style="list-style-type: none"> <li>• <b>First</b> (default): The first extracted field indexes the field.</li> <li>• <b>Last</b>: The last extracted field indexes the field.</li> </ul>



Property	Description
<p><b>Index Level</b></p>	<p>The level at which the field is to be indexed:</p> <p> <b>Note:</b> If you need to index a field above the document level for a bundle of documents of different document types, then add the same field with the same <b>Index Level</b> property to all of the document types:</p> <ul style="list-style-type: none"> <li>• If the index level is 1 (Document level), the field will be indexed for each individual document.</li> <li>• If the index level is 2 (Folder level), all documents in one folder must have the same value in this field.</li> </ul> <p> <b>Note:</b> Not applicable for table fields.</p> <p>For the Extraction module, this field is extracted from all documents in the folder and each document has its own value.</p> <p>For the Completion module, if the extracted values do not match, the last extracted value appears in the folder-level field of each document. If the operator populates this field with a different value and confirms it, this value will be propagated to all the documents in the folder. When the operator merges folders, all included documents get into the first folder and display the value specific for the first folder. Where the displayed value differs from the original one, the field requires manual confirmation.</p> <ul style="list-style-type: none"> <li>• If the index level is higher than 2 (Folder level), the field will be indexed for each individual document, but the first extracted or the last confirmed value will be stored in all documents in the index-level node.</li> </ul>
<p><b>Input Mode</b></p>	<p>One of the following options for the field:</p> <ul style="list-style-type: none"> <li>• <b>Editable/Not Required</b> (default)</li> <li>• <b>Editable/Required</b></li> <li>• <b>Read-Only:</b> Operator input is not allowed</li> <li>• <b>Set by Expression:</b> Not available for fields that are in a table.</li> </ul>




Property	Description
<p><b>Name</b></p>	<p>A name that identifies the field or control. Field names are not displayed to the operator. You can use a field name to reference a field in an expression or document type scripting. A valid field name is as follows:</p> <ul style="list-style-type: none"> <li>• Contains alphanumeric characters from the ASCII range only.</li> <li>• Starts with a letter.</li> <li>• Does not contain spaces.</li> <li>• Is a valid variable name.</li> <li>• Is unique in the list of fields in the document type.</li> </ul>
<p><b>Value</b></p>	<p>An <b>expression</b> that initializes the field when the field's input mode is set to <b>Set by Expression</b>; otherwise, leave this property blank.</p> <p>Type in the property, or click the button and build the expression in the <b>Expressions Editor</b> dialog box.</p>


The following table provides a description of validation and formatting properties for the combo box.

**Table 12-25: Validation and formatting properties**

Property	Description
Autosuggest	<p>As a user types in a string, the first value in the drop-down list that matches the string is selected. Matching criteria is as follows:</p> <ul style="list-style-type: none"> <li>• <b>Off</b> A value has a matching string at the beginning. The drop-down list of values does not automatically expand.</li> <li>• <b>Containing</b> The value has a matching string anywhere. The drop-down list automatically expands and only matching values are displayed.</li> <li>• <b>Beginning</b> The value has a matching string at the beginning. The drop-down list automatically expands and only matching values are displayed.</li> <li>• <b>All Choices</b> A value has a matching string at the beginning. The drop-down list of values automatically expands.</li> </ul> <p> <b>Note:</b> The behavior is browser-based for Web Client.</p>

Property	Description
<p><b>Choice List</b></p>	<p>The list of items from which the operator can choose a value. Add one value per line, pressing <b>ENTER</b> to start a new line. For example:</p> <pre>PrePaid PostPaid</pre> <p>To specify a default value, use the following syntax:</p> <pre>&lt;defaultchoice1&gt;() &lt;choice2&gt;() ... &lt;choiceN&gt;()</pre> <p>where <code>&lt;defaultchoice1&gt;</code> is the default.</p> <p> <b>Note:</b> The <code>&lt;ChoiceName&gt;</code> (<code>&lt;ChoiceValue&gt;</code>) syntax cannot be used when specifying the default value.</p> <p>Values must comply with the data type specified for the field:</p> <ul style="list-style-type: none"> <li>• <b>Date:</b> Use canonical date time values with hours, minutes, and seconds. For example, specify 12/12/2012 as (2012-12-12 00:00:00).</li> <li>• <b>Boolean:</b> Use only <code>true</code> and <code>false</code> in lowercase</li> </ul> <p>Optionally, you can use the</p> <pre>&lt;ChoiceName&gt; (&lt;ChoiceValue&gt;)</pre> <p>format for all items in the choice list. <code>&lt;ChoiceName&gt;</code> is the user-facing text and <code>&lt;ChoiceValue&gt;</code> is the code-facing value. For example:</p> <pre>PrePaid(1) PostPaid(2)</pre> <p>The choice list items (both values) can be used in expressions and in document type scripting.</p> <hr/> <p> <b>Note:</b> The data captured using <i>OCR</i> or set programmatically must match the values specified in the choice list <i>exactly</i>, including case. Otherwise,</p>

Property	Description
	Completion displays an error when the operator moves focus to the field.
<b>Control Type</b>	The type of UI control to display on the form.
<b>Date Range</b>	<p>Range of dates as follows:</p> <ul style="list-style-type: none"> <li>• Specific dates</li> <li>• An integer that represents the number of minutes after the time of data entry.</li> </ul> <p> <b>Note:</b> Blank or empty dates are not allowed. An empty date defaults to 1900-01-01.</p>
<b>Manual Confirmation</b>	<p>Operator must confirm the field's value by pressing <b>ENTER</b> or clicking a check mark next to the field.</p> <p> <b>Notes</b></p> <p>For more information, see the following:</p> <ul style="list-style-type: none"> <li>– For multi-line text fields, pressing <b>ENTER</b> moves the cursor to the next line instead of confirming the value.</li> <li>– If a calculation can be performed for the field, then pressing <b>ENTER</b> the first time performs the calculation and then pressing <b>ENTER</b> again confirms the calculated value.</li> </ul> <p>See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Do Not Confirm</b> (default): The extracted field is never highlighted as a work item.</li> <li>• <b>Always Confirm:</b> The field is highlighted as a work item until confirmed manually.</li> <li>• <b>Confirm On Blank:</b> If blank, the field is highlighted as a work item until confirmed manually.</li> </ul> <p> <b>Note:</b> If the list is expanded, then pressing <b>ENTER</b> the first time selects a value in the list and then pressing <b>ENTER</b> again confirms the selected value.</p>

Property	Description
<b>Number Range</b>	<p>The minimum and maximum numbers that are allowed in the field. See the following:</p> <ul style="list-style-type: none"> <li>• For <b>Validation Type</b> set to <b>Amount</b>, the <b>Number Range</b> should include a currency sign; for example, 5.12\$.</li> <li>• For <b>Validation Type</b> set to <b>Percent</b>, the <b>Number Range</b> should include a percent sign; for example, 5%.</li> </ul> <p> <b>Note:</b> Blank or empty numbers are not allowed. An empty number defaults to zero.</p>
<b>Size Range</b>	<p>The minimum and maximum number of characters that are allowed in the field.</p>

The following table provides a description of visual properties for the combo box.

**Table 12-26: Visual properties**



<b>Label</b>	<p>The label to be displayed for the field/control.</p>
<b>Label Position</b>	<p>The position of the label relative to the field/control. Option are as follows:</p> <ul style="list-style-type: none"> <li>• <b>Top:</b> Default. The label appears above the field, spanned to the same width.</li> <li>• <b>Left:</b> The label appears to the left of the field, spanned to the same height.</li> <li>• <b>Hidden:</b> The label does not appear in the data entry form.</li> </ul>
<b>Tool Tip</b>	<p>The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator's mouse hovers over the field/control.</p>

## 12.9.5 Check Box (field)

“Data Definition Properties” on page 421 provides a description of data definition properties for the check box.

**Table 12-27: Data Definition Properties**

Property	Description
Data Type	The type of data for the field. The data type determines each field's <b>Validation and Formatting</b> properties. For more information, see each field's <b>Validation and Formatting</b> properties.
Extract Page	The successfully extracted value to use for indexing the field for each document. See the following options: <ul data-bbox="964 825 1430 949" style="list-style-type: none"><li>• <b>First</b> (default): The first extracted field indexes the field.</li><li>• <b>Last</b>: The last extracted field indexes the field.</li></ul>

Property	Description
<p><b>Index Level</b></p>	<p>The level at which the field is to be indexed:</p> <p> <b>Note:</b> If you need to index a field above the document level for a bundle of documents of different document types, then add the same field with the same <b>Index Level</b> property to all of the document types:</p> <ul style="list-style-type: none"> <li>• If the index level is 1 (Document level), the field will be indexed for each individual document.</li> <li>• If the index level is 2 (Folder level), all documents in one folder must have the same value in this field.</li> </ul> <p> <b>Note:</b> Not applicable for table fields.</p> <p>For the Extraction module, this field is extracted from all documents in the folder and each document has its own value.</p> <p>For the Completion module, if the extracted values do not match, the last extracted value appears in the folder-level field of each document. If the operator populates this field with a different value and confirms it, this value will be propagated to all the documents in the folder. When the operator merges folders, all included documents get into the first folder and display the value specific for the first folder. Where the displayed value differs from the original one, the field requires manual confirmation.</p> <ul style="list-style-type: none"> <li>• If the index level is higher than 2 (Folder level), the field will be indexed for each individual document, but the first extracted or the last confirmed value will be stored in all documents in the index-level node.</li> </ul>
<p><b>Input Mode</b></p>	<p>One of the following options for the field:</p> <ul style="list-style-type: none"> <li>• <b>Editable/Not Required</b> (default)</li> <li>• <b>Editable/Required</b></li> <li>• <b>Read-Only:</b> Operator input is not allowed</li> <li>• <b>Set by Expression:</b> Not available for fields that are in a table.</li> </ul>

Property	Description
Name	<p>A name that identifies the field or control. Field names are not displayed to the operator. You can use a field name to reference a field in an expression or document type scripting. A valid field name is as follows:</p> <ul style="list-style-type: none"> <li>• Contains alphanumeric characters from the ASCII range only.</li> <li>• Starts with a letter.</li> <li>• Does not contain spaces.</li> <li>• Is a valid variable name.</li> <li>• Is unique in the list of fields in the document type.</li> </ul>
Value	<p>An <b>expression</b> that initializes the field when the field's input mode is set to <b>Set by Expression</b>; otherwise, leave this property blank.</p> <p>Type in the property, or click the button and build the expression in the <b>Expressions Editor</b> dialog box.</p>



### Notes

For more information, see the following:

- **Check Box of Boolean type:** If you select the Boolean type check box and leave the *Value List* field empty, True or False is displayed in the UimData based on the checked box True and unchecked box False. You cannot enter 1 or 0 in the *Value List* setting. Enter False or True to get an inverted result for the check box.
- **Check Box of String/Text type:** If the *Value List* field is blank, you get checked box 1 or unchecked box 0 in the UimData value. In the *Value List* setting, if you enter True or False, you get True or False. If you enter any other text in the *Value List* field (for example, Red and Blue), you will get checked box Red or unchecked box Blue in the UimData.

## Validation and formatting properties

### Control Type

The type of UI control to display on the form.

### Manual Confirmation

Operator must confirm the field's value by pressing **ENTER** or clicking a check mark next to the field.



**Notes**

For more information, see the following:

- For multi-line text fields, pressing **ENTER** moves the cursor to the next line instead of confirming the value.
- If a calculation can be performed for the field, then pressing **ENTER** the first time performs the calculation and then pressing **ENTER** again confirms the calculated value.

See the following options:

- **Do Not Confirm** (default): The extracted field is never highlighted as a work item.
- **Always Confirm**: The field is highlighted as a work item until confirmed manually.
- **Confirm On Blank**: If blank, the field is highlighted as a work item until confirmed manually.

---

**Value List**

The list of values for a check box if you want to save descriptive text as the field's value instead of the default **1** (selected) or **0** (not selected).

For example, you can define a check box on the form with the **Married** label and the following list of values:

MARRIED  
SINGLE

Then the checked box will save the value **MARRIED** as the field's value (instead of 1), and the unchecked box will save the value **SINGLE** (instead of 0).

To define the list of values, enter the value for the checked state in the first line and the value for the unchecked state in the second line. Use **ENTER** to start a new line.

Values must be compatible with the data type specified for the field. For dates, use a canonical date time value with hours, minutes, and seconds. For example, specify 12/12/2012 as 2012-12-12 00:00:00

---


The following table provides a description of visual properties for the check box.

**Table 12-28: Visual properties**

<b>Label</b>	The label to be displayed for the field/control.
<b>Tool Tip</b>	The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator's mouse hovers over the field/control.



## 12.9.6 Text Block (control)

Displays text on the form.

 **Note:** Not supported for Web Client.


“Data definition properties” on page 425 provides a description of data definition properties for the text block.

**Table 12-29: Data definition properties**

Property	Description
Name	<p>The name that identifies the field or control. Must be a valid variable name without spaces. Must be unique in the list of fields in the document type, contain alphanumeric characters from the ASCII range, and start with a letter.</p> <p> <b>Note:</b> The string Barcode &lt;#&gt;, where &lt;#&gt; is an integer of any length, is reserved for use with PixTools for Web. Therefore, if this document type field also uses population rules, then, even if the <b>Capture</b> profile <b>Detect Barcodes</b> option is disabled, PixTools for Web is used and overwrites the server-side value with a blank value.</p>
Custom Value	<p>Custom text that is stored with the field or control. The text is not visible to the operator but can be retrieved with scripting.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>


The following table provides a description of visual properties for the text block.

**Table 12-30: Visual properties**

Property	Description
Alignment	<p>The horizontal alignment of text that is displayed in the field. See the following options:</p> <ul style="list-style-type: none"> <li>• <b>Left:</b> Aligned on the left, but ragged on the right.</li> <li>• <b>Right:</b> Aligned on the right, but ragged on the left.</li> <li>• <b>Center:</b> Ragged on both the right and the left.</li> <li>• <b>Justify:</b> Aligned on both the left and the right. Space between words is adjusted.</li> </ul>
Style	<p>A custom style that defines visual characteristics, such as a font family, font size, background and foreground colors, and line thickness. The characteristics available depend on the specific field/control. For more information, see <a href="#">“Adding custom styles” on page 53</a>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
Text	<p>The text to be displayed on the data entry form.</p>
Tool Tip	<p>The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator’s mouse hovers over the field/control.</p>



### 12.9.7 Group Box (control)

Displays a box around a group of fields and controls and a title for the group on the form.

 **Note:** Not supported for Web Client.


[“Data definition properties” on page 427](#) provides a description of the data definition properties for the group box.

**Table 12-31: Data definition properties**

Property	Description
<b>Name</b>	<p>The name that identifies the field or control. Must be a valid variable name without spaces. Must be unique in the list of fields in the document type, contain alphanumeric characters from the ASCII range, and start with a letter.</p> <p> <b>Note:</b> The string <b>Barcode &lt;#&gt;</b>, where &lt;#&gt; is an integer of any length, is reserved for use with PixTools for Web. Therefore, if this document type field also uses population rules, then, even if the <b>Capture</b> profile <b>Detect Barcodes</b> option is disabled, PixTools for Web is used and overwrites the server-side value with a blank value.</p>
<b>Custom Value</b>	<p>Custom text that is stored with the field or control. The text is not visible to the operator but can be retrieved with scripting.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

The following table provides a description of the visual properties for the group box.

**Table 12-32: Visual properties**

Property	Description
<b>Group Box Name</b>	The name for the group box to be displayed on the data entry form.
<b>Style</b>	<p>A custom style that defines visual characteristics, such as a font family, font size, background and foreground colors, and line thickness. The characteristics available depend on the specific field/control. For more information, see <a href="#">“Adding custom styles” on page 53</a>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

Property	Description
Tool Tip	The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator's mouse hovers over the field/control.

## 12.9.8 Button (control)


Displays a button on the form. The button is called by a button click event handler in document type scripting.


You hide a button by using the `ShowControl` property. Thus, it could be used as a shortcut key in Identification (which does not display controls) and Completion. If you implement custom logic in the `ButtonClick` event handler and specify a shortcut for the button in document type scripting, the operator working in the main window can use this shortcut to fire the `ButtonClick` event and execute the operation. For an example, see *Emc.InputAccel.CaptureClient API Reference Guide > IUIimFormControlContext.ShowControl Method*.

For more information, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

“Data definition properties” on page 428 provides a description of data definition properties for buttons.


**Table 12-33: Data definition properties**

Property	Description
Name	<p>The name that identifies the field or control. Must be a valid variable name without spaces. Must be unique in the list of fields in the document type, contain alphanumeric characters from the ASCII range, and start with a letter.</p> <p> <b>Note:</b> The string <code>Barcode &lt;#&gt;</code>, where <code>&lt;#&gt;</code> is an integer of any length, is reserved for use with PixTools for Web. Therefore, if this document type field also uses population rules, then, even if the <b>Capture</b> profile <b>Detect Barcodes</b> option is disabled, PixTools for Web is used and overwrites the server-side value with a blank value.</p>

Property	Description
Custom Value	<p>Custom text that is stored with the field or control. The text is not visible to the operator but can be retrieved with scripting.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

The following table provides a description of visual properties for buttons.

**Table 12-34: Visual properties**

Property	Description
Image	<p>Select an icon image to be displayed on the button.</p> <p>Valid file formats are PNG, GIF, and JPG. The image is scaled to control the size.</p>
Label	The label to be displayed for the field/control.
Shortcut	Shortcut keystroke for this button. Blank, if shortcut is not required.
Style	<p>A custom style that defines visual characteristics, such as a font family, font size, background and foreground colors, and line thickness. The characteristics available depend on the specific field/control. For more information, see <i>“Adding custom styles” on page 53</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
Tool Tip	The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator’s mouse hovers over the field/control.



## 12.9.9 Image (control)

Displays an image on the form.

 **Note:** Not supported for Web Client.

“Data definition properties” on page 430 provides a description of data definition properties for images.


**Table 12-35: Data definition properties**

Property	Description
Name	<p>The name that identifies the field or control. Must be a valid variable name without spaces. Must be unique in the list of fields in the document type, contain alphanumeric characters from the ASCII range, and start with a letter.</p> <p> <b>Note:</b> The string Barcode &lt;#&gt;, where &lt;#&gt; is an integer of any length, is reserved for use with PixTools for Web. Therefore, if this document type field also uses population rules, then, even if the <b>Capture</b> profile <b>Detect Barcodes</b> option is disabled, PixTools for Web is used and overwrites the server-side value with a blank value.</p>
Custom Value	<p>Custom text that is stored with the field or control. The text is not visible to the operator but can be retrieved with scripting.</p> <p>For more information about creating scripts and using custom script parameters, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>

The following table provides a description of data definition properties for images.

**Table 12-36: Visual properties**


Property	Description
Image	<p>The image to be displayed on the form, such as a logo.</p> <p>Valid file formats are PNG, GIF, and JPG. The image is scaled to control the size.</p>

Property	Description
Style	<p>A custom style that defines visual characteristics, such as a font family, font size, background and foreground colors, and line thickness. The characteristics available depend on the specific field/control. For more information, see <a href="#">“Adding custom styles” on page 53</a>.</p> <p> <b>Note:</b> Not supported in Web Client.</p>
Tool Tip	<p>The text that helps operators determine what value should be entered in this field or how the control should be used. The tooltip displays when the operator’s mouse hovers over the field/control.</p>

## 12.9.10 Table

A table consists of fields and each field represents a column with the field label as the column header. Controls are not allowed. Field labels configured as hidden are still displayed. At runtime, the table bar is not displayed on the form. The table name can be used to reference the table in document type scripting.

 **Note:** An Advanced Recognition license is required to work with tables in the Document Type Editor.

 **Note:** In Web Client, the number of rows to display is fixed; a scroll bar is used when the number of rows exceeds this fixed number.

In Completion, tables can be displayed in either **Grid View** or **Record View**. To configure one of these views, select **Grid View** or **Record View** in the upper right corner of the **Form** tab. See the following options:

- **Grid View**

The table is displayed in the usual tabular format. All types of fields look and behave as edit boxes. For example, drop-down lists cannot be expanded and the list box is not completely displayed. Also, if a list box is populated with the *ChoiceName(ChoiceValue)* pairs from a choice list, the *ChoiceName* value is displayed but the *ChoiceValue* must be manually entered.

You could also change the field label by double-clicking it on the form or selecting it and pressing **F2**.

- **Record View**

Records are displayed as widely spaced rows separated by horizontal lines and large blank vertical space. All types of fields behave as expected. In particular,

you can view full lists with all items, expand the drop-down lists, and the label position property is applied correctly.

## Sections

When you move a field right below the table, a section bar is automatically added between them. A section indicates the end of the array of table fields. Sections are not displayed to the operator in the data entry form.

Sections are not displayed in the fields panel. You cannot add a section or rename it. However, you can use the section name to reference it in the document type scripting, for instance, to apply a group operation to all fields and controls that belong to the section. The section ends where a new table or segment starts.

## Chapter 13

# Deploying a component

This section introduces the **Deployment** tab's **System** area, explains its intended use, and provides description of the layout and options available on this tab.

### 13.1 How to use deployment

For all changes you make to profiles, document types, queries, styles, CaptureFlows, Database Connections, Email Connections, and other service components, Designer enables pushing changes to the server, *uploading* these entities from the developer's machine to the specific capture server. Once you have created a service component, it is *saved* locally to the Designer folder, and you can *upload* it to the selected server. See [“Uploading or deploying service components including profiles and CaptureFlows” on page 439](#).

Alternatively, you can use the command line utility, `<intelligent_capture_install>\Client\binnt\DeploymentUtility.exe`. You can also use this utility to download service components (except for CaptureFlows and profile scripts) from an Intelligent Capture Server to your local machine and then open them in Designer for debugging. For more information, see [the procedure of deploying CaptureFlows, Profile Scripts, and Data Files to the server using the Command Line](#).



**Note:** To access the **Deployment** tab, you need to have a capture system and a connection to a capture server. For detailed information, see [“Defining capture servers” on page 32](#) section.

Every component has a current *status* displayed on the **Deployment** tab. This allows you to see the differences between what is actual for the selected capture system and what has been changed. For detailed example showing the role of statuses and for the information on filtering service components by their statuses, see [“Filtering service components list” on page 440](#) section.



**Note:** When using test and production environments, you first upload the local version of the component to the test capture server. After its validation in the test environment you can upload the correct local version to the production server.

You can upload service components when Designer is connected to a single capture server or to the ScaleServer group. In the latter case, service components will be uploaded to all servers which belong to the ScaleServer group.

## 13.2 Understanding Deployment tab elements

The **Deployment** tab displays the current state of the service components you have previously created or modified and provides options for **managing** service components.


To access **Deployment**, navigate to **System** and select the **Deployment** tab.



**Note:** The **Deployment** tab is dimmed if you are not connected to a capture server.




“**Deployment tab elements**” on page 435 describes the layout and options available on the **Deployment** tab.


**Table 13-1: Deployment tab elements**

Element	Description
<b>Service Component</b>	<p>The column shows the content existing in local sources and on the capture server. Service components are shown in the following categories:</p> <ul style="list-style-type: none"> <li>• <b>Configuration Settings - System Styles</b></li> <li>• <b>Configuration Settings - Custom Styles</b></li> <li>• <b>Configuration Settings - Shortcuts</b></li> <li>• <b>Configuration Settings - Deployment Files</b></li> <li>• <b>Configuration Settings - Global Options</b></li> <li>• <b>Database Connection</b></li> <li>• <b>Documentum Connection</b></li> <li>• <b>Email Connection</b></li> <li>• <b>Queries</b></li> <li>• <b>Image Processing Profiles</b></li> <li>• <b>Image Conversion Profiles</b></li> <li>• <b>Document Types</b></li> <li>• <b>Document Resources</b></li> <li>• <b>Import Profiles</b></li> <li>• <b>Export Profiles</b></li> <li>• <b>Standard OCR Profiles</b></li> <li>• <b>CaptureFlows</b></li> </ul> <p>To select a service component you want to operate with, use the corresponding check box to the left of the component name.</p> <p> <b>Note:</b> Recognition projects are not displayed in <b>Deployment</b> tab.</p>
<b>Last Saved</b>	<p>Shows time when the last modification of the service component was saved in the currently selected capture system.</p>

Element	Description
<b>Status</b>	<p>Shows actual status of the service component for the current capture system. Each item in the service components list has one of the following statuses:</p> <ul style="list-style-type: none"> <li>• <b>Unchanged:</b> the item has equal versions both locally, in the currently selected capture system, and on server.</li> <li>• <b>Local Changed:</b> either the item is new and has never been uploaded to the server, or the item is modified locally and the changes are not uploaded to the server.</li> <li>• <b>Server Changed:</b> the server version of the item is newer than its local version in the currently selected capture system.</li> <li>• <b>Not in Local Source:</b> the item is on server only. Note that the deleted items in this status cannot be restored.</li> <li>• <b>Not in server source:</b> To view service components on the local system only.</li> </ul>
<b>Show</b>	<p>The filter allows you to show or hide service components in the list with a defined <b>Status</b>. To display service components with a particular status, select the corresponding check box at the top of the service components list.</p>
<b>Refresh list</b>	<p>Use the button at the upper right of the service components list to refresh the list of service components with server information. When clicked, all modified service components (if any) within the current capture system are saved, and the status of the items is automatically updated.</p>

Element	Description
<p><b>Upload</b></p>	<p>Use the button at the bottom of the service components list to upload items to the connected capture server. When clicked, the <i>selected</i> service components are saved, and the status of the items is automatically updated:</p> <ul style="list-style-type: none"> <li>• If local version of the component selected for uploading is newer than its server version or both versions are equal, the uploading is performed without notifications.</li> <li>• If local version of the component selected for uploading is older than its server version, you will be prompt to confirm the necessity to overwrite the server version.</li> <li>• Each time that a server connection is established, Database and Email connections that are uploaded to the server are copied to the current local solution.</li> </ul> <p>It is assumed that the local version of the service components you upload to the server is correct for the capture system, currently selected for the upload process. If you have any unsaved changes in the current capture system and click the <b>Upload</b> button:</p> <ul style="list-style-type: none"> <li>• Unsaved changes are saved automatically.</li> <li>• Service components are uploaded to the server.</li> <li>• Status of the items displayed in the <b>Service Components</b> list is automatically updated.</li> </ul> <p>After clicking the <b>Upload</b> button, users must manually verify that the files are successfully deployed to the client machine in the following path &lt;ApplicationData&gt;\EMC\InputAccel\Custom\bin where &lt;ApplicationData&gt; is a user-defined variable, depending on the location of the user's local ApplicationData folder. Users are required to verify whether files are deployed on a single client machine only.</p> <p>To upload server components changes to the server within the selected capture system, you can also use the <b>Upload Local Changed</b> button in the main menu. When clicked, <i>all</i></p>

Element	Description
	<p>modified service components (if any) within the current capture system are saved, uploaded to the capture server, and the items status is automatically updated.</p> <p> <b>Note:</b> Be aware of the following for uploading components:</p> <ul style="list-style-type: none"> <li>• If you click the <b>Upload</b> button at the bottom of the service components list on the <b>Deployment</b> tab, the changes are saved to the selected items only.</li> <li>• If you click the <b>Upload Local Changed</b> button in the main menu, all modified items within the whole capture system are saved.</li> </ul>
<p><b>Download</b></p>	<p>Use the button to download the selected items from capture server to the local machine. If the local copy of the downloaded component is read-only, such item cannot be downloaded.</p> <p>A file which is not in the configuration, can be downloaded without adding to the configuration. In this case the file will be marked with a triangle icon. If necessary, you can add such files to the configuration later using the <b>System Configuration</b> tab.</p> <p> <b>Note:</b> Downloading of CaptureFlows is not supported. In case you try to download an <i>XPP</i> file, the warning message displays.</p>
<p><b>Delete From Server</b></p>	<p>Use the button to delete the selected items from capture server. When clicked, the deletion confirmation dialog is shown.</p> <p> <b>Note:</b> If you delete from the server the items in the <b>Not in Local Source</b> status, you cannot restore them.</p>

 **Note:** Recognition projects are not stored on the capture server, thus, they are not displayed on the **Deployment** tab.

## 13.3 Uploading or deploying service components including profiles and CaptureFlows

### To upload service components to the server:

1. Navigate to **System**, select the **Deployment** tab.
2. Click the **Refresh list** button to update the list with new information on service components from the server. Also, if the list contains any modified but not yet saved changes, they are automatically saved on the local machine.
3. Select item/items you want to upload by selecting the corresponding check boxes at the left. For easy search, you can use the **Show** filter to show only the service components with a particular status.
4. Click the **Upload** button.

### To deploy CaptureFlows, profile scripts, and data files to the server using the command line:

1. Create a `\bin` folder under the `<CaptureSystem>` directory if it does not exist.
2. Copy your custom DLLs and data files to `<CaptureSystem>\bin`.
3. Copy any additional data files to the same folder.
4. Use the command line utility, `<path to client install>\Client\binnt\DeploymentUtility.exe` to deploy the profile scripts, data files and CaptureFlows. To get help on how to run the utility, execute the following command:

```
deploymentutility.exe -help
```



### Notes

For more information, see the following:

- Make sure that you've already compiled any *XPPs*; this utility does not compile them.
- The maximum length of the command line is limited by Windows.
- Unicode characters are supported for solution names, profile names, and for the arguments file.
- When deploying a process instance, you can specify a name for it that is different from its associated CaptureFlow name using the following parameter:

```
<configuration>_filter:<configname>[,<process-instance>];
```

where:

– `<configuration>`

(Required) A configuration specified in the `-configurations` parameter. You must specify the configuration or `all` in the `-configurations` parameter.

– `<configname>`

(Required) The name of a specific configuration; for example, the name of an XPP.

– `<process-instance>`

(Optional) The name of the process instance that corresponds to the XPP specified in `<configname>`. If you do not specify `<process-instance>`, then the XPP name is used.

– You can use a semicolon to separate multiple `<configname>`s.

5. Restart the client modules.

## 13.4 Filtering service components list

You can filter service components to show or hide them in the list according to their status.

Use the **Show** filter at the top of the service components list to define the status you want to display the components with. Select the corresponding check boxes:

- **Server changed:** To view the items with the server version newer than the local version.
- **Local changed:** To view the items that are newly created and have never been uploaded to the server before, or the items modified locally and the changes are not uploaded to the server after the modification.
- **Unchanged:** To view service components that are equal locally and on server.
- **Not in local source:** To view service components on server only.
- **Not in server source:** To view service components on the local system only.

### Example 13-1: Understanding service components status

The following example can help you understand the role of service component status and how the changes made for a service component affect its status. For instance, you have the following capture systems: **Capture System 1** and **Capture System 2**. You create a component in **Capture System 1**, let's say, Image Conversion profile named **IC Profile**, and its status shows **Not in server source**. When you upload the profile to capture server, the status is changed to **Unchanged** which means the local and server versions are equal. After that, you create the component of the same type with the same name for your **Capture System 2**, namely, Image Conversion profile **IC Profile**.

After connecting to the same capture server, the status for this component shows **Not in server source**, as it is new for currently selected **Capture System 2** and has never been uploaded to the server. After uploading to the server, the

**IC Profile** status is changed to **Unchanged**. If you switch to the **Capture System 1** now, **IC Profile** has the **Server Changed** status. Note that if you remove the **IC Profile** component from the **Capture System 1**, the status will be changed to the **Not in Local Source** value for **Capture System 1**, but remains **Unchanged** for **Capture System 2**.



## 13.5 Deleting service components

**To delete a service component from capture server:**

1. Navigate to **System**, select the **Deployment** tab.
2. Select item/items you want to delete using the corresponding check boxes at the left. For easy search, you can use the **Show** filter to show only the service components that have with a particular status.
3. Click the **Delete From Server** button.
4. When prompted, confirm you want to delete the service component from the server.



### Notes

- When deleting any server components from the server, remember that this operation cannot be undone. Also, if you delete service components which are in the **Not in Local Source** status, you cannot restore them. Whereas, if you delete service components which are in other statuses, the local versions remain and you can upload them to the server once again if necessary.
- When you delete scripting assemblies from the server, they are deleted from the client machine as well.



## Chapter 14

# Testing and debugging a CaptureFlow

## 14.1 Compiling a CaptureFlow

If you design a CaptureFlow that includes many steps, complex routing, scripting for steps, it is recommended to compile it periodically. This operation helps you check if your CaptureFlow contains design errors.

### To compile a CaptureFlow:

1. Open the CaptureFlow chart in CaptureFlow Designer.
2. Press the **Compile** button on the toolbar to compile the selected CaptureFlow. The compilation progress is displayed in the **Compile** panel.

The compilation includes creating the IAP and IPP files and stores them in the current user's Documents folder at the following path: <installation\_dir>\InvoiceCapture\GlobalData\XPP\*<captureflow name>*-dir.

IA value assignments are compiled into an assembly file AG\_*<captureflow name>*.dll that is stored in the \*<captureflow name>*-dir\bin\Release folder.

If the CaptureFlow has scripting, the Emc.InputAccel.CodeBehind.*<captureflow name>*.dll assembly is built and stored in the \*<captureflow name>*-dir\bin\Release folder.

If the chart or scripting contains an error, the compilation fails and the **Problems** panel appears. You can expand this panel to learn about the compilation errors.

### Related Topics:

[“Monitoring warnings and errors” on page 144](#)

## 14.2 Installing a CaptureFlow

You can reuse a CaptureFlow chart to install multiple processes and then set up each process for certain testing needs. Besides, a process can have one or more versions which lets a customer assign batches to a specific version of this process and manage batch execution appropriately if the process needs to be changed later. When you click to install a process, this operation saves the last changes to the **XPP** file, compiles the CaptureFlow, and uploads an instance of the compiled process (IAP and DLL files) on the server. On the server, the changes are saved in a process version folder which contains **XPP**, **IAP**, and **DLL** process files. Thereafter, the version folder is created every time the process changes are uploaded to the server.

Optionally, you can install a process with setup settings copied from another process installed from the same chart.

You can install a process with a unique name or reuse the name of the existing process. In the latter case, the new installed process will rewrite the existing one on the server. This includes replacing the IAP and DLL files.



**Caution**

If the process to be replaced has running batches, rewriting the process files on the server may corrupt these batches. This may happen because the batch (IAB file) reads IA value assignments from a separate DLL file that is compiled and installed with other process files on the server. If the new process has at least one modified (renamed, moved, deleted, or added) step, condition, branch, or custom value, the new DLL with IA values gets out of sync with the running batch.

Reinstalling a process that has a running batch may be reasonable only if you want to replace CaptureFlow scripting (DLL) during the batch execution.


**To install a CaptureFlow:**

1. Open a CaptureFlow chart in CaptureFlow Designer.
2. Expand the **(Choose Installed Process)** drop-down list in the CaptureFlow Designer toolbar and select the **(Install New Process)** option.
3. In the **New Installed Process** dialog box, specify the settings applicable to your use case in the following table.

**Table 14-1:**

Use case	Required settings	Notes
Install a new process. The chart has no other installed processes.	<ol style="list-style-type: none"> <li>1. Specify the unique process name in the <b>Install process with name</b> field.</li> <li>2. Check the <b>Default</b> option to install a process without setup settings.</li> </ol>	<p>The <b>Installed processes</b> list box displays the names of the processes already installed on the selected capture server.</p> <p>The <b>Overwrite process if exists</b> check box is grayed out if the entered process name is unique.</p> <p>The <b>Download from server</b> option is grayed out. This option is only available if you check the <b>Overwrite process if exists</b> option.</p>

Use case	Required settings	Notes
Install a new process. The chart has other installed and set up processes on the server.	<ol style="list-style-type: none"><li>1. Specify the unique process name in the <b>Install process with name:</b> field.</li><li>2. Check any of these available options:<ul style="list-style-type: none"><li>• <b>Default:</b> Check to install a process without setup settings.</li><li>• <b>Copy from process:</b> Check to install a process with setup settings copied from another process installed from the same XPP. Expand the list and pick up the process whose setup settings must be copied.</li></ul></li></ol>	

Use case	Required settings	Notes
<p>Install a new process with the existing process name (reuse the name).</p>	<ol style="list-style-type: none"> <li>1. Enter the existing process name or double-click the process in the <b>Installed processes</b> list.</li> <li>2. Check the <b>Overwrite process if exists</b> check box to confirm the reuse of the process name. The new installed process will rewrite the previous one on the server.</li> <li>3. Specify the setup option for the new process: <ul style="list-style-type: none"> <li>• <b>Default:</b> Check to install a new process without setup settings.</li> <li>• <b>Copy from process:</b> Check to copy setup settings from a different process that originates from the same XPP. Expand the list and pick up the process (its latest version will be handled) whose setup settings must be copied.</li> <li>• <b>Download from server:</b> Check to preserve the setup settings of the replaced process (its latest version will be handled) and apply them to the new installed process.</li> </ul> <p> <b>Note:</b> If the overwritten process and the new one have different steps, the setup settings will apply where the step's name</p> </li> </ol>	<p>The server-side files of the initial process (IAP and DLL) are overwritten with the files of the new process. In other words, the process name remains but the process is replaced with the new one.</p>

Use case	Required settings	Notes
	and module match; the <b>Problem</b> panel will display a warning for each step that requires manual setup.	
Install a process that has been copied while: <ul style="list-style-type: none"> <li>• converting a non-versioned CaptureFlow</li> <li>• saving a CaptureFlow with a different name (cloning a CaptureFlow)</li> </ul>	<ol style="list-style-type: none"> <li>1. Expand the <b>(Choose Installed Process)</b> drop-down list and select the process instance you want to install to the server.</li> <li>2. Click the <b>Edit Process</b> button on the CaptureFlow Designer toolbar. The <b>Edit Process</b> dialog box opens.</li> </ol>	In the CaptureFlow Designer, you need to do this for each of the copied process instances that you want to install.

4. Click **Save**.

### Related topics:

[“Saving a CaptureFlow” on page 121](#)

[“Saving a CaptureFlow with a different name” on page 121](#)

[“Converting a CaptureFlow to the versioned format” on page 93](#)

[“Compiling a CaptureFlow” on page 443](#)

[“Editing an installed CaptureFlow” on page 450](#)

[“Deleting a CaptureFlow instance” on page 452](#)

[“Monitoring warnings and errors” on page 144](#)

## 14.3 Setting up CaptureFlow steps


To have an installed process ready for testing, you need all of its steps be set up. You have several options as follows:

- When **installing a process**, you have an option to copy the module settings from a different process installed from the same XPP file.
- You can install a process with the existing process name and reuse the module settings of that existing process.
- You can install a process without module settings and then set up each process step in CaptureFlow Designer.

Later, you can use CaptureFlow Designer to update module settings for any process step and continue testing it.

When setting up process steps in CaptureFlow Designer, you call the module's setup window for each step and specify the module configuration settings. When a step is triggered in production, the associated module is executed on the client machine with these module settings. Other steps that trigger the same module need to be set up independently.


CaptureFlow Designer saves the module settings in the capture server database and updates the XPP file accordingly.

 **Note:** The module settings can also be modified using Intelligent Capture Administrator or CaptureFlow Designer running on a different desktop. When this happens, the updated module settings are saved on the server, while the local XPP file on your desktop is not updated. CaptureFlow Designer can trace such inconsistencies and help you synchronize your XPP file with the server-side module settings. Besides, you can forcibly synchronize your local XPP file with the server-side module settings by clicking the **Download Settings** button from the CaptureFlow Designer toolbar.

Using both Intelligent Capture Administrator and CaptureFlow Designer for step setup is not recommended.


### To set up a CaptureFlow step:

1. Open the CaptureFlow chart and select the installed process in the (**Choose Installed Process**) list.  
Make sure the **Connected** status is displayed. Reconnect to the server if necessary.
2. Double-click a process step or right-click it and select **Module Settings** from the context menu to invoke the module's setup window.

 **Note:** If the module is not installed on your desktop, the module's setup window cannot be displayed. You can set up such steps later **on a different workstation** and continue with other steps on your desktop. For other

module error causes, check the module's log file. For more information about the error messages and logged codes for each module, see *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*.

3. (Optional) If CaptureFlow Designer detects inconsistencies between the local (XPP) module settings and the server side settings, you can view one of the following message boxes:
  - **Installed Steps are Different:** This message indicates that you have installed a process on the server and then modified and saved the CaptureFlow chart in CaptureFlow Designer (without reinstalling the updated process). The changes are made in the local XPP file. To be able to modify the module settings, you need to reinstall the updated CaptureFlow on the server. Choose the appropriate option:
    - **Update:** Click to **reinstall** the updated process on the server.
    - **Cancel:** Click to cancel the step setup operation.
  - **Download Settings:** This message indicates that the module settings have been updated externally, from Intelligent Capture Administrator or from CaptureFlow Designer running on a different machine. Choose the appropriate option:
    - **Use server settings and overwrite local settings:** Click to download module settings from the server and update the local XPP file accordingly.
    - **Use local settings and overwrite server settings:** Click to upload the local module settings to the server.
4. Specify the required setup parameters in the setup window. Find the details for each module in the appropriate module guide (as described in *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*).
5. To remember the module settings, use one of the following options:
  - **Cancel:** Press this button to quit the step setup and close the setup window. The setup settings are not saved on the server. The XPP file remains unchanged.
  - **Apply:** Press this button to save the current setup settings. This button may be missing in the module's setup window.

 **Note:** If you click it and then choose to cancel, incomplete settings are saved on the server, but the XPP file is not updated. When you try to set up this step again, the previously saved settings are copied to the XPP file and appear in the setup window. This does not corrupt the XPP file. Complete the step's setup as planned.

  - **Save:** Press this button to save the step setup configuration and close the setup window.

The step setup settings are saved on the server. The locally stored XPP file is updated to incorporate all recent changes, which includes the step setup configuration. The updates in the XPP file cannot be reverted.

### Related Topics

[“Installing a CaptureFlow” on page 443](#)

[“Editing an installed CaptureFlow” on page 450](#)

[“Setting up a step on a different machine” on page 450](#)

## 14.3.1 Setting up a step on a different machine

If you cannot set up a CaptureFlow step because the required module is missing on your machine, you can set up this step on a different designer's machine that has this module installed.

### To set up a step on a different workstation:

1. Copy your XPP file to a different machine that has the required module and Designer installed. Copy your XPP to the following folder:  
`C:/Users/<user_name>/My Documents/ <product_name_and_version>/<capture_system_name>/GlobalData/XPP`
2. Run Designer and open the CaptureFlow chart.
3. Set up the required step. Save the settings and close the module's setup window.  
The step setup configuration is saved on the server.
4. Open the CaptureFlow chart on your desktop.
5. On the CaptureFlow Designer toolbar, press the **Download settings** button. This command downloads the step setup configuration from the server and saves it to the local XPP file.

## 14.4 Editing an installed CaptureFlow

You can use CaptureFlow Designer to edit any process available on the server, including processes that have running batches. This may be necessary in the following cases:

- You need to apply and test the latest changes in the CaptureFlow chart and/or in scripting
- You need to test the process with different step setup settings



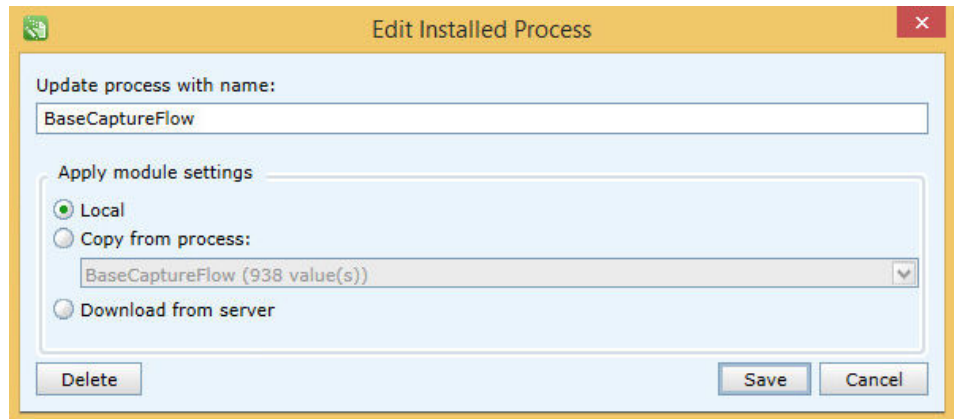
**Note:** You cannot edit the name of the process. To do it, **install** the process with the required name and use the **Copy from process** option to copy the existing setup settings to the new process.

When you confirm the editing, the CaptureFlow is compiled and the new process files (IAP and DLL files) are uploaded on the server, replacing the existing files.

Editing a process that has a running batch can be reasonable only if you want to replace CaptureFlow scripting (DLL) during the batch execution. If the new process has at least one modified (renamed, moved, deleted, or added) step, condition, branch, or custom value, delete the batch in Intelligent Capture Administrator and then edit the process.

#### To edit an installed CaptureFlow:

1. Open the CaptureFlow chart from which the capture process was installed.
2. Expand the **(Install New Process)** list and select the name of the capture process that you want to edit.
3. Click the **Edit Process** button on the CaptureFlow Designer toolbar.



**Figure 14-1: Editing an installed process**

The **Edit Installed Process** dialog box displays the name of the process you are going to edit.

4. Specify how you want to edit setup settings of the installed process:
  - **Local:** Choose this option to copy the local setup settings (kept in the local XPP on your machine) to the server.  
This can be necessary if the process was installed from a different designer's machine, or setup settings of the installed process were modified from Intelligent Capture Administrator or from CaptureFlow Designer installed on a different machine. As a result, your local XPP keeps different setup settings that the installed process on the server side. If you apply this option, you can continue debugging the installed process with your local settings. They will be copied to the server to replace the existing ones.
  - **Copy from process:** Choose this option to take setup settings from a different installed process. Expand the drop-down list and select the process

whose settings you need to copy. Processes without setup settings are displayed as **<process\_name> (empty)**.

Setup settings are copied between processes for each pair of steps where the step names match. The order of steps in each process can be different. Steps without a match do not synchronize their setup settings.



**Note:** Modules that stand behind the matching step names are not checked. Copying setup settings between different modules fails without a warning message.

- **Download from server:** Choose this option to copy the server-side setup settings to the local XPP.

This can be necessary if setup settings of the installed process were modified from Intelligent Capture Administrator or from CaptureFlow Designer installed on a different machine. As a result, your local XPP keeps different setup settings that the installed process on the server side. If you apply this option, your local XPP file will be updated with the server-side settings.

5. Click **Save**.
6. (Optional) If the process has running batches, the **Overwritten process has batch** dialog box displays two options:
  - **Overwrite process anyway:** Rewrite the process files on the server.
  - **Interrupt overwriting the process:** Cancel the rewriting of the new process.

### Related topics:

[“Installing a CaptureFlow” on page 443](#)

[“Setting up CaptureFlow steps” on page 448](#)

[“Deleting a CaptureFlow instance” on page 452](#)

## 14.5 Deleting a CaptureFlow instance

You can use CaptureFlow Designer to delete corrupted or unnecessary processes from the server.



**Note:** You can also delete unnecessary uninstalled process instances that have been copied while cloning a CaptureFlow or converting a non-versioned CaptureFlow.

### To delete an installed CaptureFlow from the server:

1. Open the CaptureFlow chart from which the capture process was installed.
2. Expand the **(Install New Process)** list and select the name of the installed capture process that you want to delete from the server.

3. Click the **Edit Process** button on the toolbar.
4. Click **Delete** in the **Edit Installed Process** dialog box. Select **Delete installed process on server** or **Keep installed process on server** in the prompt dialog box.



**Note:** For an uninstalled process, click **Delete** in the **Edit Process** dialog box.

Before removing a process from the server, make sure all batches created for this process have finished processing. CaptureFlow Designer cannot uninstall a process if any running batches are found. A message box appears to inform you that the process cannot be deleted because of <N> batches running on the server. Also, if a process has at least one associated non-versioned batch, it cannot be deleted and the appropriate notification appears.

If no running or non-versioned batches are detected, the selected capture process is uninstalled from the server. The process files (IAP and DLL) are physically removed from the server, and the step setup information is deleted from the server database. However, all the process version folders remain unchanged in the \IAS\process\Versions\ directory. The locally stored XPP file is updated accordingly.

#### **Related topics:**

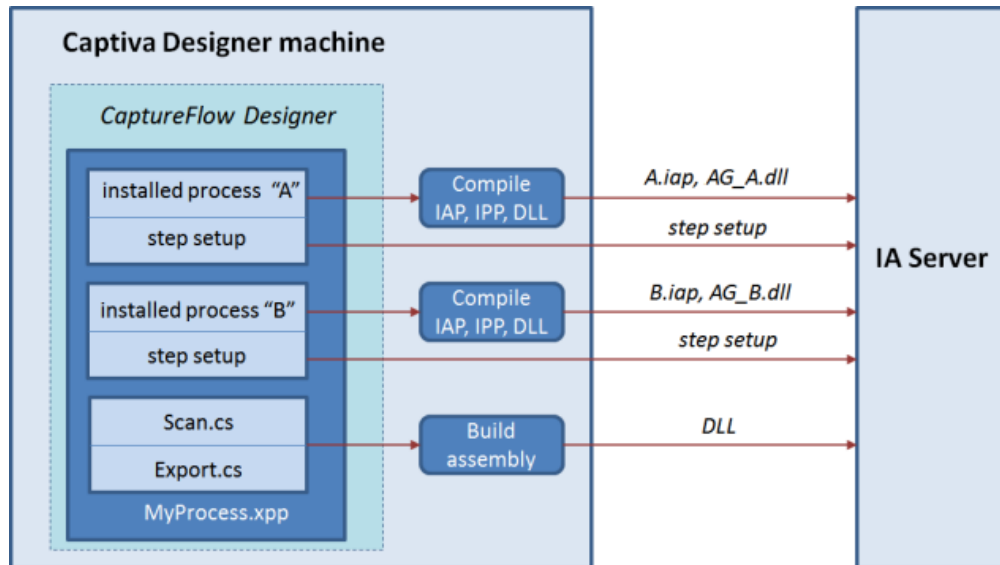
[“Saving a CaptureFlow with a different name” on page 121](#)

[“Converting a CaptureFlow to the versioned format” on page 93](#)

## **14.6 Uploading a CaptureFlow**

If an XPP file has server side installations, CaptureFlow Designer can upload them all and apply their step setup settings on the selected server with a single click.

The following example describes the upload of a “MyProcess” CaptureFlow. The MyProcess.xpp file stores the information about two installed processes (“A” and “B”) with step setup settings and with scripting added for two steps – “Scan” and “Export”:



**Figure 14-2: Uploading a CaptureFlow**

When upload starts, the installed processes remembered in the XPP are reinstalled on the selected server. For each installed process filed in the XPP:

- The IAP and IPP process files are generated. The IAP file is uploaded on the server.
- IA value assignments are compiled and the DLL file is uploaded on the server.
- A copy of the XPP file is uploaded on the server.
- The step setup settings for the reinstalled process are uploaded on the server.

Next, the script code is compiled and the assembly is uploaded on the server.

**To upload a CaptureFlow:**

1. Open the **CaptureFlow List** tab and open the required the CaptureFlow.
2. Click the **Upload Local Changed** button on the CaptureFlow Designer toolbar.

The **Uploading service component** dialog box displays the progress for the compilation process and for the creation of the IAP , IPP, and DLL files.

If the XPP file includes scripting, the **Output** dialog box opens. The assembly files are built after the IAP and IPP files.

If the capture processes are uploaded with their setup settings, they are ready for use in production mode.



**Note:** Some Intelligent Capture modules store part of their step setup settings on the server as global data. Such settings are never saved to an XPP file. This refers to the following modules: ScanPlus, Web Services Input, Standard Import (File System and Email), and eIndex. If you upload an XPP file with any of the above steps to a different server, you need to repeat the setup procedure for those steps on the new server.

If the uploaded capture process uses other capture system components in production, such as module profiles, styles, recognition projects, and others, they all must be located on the same server. To learn how to upload capture system components, see section [“Deploying a component” on page 433](#). To learn how to upload recognition projects, see *OpenText Intelligent Capture - Recognition Designer Guide (ECPCORE-CRC)*.

## 14.7 Downloading components from the capture server

Module profiles, document types, and configuration files can be opened and analyzed on the Designer host machine where they have been developed. After deployment they are stored on the server as database records rather than files. They become available for use on all Designer instances that can connect to this server, but other designers cannot open their content neither for editing nor for viewing.

If you need to see the content of the components installed on the server, use the [Deployment Utility](#) with the `-downloadfromserver` argument to download the required capture system components as files from the server to your local file system.

### Related Topics:

[“Deployment Utility” on page 520](#)



## Chapter 15

# Programming Reference—Expressions

This section contains reference information for creating expressions in Designer tools, such as CaptureFlow Designer, profile editors, Expressions Editor, and Format String Editor.

An expression is a string that evaluates to a single value and can be made up of four types of tokens: literals, data path addressing, standard operators (+, -, \*, /, =, <>, and others), and limited functions.

[“Expressions: types” on page 457](#) provides information on expressions types and where to use them.

**Table 15-1: Expressions: types**

Type of expression	When used	Evaluates to:	Runtime handling/ Runtime data	Where supported
Conditional expression (filter)	Used to process an “item” depending on a condition which is evaluated at runtime.	Boolean value	The expression uses data paths to access runtime data.  Curly braces are not used.	Designer (no exceptions)

Type of expression	When used	Evaluates to:	Runtime handling/ Runtime data	Where supported
Format expression	Used to format strings at runtime by substituting runtime data.	String	<p>The expression uses data paths to access runtime data.</p> <p>All substitutions in the expression are indicated by an element within curly braces. That element has two parts separated by pipe:</p> <ul style="list-style-type: none"> <li>• The data type followed by the string format specifier.</li> <li>• The value expression.</li> </ul> <p>The expression is resolved at runtime and is coerced to the given data type. Then it is formatted as a string. If the value cannot be coerced or if the value expression is invalid, the result of the substitution is a question mark (?).</p>	Designer (except CaptureFlow Designer)
Value expression	Used to read or write data in batch node IA values and UimData fields. This includes <b>data access</b> and data processing using supported <b>functions</b> and <b>operators</b> .	Value (string, int, double, datetime, or boolean)	<p>The expression uses data paths to access runtime data.</p> <p>Curly braces are not used.</p>	Designer (no exceptions)

## 15.1 Data paths

Designer expressions allow you to access data stored in IA values of the batch nodes and in UimData fields. To access an IA value or a UimData field, you need to specify the data path. General rules for data paths are as follows:

- The data type (such as Integer, String, or Double) that a path refers to is only evaluated at runtime on the server. Designer does not check whether the IA value your path points to is the appropriate data type for the assignment or expression.
- Object member access in data paths uses a period (.) as a delimiter.
- Data paths must not contain any embedded spaces or other white space.

The following IA value paths are supported in Designer expressions and format strings:

- *\_Batch.<ValueName>*
- *\_Node:<level0-7>.<NodeProperty>*
- *<StepName>:<level0-7>.<ValueName>.<MemberName>*
- *<StepName>:<level0-7>.\$Runtime.<Value>*
- *<StepName>:<level0-7>.UimData.<FieldName>*



**Note:** UimData paths are not supported in CaptureFlow Designer.

## 15.2 Operators

The Designer expression syntax supports the operators outlined in “Operators” on page 459.

**Table 15-2: Operators**

Operator	Description
AND	Narrows search and retrieve records containing all of the words it separates.
OR	Broadens search and retrieve records containing any of the words it separates.
NOT	Narrows search and retrieve records that do not contain the term following it.
MOD	Divides two numbers and returns only the remainder.
>	Greater than.

Operator	Description
>=	Greater than or equal to.
<	Less than.
<=	Less than or equal to.
=	Equal to.
<>	Not equal to.
+	Addition operator.
-	Subtraction operator.
/	Division operator.
*	Multiplication operator.
&	Converts left value to string, converts right value to string, and then concatenates. Types are converted to string using canonical format. See examples in <a href="#">“Data types” on page 460</a> .

## 15.3 Data types



Every data type can be specified with relational operators; however, the data types on either side of the relational operator must be identical.

The supported data types for an IA value are described in [“Data types” on page 460](#).

**Table 15-3: Data types**

Data type	Description
Boolean	A Boolean value can be set to <code>True</code> or <code>False</code> .

Data type	Description
Date	<p>Dates include a date and a time. You must use the following unambiguous format (the # signs are required delimiters that denote a Date data type):</p> <pre>#yyyy-MM-dd HH:mm:ss#</pre> <p>&lt;yyyy&gt; is the four-digit year.          &lt;MM&gt; is the two-digit month (01–12).          &lt;dd&gt; is the two-digit day (01–31).          &lt;HH&gt; is Two-digit, 24-hour-clock hour (00 - 23).          &lt;mm&gt; is the two-digit minute (01 - 59).          &lt;ss&gt; is the two-digit second (01 - 59).</p> <p>If you specify the date part only, then the time part is implicitly set to 00:00:00. If you specify the time part only, then the date part is implicitly set to the current date.</p>
Number	<p>See the following:</p> <ul style="list-style-type: none"> <li>• Negative sign must precede negative numbers.</li> <li>• Numbers must not have group separators.</li> <li>• Decimal separator is period character and is shown only for Double.</li> <li>• Unary operations are applicable for numbers only. Positive value (+) is implied if negative symbol (–) is not present.</li> <li>• Additive, multiplicative, and unary operations may only be used with numbers.</li> </ul> <p>Syntax: [ - ]ddd[ .ddd]</p>

Data type	Description
String	<p>See the following:</p> <ul style="list-style-type: none"> <li>• When a string is processed: <ul style="list-style-type: none"> <li>– Any double quote character inserted into the string is automatically escaped by another double quote.</li> </ul> </li> </ul> <p> <b>Note:</b> The enclosing double quotes and double quotes used as escape characters are not displayed in Designer; they are only shown in the resulting file (for example, the exported CSV file) in a text editor. For example, if a double quote is added to the beginning of a string, then three double quotes would be displayed in a text editor: the first one is the required enclosing double quote, the second one is the escape character, and the third one is the literal double quote.</p> <ul style="list-style-type: none"> <li>• When equality (=) is applied to strings, a case-insensitive comparison is performed.</li> <li>• Individual characters can also be encoded using either "&amp;#"; where "#" is the decimal value of the character's ASCII code or "&amp;x#"; where "#" is the hexadecimal value of the character's ASCII code. For example, the quote character, whose ASCII code is 34 in decimal and 22 in hexadecimal, can be encoded as "&amp;34"; or "&amp;x22;".</li> </ul> <p> <b>Note:</b> Character encoding is not supported in the Initial Value field of the Custom Values window.</p>

## 15.4 Functions

Expressions created in Designer tools may include supported functions. When you start typing in the Expressions Editor or in the Format String Editor, the autocompletion feature displays available functions.

Expression functions can be nested. For example:  
`SubString(Trim(CustomValues:7.TestString), 5)`

### 15.4.1 Date functions

The supported Date/Time functions are as follows:

#### 15.4.1.1 DateAdd

**Syntax:**

```
DateTime DateAdd(DateTime <datesource>, int <amount>, int <part>)
```

Adjusts the given <datesource> by the amount of year, month, day, hour, minute, or seconds. A negative amount means the date is decreased. The part indicates the unit for the amount. The <part> is:

- 0: Year
- 1: Month
- 2: Day
- 3: Hour
- 4: Minute
- 5: Second

**Example:**

Add 10 days to the date and store in <DateVal>

```
CustomValues:7.DateVal = DateAdd(CustomValues:7.SomeDate, 10, 2)
```

#### 15.4.1.2 DateDiff

**Syntax:**

```
int DateDiff(DateTime <date1>, DateTime <date2>, int <part>)
```

Finds the difference between two dates in terms of the <part> specified. The difference is determined by subtracting date1 from date2 (that is, date2 - date1). To ensure a positive value, take the absolute value of the return value. The return value is truncated to the maximum value that may be held by an integer. The <part> is:

- 0: Year
- 1: Month
- 2: Day
- 3: Hour
- 4: Minute
- 5: Second

**Example:**

Calculates the number of days difference between two dates and stores them in `<DateVal>`

```
CustomValues:7.DateVal = DateDiff(CustomValues:7.SomeDate1,CustomValues:7.SomeDate2, 2)
```

**15.4.1.3 FormatDate****Syntax:**

```
string FormatDate(DateTime <value>, string <dateFormatSpecifier>)
```

Formats the `DateTime` object using `<dateFormatSpecifier>`, which consists of a series of date/time part formats. For a single date/time part, you can only use one of the allowed formats.

The date format separator (optional) is either a forward slash (/), minus (-), or period (.).

Time formats may be separated by a colon (;) character only.

The format output may use the locale of execution unit. Do not use this function to generate locale-specific output strings.

If the `<dateFormatSpecifier>` string is empty, the original date is converted to a date in accordance with the standard “G” format for a specific culture. For details, see the Microsoft website ([https://msdn.microsoft.com/en-us/library/8tfzyc64\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/8tfzyc64(v=vs.110).aspx)). For instance:

- *de-DE Culture* 01.10.2008 17:04:32
- *en-US Culture* 10/1/2008 5:04:32 PM
- *es-ES Culture* 01/10/2008
- *fr-FR Culture* 01/10/2008 17:04:32

The following table provides a description of the standard date/time conversions.

Date/Time part	Format	Description
Year	yy	Two-digit year

Date/Time part	Format	Description
	YYYY	Four-digit year
Month	M	Single-digit month (1 - 12)
	MM	Two-digit month (01 - 12)
Day	d	Single-digit day (1 - 31)
	dd	Two-digit day (01 - 31)
Hour	H	Single-digit, 24-hour-clock hour (0 - 23).
	HH	Two-digit, 24-hour-clock hour (00 - 23).
	h	Single-digit, 12-hour-clock hour (0 - 12).
	hh	Double-digit, 12-hour-clock hour (00 - 12).
Minute	m	Single-digit minute (0 - 59)
	mm	Two-digit minute (00 - 59)
Second	s	Single-digit second (0 - 59)
	ss	Two-digit second (00 - 59)
AM or PM	tt	AM/PM indicator. The exact text is locale dependent.

➔ **Example 15-1:**

<code>FormatDate(#2010-2-10 03:08:00#, "yy")</code>	'Returns "10"
<code>FormatDate(#2010-2-10 03:08:00#, "HH")</code>	'Returns "15"
<code>FormatDate(#2017-02-08 07:32:01#, "yyyy/dd/MM HH:mm:ss")</code>	'Returns "2017/08/02 07:32:01"
<code>FormatDate(#2017-2-18 07:32:01#, "dd/MM/yyyy HH:mm:ss tt")</code>	'Returns "18/02/2017 07:32:01 AM"
<code>FormatDate(#2010-2-10 15:08:00#, "h")</code>	'Returns "3".



### 15.4.1.4 GetDatePart

**Syntax:**

```
int GetDatePart(DateTime <date>, int <part>)
```

Returns the date part from date. The <part> is:

- 0: Year
- 1: Month (1 = January, 12 = December)
- 2: Day (1 to 31)
- 3: Hour (0 to 23)
- 4: Minute (0 to 59)
- 5: Second (0 to 59)
- 6: Week Day (0 for Sunday, 1 for Monday, ..., 6 for Saturday)

**Examples:**

```
CustomValues:7.DateVal = GetDatePart(CustomValues:7.SomeDate, 0) ' Returns year  
CustomValues:7.DateVal = GetDatePart(CustomValues:7.SomeDate, 2) ' Returns day
```

### 15.4.1.5 Now

**Syntax:**

```
DateTime Now()
```

Returns the current date/time value for the current culture.

**Example:**

```
CustomValues:7.DateVal = Now()
```

### 15.4.1.6 ResetDate

**Syntax:**

```
DateTime ResetDate(DateTime <date>)
```

Resets the <date> part to 2000-01-01. The time part is not changed.

**Example:**

```
CustomValues:7.DateVal = ResetDate(CustomValues:7.SomeDate)
```

### 15.4.1.7 ResetTime

**Syntax:**

```
DateTime ResetTime(DateTime <date>)
```

Resets the time part to 00:00:00. The date part is not changed.

**Example:**

```
CustomValues:7.DateVal = ResetTime(CustomValues:7.SomeDate)
```

## 15.4.2 Get/Set functions

The supported Get/Set functions are as follows:

### 15.4.2.1 GetDataProperty

**Syntax:**

```
string GetDataProperty(string <DataProperties>, string <PropertyName>)
```

Searches the DataProperties for a property substring in the following format:  
property = value\n (new line character).

If a matching substring was found, returns portion of the substring between “=” and “\n”. If no match was found, returns an empty string .

**Example:**

```
StepA:0.MyValue = GetDataProperty(FWComp:0.InputData.DataProperties, "SomeName")
```

## 15.4.3 Numeric functions

The following Numeric functions are supported:

### 15.4.3.1 Abs

**Syntax:**

```
double Abs(double <value>)
```

Returns the absolute of <value> as a positive number.

**Examples:**

```
CustomValues:7.DoubleVal = Abs(CustomValues:7. SomeDoubleVal)
```

```
CustomValues:7.DoubleVal = Abs(9.8) ' Returns 9.8
```

```
CustomValues:7.DoubleVal = Abs(-9.8) ' Returns 9.8
```

### 15.4.3.2 Ceiling

**Syntax:**

```
double Ceiling(double <value>)
```

Returns the upper bound for the double <value>.

**Examples:**

CustomValues:7.DoubleVal = Ceiling(CustomValues:7.SomeDoubleVal)
CustomValues:7.DoubleVal = Ceiling(9.8) 'Returns 10
CustomValues:7.DoubleVal = Ceiling(9.2) 'Returns 10
CustomValues:7.DoubleVal = Ceiling(-9.8) 'Returns -9
CustomValues:7.DoubleVal = Ceiling(-9.2) 'Returns -9

### 15.4.3.3 Floor

**Syntax:**

```
double Floor(double <value>)
```

Returns the lower bound for the double <value>.

**Examples:**

CustomValues:7.DoubleVal = Floor(CustomValues:7.SomeDoubleVal)
CustomValues:7.DoubleVal = Floor(9.8) 'Returns 9
CustomValues:7.DoubleVal = Floor(9.2) 'Returns 9
CustomValues:7.DoubleVal = Floor(-9.8) 'Returns -10
CustomValues:7.DoubleVal = Floor(-9.2) 'Returns -10

### 15.4.3.4 FormatNumber

**Syntax:**

```
string FormatNumber(double <value>, string <numberFormatSpecifier>)
```

Formats the number using the format specifier. The specifier string uses a combination of one of the following placeholder characters. The formatted output may use the locale of the execution unit. Do not use this function to generate locale-specific output string. See the following:



**Note:** The specifier cannot be an empty string.

- '#': Digit placeholder. If the value being formatted has a digit in the position where '#' appears in the format string, then that digit is copied to the result string. Otherwise, nothing is stored in that position in the result string.

- '0': Zero placeholder. If the value being formatted has a digit in the position where '0' appears in the format string, then that digit is copied to the result string; otherwise, '0' appears in the result string. The position of the leftmost '0' before the decimal point and the rightmost '0' after the decimal point determines the range of digits that are always present in the result string.
- '.', ': Decimal point. Only one character must be used. The actual character used as a decimal separator is locale-dependent.
- ',', ': Thousand separator. The actual character used is locale-dependent. If one or more ',' characters are specified between two-digit placeholders ('0' or '#') that format the integral digits of a number, a group separator character is inserted between each number group in the part of the output.
- ';': Section separator. The ';' character is used to separate sections for positive, negative, and zero numbers in the format string. If there are two sections in the custom format string, the leftmost section defines the formatting of positive and zero numbers, while the rightmost section defines the formatting of negative numbers. If there are three sections, the leftmost section defines the formatting of positive numbers, the middle section defines the formatting of zero numbers, and the rightmost section defines the formatting of negative numbers.

**Examples:**

```
FormatNumber(8, "#;(#)") 'Returns "8"
```

```
FormatNumber(-8, "#;(#)") 'Returns "(8)"
```

```
FormatNumber(8, "%") 'Returns "800%"
```

**15.4.3.5 Random****Syntax:**

```
int Random(int <range>)
```

Returns a random integer between 0 and (<range> -1). If <range> is less than zero, returns 0.

**Examples:**

```
CustomValues:7.IntVal = Random(10) 'Returns random number from 0 thru 9
```

### 15.4.3.6 ToNumber

**Syntax:**

```
double ToNumber(object value)
```

Converts the object to a number as follows. Does not throw any errors. See the following:

- Null value: 0
- Number: No conversion needed.
- Boolean: True:1, False:0
- DateTime: A number representing number of days and fractional day from 1899, Dec 30.
- String: If empty, converts to 0. If numeric string, it is parsed using period as a decimal separator. If it starts with a numeric string, then only the string up to the non-numeric, non-decimal separator is parsed.

**Example:**

```
CustomValues:7.DoubleVal = ToNumber("1234 Trade St") 'Returns 1234
```

## 15.4.4 String functions

A number of String functions are supported.

### 15.4.4.1 Capitalize

**Syntax:**

```
string Capitalize(string <value>)
```

Returns a capitalized string based on the following rules:

- If the first character is a letter, it is always capitalized.
- If the next character after a white space is a letter then it is capitalized.

### 15.4.4.2 ApproximateMatch

**Syntax:**

```
int ApproximateMatch(string <source1>, string <source2>, bool  
<caseSensitive>)
```

Returns an approximate match value in the range of 0 - 100 where 100 indicates complete match and 0 means no match at all. The match is determined by the longest string length minus minimum number character replacements needed to get a total match:

```
((maxlen - replaceCharCount) * 100) / maxlen
```

**Example:**

```

ApproximateMatch("abcd", "abcx", 1 )      'Returns 75
ApproximateMatch("abcd", "abcxy", 1 )     'Returns 65
ApproximateMatch("abcd", "abcd", 1 )      'Returns 100
ApproximateMatch("abcd", "xabcd", 1 )     'Returns 80
ApproximateMatch("abc", "xyz", 1 )        'Returns 0
ApproximateMatch("abcdefg", "abxyz12", 1) 'Returns 28

```

This operation is very compute-intensive. To limit the computing time, only first 80 characters are used for detecting minimum number of corrections needed. After 80 characters, a simple positional comparison is used. For strings longer than 80 characters, the return value may not represent minimum changes needed but approximately identifies the amount of changes needed.

**15.4.4.3 CompareStrings****Syntax:**

```
int CompareStrings(string <source1>, string <source2>)
```

Returns result from comparing <source1> and <source2>. The strings are compared using case sensitive comparison. The result is one of the following:

- 0: Strings are equal.
- 1: <source1> is greater than <source2>.
- -1: <source1> is less than <source2>.

If a parameter's data type is not a string then it is converted to a string and compared.

**Examples:**

```

CompareStrings(CustomValues:7.StringVal1, CustomValues:7.StringVal2) = 0
CompareString(CustomValues:7.StringVal1, "Sample string") <> 0

```

**15.4.4.4 CreateGUID****Syntax:**

```
string CreateGUID(int <format>)
```

Returns a globally unique identifier (GUID). The following format values are supported:

- 0: Default format. Returns the string representation of the generated GUID for Guid.NewGuid().ToString().
- 1: Same as above in uppercase.
- 2: Default format after stripping "-" characters.
- 3: Same as above converted to uppercase.

- Other values: Same as default format.



**Note:** This function is not supported in CaptureFlow Designer expressions.

#### 15.4.4.5 IndexOf

**Syntax:**

```
int IndexOf(string <source>, string <search>, int <startIndex>, bool <caseSensitive>)
```

Returns a zero-based index of the first occurrence of the <search> string inside the <source> string. Returns -1 if the <search> string does not exist in the <source> string. The search starts from the given <startIndex> and is either case-sensitive or insensitive based on the <caseSensitive> parameter.

#### 15.4.4.6 Length

**Syntax:**

```
int Length(String <source>)
```

Returns the number of characters in the <source> string.

**Examples:**

```
CustomValues:7.IntegerVal = Length(CustomValues:7.SomeStringVal)
```

```
CustomValues:7.IntegerVal = Length("This is a string with 35 characters")
```

#### 15.4.4.7 MatchAny

**Syntax:**

```
int MatchAny(string <sourceList>, string <valueList>)
```

Returns the number of non-empty tokens in the comma-separated <sourceList> that match with non-empty tokens in the comma-separated <valueList>. The comparison is case-insensitive. If any list has a token with a comma or an ampersand, these must be escaped using XML escape syntax. If tokens are duplicated, the count includes all duplicate matches. For example, MatchAny("a,b,c,d", "b,d,e,f") returns 2.

### 15.4.4.8 RegexMatch

#### Syntax:

```
bool RegexMatch(string <value>, string <pattern>)
```

Returns true if the given <value> matches with the regex <pattern>. The regex pattern is based on the .NET Regex class.

### 15.4.4.9 RegexReplace

#### Syntax:

```
string RegexReplace(string <value>, string <pattern>, string  
<replaceWith>, bool <all>, bool <caseSensitive>)
```

Uses the <pattern> string (regular expression) to make character replacements in the <value> string and returns one or more replacement occurrences. Is based on the standard Regex replace functionality. For more detail, see Microsoft Regular Expression tutorial ([http://msdn.microsoft.com/en-us/library/az24scfc\(v=vs.120\).aspx#character\\_classes](http://msdn.microsoft.com/en-us/library/az24scfc(v=vs.120).aspx#character_classes)).

See the following:

- If <all> is true then all occurrences in the <value> string are replaced with the <replaceWith> string; otherwise, only the first match is replaced.
- If <caseSensitive> is true then the search is case-sensitive.

### 15.4.4.10 Basic string manipulations with RegexReplace

“Examples: Using RegexReplace for common string manipulations” on page 473 shows how to use the RegexReplace function for most typical string manipulations.

**Table 15-4: Examples: Using RegexReplace for common string manipulations**

Use Case	<pattern>	<replaceWith>	<all>	<caseSensitive>	Example
Remove spaces and CR/LF	"\S"	"" (an empty string)	"true"	either	"12 34 56" To: "123456"
Character replacement	"0"	"O"	"true"	either	"CODE" To: "CODE"
Delete a word (and its trailing space)	"to\S"	"" (an empty string)	"true"	either	"A to B" To: "A B"

Use Case	<pattern>	<replaceWith>	<all>	<caseSensitive>	Example
Keep the first three characters	"(.{3}).*"	"\$1"	"false"	either	"12345" To: "123"
Keep the last three characters	".*(.{3})"	"\$1"	"false"	either	"12345" To: "345"
Normalize, e.g., by trimming leading/trailing non-numeric characters	"\D*(.*\d)\D*"	"\$1,"	"false"	either	"a b123c4d" To: "123c4"
Punctuation removal	"[,.]"	"" (an empty string)	"true"	either	"John A. Smith, M.D." To: "John A Smith MD"
Find/keep a word (string is unchanged otherwise)	".*(yes).*"	"\$1"	"false"	either	"Married? yes" To: "yes"
Keep everything between two words (string is unchanged otherwise)	".*call ([0-9-]+) after.*"	"\$1"	"false"	either	"call 888-643-3084 after 5:00" To: "888-643-3084"
Keep the first phone number	".*phone(?:[0-9-]+).*phone([0-9-]+).*"	"\$1"	"false"	either	"A phone 12-3 B phone 45-6 C" To: "12-3"
Reject values with "car" (exclusion keyword)	".*car.*"	"" (an empty string)	"false"	either	"Red car" To: "" (an empty string)

### 15.4.4.11 String manipulations with RegexReplace

“Examples: Using RegexReplace in use cases not covered by the other functions” on page 475 shows how to use the `RegexReplace` function in use cases when the other supported string functions cannot be used.

**Table 15-5: Examples: Using RegexReplace in use cases not covered by the other functions**

Use case	<value>	<pattern>	<replaceWith>	<all>	<caseSensitive>	Result
Delete the text matching a pattern (redact an SSN)	“The statement for SSN 915-23-1634 is attached.”	“(.*\d{3}-\d{2}-\d{4})(.*)”	“\$1XXX-XX-XXXX \$2”	“false”	either	“The statement for SSN XXX-XX-XXXX is attached.”
Keep everything between two keywords	“Your agent is Michael Smith today.”	“.*is (.*) today.*”	“\$1”	“false”	either	“Michael Smith”
Keep one of two keywords (“bill” or “receipt”) in a longer string	“Attached is your bill.”	“.*(bill receipt).*”	“\$1”	“false”	either	“bill”
Extract the third number in a string	“123 abc 456 def 789 ghi”	“.*?(?:\d+\D+){2}(\d+).*”	“\$1”	“false”	either	“789”
Keep one of many values with a confirming keyword (keep the first number that looks like a weight in kg)	“123 abc 456 kg 789 ghi”	“.*?(\\d+)kg.*”	“\$1”	“false”	either	“456”
Extract all numbers	“123 abc 456 def 789 ghi”	“\\D*(\\d+)\\D*”	“\$1,”	“true”	either	“123,456,789,”

Use case	<value>	<pattern>	<replaceWith>	<all>	<caseSensitive>	Result
Case-sensitive search (extract a name as two capitalized words)	"Sent by John Smith"	".*([A-Z] \w+ [A-Z] \w+).*"	"\$1"	"false"	"false"	"John Smith"
Normalize a phone number, stripping non-digits	"(800)555-1212"	"\D"	"" (empty string)	"true"	either	"8005551212"
Normalize a phone number, unifying the format	"800.555.1212"	".*(\d{3}).*(\d{3}).*(\d{4}).*"	"(\$1) \$2-\$3"	"false"	either	"(800) 555-1212"

#### 15.4.4.12 RemoveCharacters

##### Syntax:

```
string RemoveCharacters(string <value>, string <charactersToRemove>)
```

Returns a new string after removing all occurrences of characters from <value> that match any character given in the <charactersToRemove> string.

If <charactersToRemove> is passed as literal string then any non-printable character must be passed as an XML encoded character.

##### Example:

Remove the line feed:

```
CustomValues:7.StringVal=RemoveCharacters(CustomValues:7.SomeStringVal, "&#10;")
```

#### 15.4.4.13 RemoveWhiteSpace

##### Syntax:

```
string RemoveWhiteSpace(string <value>)
```

Returns a new string after removing all white space characters (as specified by .NET) from <value>.

#### 15.4.4.14 Replace

##### Syntax:

```
string Replace(string <source>, string key, string value)
```

Returns the <source> string value after replacing every occurrence of <key> with <value>. No replacement occurs if the <key> is empty or <value> is empty. The key comparison is case sensitive.

##### Examples:

```
CustomValues:7.StringVal=Replace(CustomValues:7.SomeStringVal, "~", " ")
```

```
CustomValues:7.StringVal=Replace("Replace all s with x", "s", "x")
```

#### 15.4.4.15 ReplaceCharacters

##### Syntax:

```
string ReplaceCharacters(string <value>, string  
<replaceCharacterPairs>)
```

Returns the <value> string after replacing any occurrence of the specified character with the replacement character.

The <replaceCharacterPairs> parameter is treated as a string of character pairs where the first character is the existing character and the following character is the replacement character.

##### Example:

Replace "0" with "O", "1" with "l", "B" with "8", and tab with space:

```
CustomValues:7.StringVal=ReplaceCharacters(CustomValues:7.StringVal, "00I1B8&#9; ")
```

#### 15.4.4.16 Select

##### Syntax:

```
string Select(bool <condition>, string <>trueValue>, string  
<>falseValue>)
```

Returns one of two strings based on a condition. If <condition> is "true" then <>trueValue> is returned. If <condition> is "false" then <>falseValue> is returned.

### 15.4.4.17 Substring

**Syntax:**

```
string Substring(string <source>, int <startIndex>, [optional] int <length>)
```

Returns a substring from <source>. The substring starts at a specified character position (<startIndex> is 0-based) and has a specified length. See the following:

- Returns an empty string if <startIndex> is greater or equal to string length or length is less than or equal to 0.
- Returns remaining string from <startIndex> if <startIndex> + <length> is greater than <source> string length.

**Examples:**

```
CustomValues:7.StringVal = Substring(CustomValues:7.SomeStringval, 5)
CustomValues:7.StringVal = Substring("Only get last word", 14) 'Returns "word"
CustomValues:7.StringVal = Substring("Only get last word", 14, 4) 'Returns "word"
```

### 15.4.4.18 ToLower

**Syntax:**

```
string ToLower(string <source>)
```

This function returns string with all lowercase characters using the current culture.

**Examples:**

```
CustomValues:7.StringVal = ToLower(CustomValues:7.SomeStringval)
CustomValues:7.StringVal = ToLower("Convert this string to all lowercase letters.")
```

### 15.4.4.19 ToUpper

**Syntax:**

```
string ToUpper(string <source>)
```

This function returns a string with all uppercase characters using the current culture.

**Examples:**

```
CustomValues:7.StringVal = ToUpper(CustomValues:7.SomeStringval)
CustomValues:7.StringVal = ToUpper("Convert this string to all uppercase letters.")
```

#### 15.4.4.20 Trim

##### Syntax:

```
string Trim(string <source>)
```

This function trims white spaces from the front and back of the <source> string and returns the result.

##### Examples:

```
CustomValues:7.StringVal = Trim(CustomValues:7.SomeStringVal)
```

```
CustomValues:7.StringVal = Trim("  Trim the whitespace from the front and back  ")
```

## 15.5 Data formatting

Formatting is the process of converting an instance of a class, structure, or enumeration value to its string representation, often so that the resulting string can be displayed to users or deserialized to restore the original data type.

In Designer, numeric and date and time data types can be converted to string representation as per format specifications. Data formatting is used in **format expressions**.



**Note:** CaptureFlow Designer does not support data formatting and format expressions.

### 15.5.1 Format string syntax

The format string is structured as follows:

```
[<any text>]{[<Format Specification>|]<Expression>}[<any text>]
```

where

<Format Specification> stands for the following sections:

```
<Data Type Identifier><Format Specifier>
```

where:

- <Data Type Identifier> can be:
  - String ID: 'S'
  - Numeric ID: 'N'
  - DateTime ID: 'T'
- <Format Specifier> is the format specifier. Designer uses Microsoft .NET Framework DateTime and Number formats. Support is limited to the *customformat* strings for those formats; standard format strings are not supported.

For example:

```
{N0.00|Completion:1.UimData.Price}
```

The format specification is optional. If a format string is missing both the *data type identifier* and the *format specifier*, the default settings are used to resolve the string. If only the data type identifier is specified (S, N, T, or I), the default format specification settings will be applied for that data type.

## 15.5.2 Custom number format examples

The culture for the examples outlined in the following table is “en-US”.

**Table 15-6: Custom number format examples**

Number	Format specifier	Formatted string
123	00000	00123
.56	0.0	0.6
1234567890	#, #	1,234,567,890
123	#####	123
1234567890	#	1234567890
123.56	##00.00	123.56

## 15.5.3 Custom date and time format examples

The culture for the examples outlined in the following table is “en-US”.

**Table 15-7: Custom date time format examples**

Date time	Custom format specifier	Formatted string
10/27/2009 10:47:50 AM	dd-MM-yyyy	27-10-2009
10/27/2009 10:47:50 AM	hh:mm:ss tt	10:47:50 AM
10/27/2009 10:47:50 AM	MMM. d, yyyy	Oct. 27, 2009
10/27/2009 10:47:50 AM	dd-MM-yyyy hh:mm:ss tt	27-10-2009 10:47:50 AM

## Chapter 16

# Reference


### 16.1 Naming Conventions

“Capture system components: naming convention” on page 481 describes the naming conventions for the objects.

**Table 16-1: Capture system components: naming convention**

Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
CaptureFlow	Must be unique in the capture system.	The maximum allowed length is 64 bytes.	Must begin with an alphabetic character and include alphanumeric characters. Underscores, spaces, square brackets are not allowed.  Use ASCII characters if you have the English locale set on your machine. Otherwise, use the code page that is installed on the server on which the CaptureFlow will be uploaded and executed.	Because the CaptureFlow name is used as the class name in CaptureFlow scripting, the CaptureFlow name cannot be the exact match of any C# or VB.NET reserved word. Furthermore, whether or not a CaptureFlow contains scripting, OpenText does not recommend that a CaptureFlow name be the exact match of a C# or VB.NET reserved word. For a list of C# and VB.NET reserved words, see the Microsoft documentation.

Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
Capture Server	Must be unique in the Capture Server list.	The maximum allowed length is 64 characters.	Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters and single spaces. Underscore is not allowed.	

Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
Capture System	Must be unique on the <b>Capture System list</b> tab of Designer.	The maximum allowed length is 64 characters.	<p>Characters must be alphanumeric characters in ASCII or the current code page; single spaces could also be included. Underscore is not allowed.</p> <p> <b>Note:</b> Because a process cannot be compiled if the Capture System name contains characters that are not in the code page of the machine on which the process is being compiled, you need to select the same settings that correspond to the code page of the characters in the Capture System name by setting the following in <b>Control Panel &gt; Region</b> (or</p>	

Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
			<b>Region and Language)</b> : • <b>Formats</b> > Format • <b>Administrative</b> > Language for non-Unicode programs > Change system locale	
Connection	Must be unique in the capture system.	The maximum allowed length is 64 characters.	Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters and single spaces. Underscore is not allowed.	
Custom Style	Must be unique in the capture system.	The maximum allowed length is 64 characters.	Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters and single spaces. Underscore is not allowed.	

Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
Custom Value		Not limited	A custom value name must conform to the language for the system code page and contain alphanumeric or underscore (_) characters. The first character must be alphabetic.	Custom value names cannot start with IA or match the reserved IA value names.
Document Type	Must be unique in the Document Types list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces. Underscores and spaces are not allowed.	
Export profile	Must be unique in the Export Profiles list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces. Underscores and spaces are not allowed.	
Image Conversion profile	Must be unique in the Image Conversion Profiles list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces. Underscores and spaces are not allowed.	

Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
Image Processing profile	Must be unique in the Image Processing Profiles list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces. Underscores and spaces are not allowed.	
Import profile	Must be unique in the Import Profiles list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces. Underscores and spaces are not allowed.	
Population Rule / Validation Rule	Must be unique in the Document Type	The maximum allowed length is 64 characters.	Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters. Underscores and spaces are not allowed.	If you are defining a rule with scripting, the rule's name will be included in the .NET method: <code>ExecutePopulationRule&lt;rulename&gt;</code> or <code>ExecuteValidationRule&lt;rulename&gt;</code> .
Query	Must be unique in the Query Name list.	The maximum allowed length is 64 characters.	Must begin with a Unicode alphabetic character and include Unicode alphanumeric characters and single spaces. Underscore is not allowed.	


Capture system component	Naming rules			
	Scope	Length	Allowed characters	Special requirements
Recognition Project	Must be unique in the Recognition Projects list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces.	
Standard OCR profile	Must be unique in the Standard OCR Profiles list.	The maximum allowed length is 64 characters.	Must begin with an alphabetic character and include only alphanumeric characters and single spaces. Underscores and spaces are not allowed.	


### 16.1.1 Reserved IA values


Reserved IA values have special meaning inside the Intelligent Capture Server process code (even though they are not defined in an *MDF*). Do not redefine the variables using these names within an *IPP* or *MDF*. If you want to access or manipulate any of the following values, assign them to another variable.

See “Reserved IA values” on page 487 for a list of IA values that are reserved.

**Table 16-2: Reserved IA values**

IA value	Description
<Any>	Used to access common member variables in an <i>IAVariant</i> data type.   <b>Note:</b> <i>IAVariant</i> is supported in Process Developer, but it is not supported in Designer.
<BatchHold>	Places a batch in a hold state.
<BatchError>	Places a batch in an error state.
<BatchID>	Identification number assigned to a batch by the Intelligent Capture Server.
<Batchname>	Name assigned to a batch when it is created.
<Compile_AppVersion>	Version of Process Developer used to compile a process.

IA value	Description
<Compile_Time>	Time that Process Developer compiled a process. The format of this value is dependent on the date/time/internalization settings of the Intelligent Capture Server.
<Compile_VbaVersion>	Version of VBA that Process Developer used to create an IPP file.
<Description>	Description of the batch, as entered or modified by a ScanPlus or Intelligent Capture Administrator operator.
<End>	Keyword used in the MDF file.
<ErrorStatus>	Places a task in an error state.
<IComDLLVersion>	Internal version number of the IComDLL currently installed on the Intelligent Capture Server.
<IADepartment>	Department name that is associated with a step in an IPP file.
<IATaskPriority>	<p>A task-level IA value that specifies the priority of the current task. This value overrides the task's current batch priority (specified by the &lt;Priority&gt; IA value), unless the &lt;Priority&gt; IA value is set to 0 (zero):</p> <ul style="list-style-type: none"> <li>• Valid values: integers 0 - 99, inclusive</li> <li>• Default value: 0 (zero)</li> </ul> <p> <b>Note:</b> If this value is equal to 0 (zero), then the priority of this task uses the current batch priority.</p>
<IATaskRouting>	Department name that is associated with a task in an IPP file.
<Item>	Used to access dynamic values.
<LockDebug>	Internal value that indicates the lock status of the node being processed.
<Priority>	<p>Priority of a batch (default 50).</p> <p>By default, task priority is defined by the batch priority. You could also set the priority of a specific task by setting the IATaskPriority IA Value. For more information, see <a href="#">“Prioritizing tasks for a step” on page 110</a> and <a href="#">“Reserved IA values” on page 487</a>.</p>
<ProcessCompilerVersion>	Internal number of the compiler used to create the process. Version 3.0 of Process Developer returns 10.

IA value	Description
<ProcessName>	Name of the process from which a batch was derived.
<Ready>	Trigger IA value used to indicate that the specified node is ready for processing.
<RetriesLeft>	Special counter IA value controlling how many times a task that encounters an error should be reprocessed by the module where the error occurred.
<Status>	<p>Processing status of a specified node. Possible values include:</p> <ul style="list-style-type: none"> <li>• 0: Not applicable</li> <li>• 1: Not ready</li> <li>• 2: Ready</li> <li>• 3: Working</li> <li>• 4: Done</li> </ul>
<TaskCount>	Number of pending tasks in this batch for the specified step.
<TaskCountA>	Number of active tasks in this batch for the specified step.
<TaskCountQ>	Number of queued tasks in this batch for the specified step.
<Type>	<p>The name of the object type of an Object, or the name of the currently active object type of an IAVariant.</p> <p> <b>Note:</b> IAVariant is supported in Process Developer, but it is not supported in Designer.</p>
<TreeVersion>	Local Intelligent Capture Server version number for the Intelligent Capture tree, used internally to keep the tree synchronized.
<Trigger_*>	Reserved value where "*" is wild card.
<TriggerLevel>	The trigger level of the step as specified in the IPP and XPP files.

## 16.2 IA Values

The topics in this section describe IA Values that can be used in expressions and scripting.

### 16.2.1 `_Batch` IA Values

The following `<_Batch>` IA Values are write-only and available in the **IAValue** field in the **Assign Values** window. These values require a value of True and enable you to stop batch processing. You can release a batch manually in the Intelligent Capture Administrator or automatically using the Timer module. For more information, see *OpenText Intelligent Capture - Utilities Modules Guide (ECPCORE-CMU)*.

See “`<_Batch>` IA values” on page 490 for the `<_Batch>` IA values.

**Table 16-3: `<_Batch>` IA values**

IA value	Description
<code>&lt;_Batch.BatchError&gt;</code>	Places batches in an error state.
<code>&lt;_Batch.BatchHold&gt;</code>	Places batches in a hold state.

The `<_Batch>` IA Values outlined in the following table are read-only and are not available in the **IAValue** field in the **Assign Values** window.

**Table 16-4: `<_Batch>` IA values**

IA value	Description
<code>&lt;_Batch.BatchID&gt;</code>	Provides the ID of the batch for identification purposes.
<code>&lt;_Batch.ProcessName&gt;</code>	Returns the name of the process that this batch was created from.

The `<_Batch>` IA Values outlined in “`<_Batch>` IA values” on page 490 are read-write and are available in all of the fields in the **Assign Values**, **Step Departments**, and **Condition** windows.

**Table 16-5: `<_Batch>` IA values**

IA value	Description
<code>&lt;_Batch.Description&gt;</code>	Updates the description attached to the batch.
<code>&lt;_Batch.BatchName&gt;</code>	Provides the actual name of the batch.

IA value	Description
<_Batch.Priority>	<p>Sets the priority of the batch in Intelligent Capture Server.</p> <p>By default, task priority is defined by the batch priority. You could also set the priority for a specific task by setting the IATaskPriority IA Value. For more information, see <a href="#">“Prioritizing tasks for a step” on page 110</a> and <a href="#">“Reserved IA values” on page 487</a>.</p>



## Related Topics

[“Programming Reference—Expressions” on page 457](#)

### 16.2.2 <\_Node> IA Values

The <\_Node> properties outlined in [“<\\_Node> IA values” on page 491](#) are read-only and are only available in expressions.

**Table 16-6: <\_Node> IA values**

IA value	Description
<_Node.NodeID>	<p>Retrieves the ID from the current node.</p> <p>Example: &lt;_Node:0.NodeID&gt;</p>
<_Node:<level>.NumChildrenAt[L0...L6]>	<p>Gets the number of direct descendants that belong to a specified level. &lt;[L0...L6]&gt; is replaced with the actual level based on the specified level of &lt;_Node&gt;.</p> <p> <b>Note:</b> This IA Value was renamed from &lt;NumChildren&gt;. For additional information regarding &lt;NumChildren&gt;, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p>
<_Node:<level>.NodeIndexFrom[L1...L7]>	<p>Returns the index of a current node within a specified higher level. &lt;[L1...L7]&gt; is replaced with the actual level based on the specified level of &lt;_Node&gt;.</p> <p> <b>Note:</b> Renamed from &lt;NodeCount&gt;. For additional information regarding &lt;NodeCount&gt;, see <i>OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)</i>.</p>

## Related Topics

[“Programming Reference—Expressions” on page 457](#)

### 16.2.3 Custom values

To reference a custom value in expressions, use the following syntax:

CustomValues:<level>.<value name>

#### Example 16-1:

CustomValues:0.OrderDate



## Related Topics

[“Defining a custom value for a CaptureFlow” on page 122](#)

[“Programming Reference—Expressions” on page 457](#)

### 16.2.4 Dynamic IA Values (\$Runtime)

The <\$Runtime> IA Value outlined in the following table accesses dynamic values.

**Table 16-7: Dynamic IA values**


IA value	Description
<\$Runtime>	<p>Accesses a dynamic IA Value. The &lt;\$Runtime&gt; value is not validated at validation or compile time for existence because dynamic values do not exist at design time. The only validation is a syntax check:</p> <ul style="list-style-type: none"> <li>• Attribute: String, Output</li> <li>• Level: 0–7</li> </ul> <p><b>Example:</b> &lt;ScanPlus:0.\$Runtime.MyRuntimeValue&gt;</p>

## Related Topics

[“Programming Reference—Expressions” on page 457](#)

## 16.2.5 ErrorCode IA Value

You configure error handling for each client module in setup mode. In CaptureFlow Designer, you configure error handling options that occur after the client module attempts to handle the error itself. In production mode, after the client module handles the error CaptureFlow Designer populates the `<ErrorCode>` IA Value when you select the **Continue on to the next step** option in the **Properties** window of the **Step** icon. A complete list of client module error and log codes is listed in *OpenText Intelligent Capture - Module Reference (ECPCORE-CMD)*.

 **Note:** If the client module runs as a service and encounters an error, none of the error handling options configured in setup mode apply.

See the `<ErrorCode>` IA Value outlined in “`<ErrorCode>` IA value” on page 493.

**Table 16-8: <ErrorCode> IA value**

IA Value	Description
<code>&lt;ErrorCode&gt;</code>	<p>Contains the error code that is passed from the client module after an error is received by the Intelligent Capture Server. The error code passed from the client module may not be representative of the actual error and may be a generic non-zero error code. The error depends on the client module and its setup configuration. To diagnose the error, obtain additional information by using the client module IA Values such as <code>&lt;ErrorNumber&gt;</code> and <code>&lt;ErrorText&gt;</code>. The individual client module guides (as listed in <i>OpenText Intelligent Capture - Module Reference Help (ECPCORE-H-CMD)</i>) contain information about configuring error handling in setup mode and specific IA Values that return error information.</p> <p>The <code>&lt;ErrorCode&gt;</code> IA Value occurs at the trigger level only. For example, if the step runs at level 1, then it exists only at level 1. See the following:</p> <ul style="list-style-type: none"> <li>• Attribute: Integer, Output</li> <li>• Level: 0-7</li> </ul>

### Related Topics

“Programming Reference—Expressions” on page 457

## 16.2.6 Step IA Values

Specifies the step names. The syntax is: `<step name>:<level>.<ValueName[.MemberName...MemberName]>`



**Note:** `<MemberName>` is used to access object members.

`<ScanPlus:7.BatchProcessingTime>`

### Related Topics

[“Programming Reference—Expressions” on page 457](#)

## 16.2.7 Tree IA Values

Tree node events occur in the following modules: ScanPlus, RescanPlus, and Image Converter. These `<Tree>` IA Values are only valid when you create a process in CaptureFlow Designer. The `<SubTreeModified>` and `<TreeNodeModified>` IA Values hold the result of the tree events after they occur, and the results determine flow decisions. These values are not automatically cleared during processing and must be cleared as needed.

See the tree IA Values outlined in [“Tree IA Values” on page 494](#).

**Table 16-9: Tree IA Values**

<code>&lt;SubTreeModified&gt;</code>	<p>Indicates when a subtree (child) was modified:</p> <ul style="list-style-type: none"> <li>• 1: Set when a subtree (child) was added, moved, or deleted.</li> <li>• 0: Set when the subtree was not added, moved, or deleted.</li> <li>• Attribute: Integer, Output</li> <li>• Level: 1-7</li> </ul>
<code>&lt;TreeNodeModified&gt;</code>	<p>Indicates whether the node was modified:</p> <ul style="list-style-type: none"> <li>• 0: No change.</li> <li>• 1: Tree node was added.</li> <li>• 2: Tree node was moved.</li> <li>• Attribute: Integer, Output</li> <li>• Level: 0-7</li> </ul>

### Related Topics

[“Programming Reference—Expressions” on page 457](#)

## 16.2.8 UimData IA Value

The UimData IA Value serves to store a document type definition. This definition includes a collection of named fields and arrays of fields that are populated with data when scanning a real document in production.

The data stored in UimData fields can be used in Designer expressions, for instance, in Standard Export profiles. UimData cannot be accessed in CaptureFlow Designer expressions.

The table that follows includes the UimData paths that can be used in Designer expressions. In all cases, *<Uimdata path>* resolves to *<Step Name>:<level>.UimData*.

**Table 16-10: UIM data paths**

UimData path	Data type	Description
<i>&lt;Uimdata Path&gt;.&lt;Field Name&gt;</i>	Field's data type	Read only. The value in the field. For array fields, the first field value is returned.
<i>&lt;Uimdata Path&gt;.&lt;Field Name&gt;.Text</i>	string	Read only. Text property of field object. This is the text value from recognition, or in UimData, or user input.
<i>&lt;Uimdata Path&gt;.&lt;Field Name&gt;.FormattedText</i>	string	Read only. Value formatted as text using view format in field validation.
<i>&lt;Uimdata Path&gt;.&lt;TableSection Name&gt;.RowCount</i>	int	Read only. Number of rows in table section.

## 16.2.9 PreUimDataImportMode IA Value

The PreUimDataImportMode IA Value is used when setting up CaptureFlows for extraction.



**Note:** This IA Value applies only to Intelligent Capture for Invoices.

**Table 16-11: PreUimDataImportMode IA Value**

IA Value	Description
PreUimDataImportMode	<p>Specifies the name of the UIM data input binding flag, which is created during runtime when sending the value in the Custom Value field to the task parameters. The available values are:</p> <ul style="list-style-type: none"> <li>• 0—Does not pass the PreUim input values to the task parameters</li> <li>• 1—Passes the PreUim input values to the task parameters</li> </ul>

## 16.3 Capture System Files and Folders

All capture system files are stored in the <<Capture System name>> folder on the Designer host machine. This folder is created for each capture system in the Designer working directory, which is C:\Users\<<username>>\My Documents \<product\_name\_and\_version>\Default by default. This location is **configurable** and can be set to a different folder in the local system or to a network file share. Using the default working directory implies that the user is granted the <Write> permission for the Documents folder and all of its nested folders.

The capture system folder is created with a predefined structure of subfolders. “**Capture system directory structure**” on page 496 describes the file and folder structure of a capture system.



### Caution

The GlobalData folder is reserved for use by Designer only. Storing there any files not originally generated by Designer as well as corrupted files may result in unexpected behavior. The files created for the capture system components by Designer are not meant for manual opening or editing in other applications such as Windows Explorer. Opening files in a different application may cause Designer to stop responding while attempting to rename or lock required files.

**Table 16-12: Capture system directory structure**

Folder	Description and contents
bin	Stores user-defined assemblies and accessory files to be deployed (except for the CaptureFlow Script Editor assemblies). User determines this folder contents.
GlobalData\CustomStyle	Stores customer-defined styles. The directory contains a list of <Style name>.config files.

Folder	Description and contents
GlobalData\DesktopShortCutKeys	Stores the Completion module's shortcut definitions in the DesktopShortCutKeys.config file.
GlobalData\DocumentResource	Stores image files that are loaded by a document type.
GlobalData\DocumentTypes	Stores document types. The directory contains a list of <DocType Name>.xml files.
GlobalData\DqlConnection	Stores Documentum connections as <Documentum connection name>.config files.
GlobalData\EmailConnection	Stores email connections. The directory contains a list of <Email Connection name>.config files.
GlobalData\EmailImportProfile	Stores email import profiles.
GlobalData\ExportProfile	Stores export profiles. The directory contains a list of <Profile name>.xml files.
GlobalData\FileDirectoryConnection	Stores file system connections as <File system connection name>.config files.
GlobalData\FileImportProfile	Stores file system profiles.
GlobalData\GlobalOptions	Stores miscellaneous global options available for control in Designer. The directory contains a GlobalOptions.config file.
GlobalData\ImageConversion	Stores image conversion profiles. The directory contains a list of <Profile name>.config files.
GlobalData\ImageProcessing	Stores image processing profiles. The directory contains a list of <Profile name>.xml files.
GlobalData\NamedQuery	Stores named queries. The directory contains a list of <Query name>.config files.
GlobalData\OcrProfile	Stores Standard OCR profiles. The directory contains a list of <Profile name>.config files.
GlobalData\OdbcConnection	Stores ODBC connections as <ODBC Connection name>.config files.
GlobalData\Recognition	Stores DPP project files. Each <Recognition Project name> folder in this directory contains a <Project name>.dpp file.
GlobalData\ScanImportProfile	Stores distributed capture profiles.

Folder	Description and contents
GlobalData\SystemStyle	Stores client modules styles ( <i>&lt;Style name&gt;.config</i> files) predefined in Designer.
GlobalData\XPP	Stores <i>&lt;CaptureFlow name&gt;.XPP</i> files (CaptureFlows) and files generated when compiling and installing CaptureFlows. The directory includes: <ul style="list-style-type: none"> <li>• <i>&lt;CaptureFlow name&gt;.xpp</i></li> <li>• <i>&lt;CaptureFlow name&gt;-dir</i> folder that contains IAP and IPP files</li> <li>• <i>&lt;CaptureFlow name&gt;-dir\bin\release</i> folder that contains IA value assignments compiled in Emc. InputAccel.CodeBehind. AG<i>&lt;CaptureFlow_name&gt;.DLL</i></li> </ul>
Images	Stores images for sample CaptureFlows.
IAServers.xml	The list of capture servers added in the capture system.
ScriptSource	The default directory recommended for storing script source code.

## 16.4 Sample XPP Files

The topics in this section describe the XPP samples that can be optionally included in the capture system.

### 16.4.1 AdvancedMultiDocCapture Process

`AdvancedMultiDocCapture.xpp` is a sample process similar to the [BasicMultiDocCapture sample](#). In addition to the basic multi-page capture logic, it demonstrates how the Classification and Identification modules work with Extraction and Completion. Also, it shows how the recognition project templates can be configured for multi-page extraction, and how profile scripting can be used for table events and operations, for showing hidden sections. The sample also shows how the Production Auto Learning service can be used.

The `AdvancedMultiDocCapture` process includes the steps outlined in the following table.

**Table 16-13: AdvancedMultiDocCapture sample process: included steps**

Step name	Module name (trigger level)	Description
<b>ScanPlus</b>	ScanPlus (7)	Creates a batch and imports or scans all pages into it. All images are located in the local folder <code>... \&lt;capture system name&gt;\Images\</code> . The step scans images from subfolders <code>\life insurance</code> , <code>\statements</code> , and <code>\production auto learning</code> .
<b>ImageProcessor</b>	ImageProcessor (0)	Removes overscanned and deskewed pages, detects patch codes, and normalizes images to 300 <i>DPI</i> . Processing is based on the <b>BasicCleanup</b> sample profile.
<b>Classification</b>	Classification (7)	Added for splitting and merging documents, and for choosing the document type.
<b>Identification</b>	Identification (7)	Added for manual population of fields with data rubberbanded from the image of the page.
<b>Extraction</b>	Extraction (1)	Extracts image data the multi-page structured documents. The step uses the <b>Advanced</b> sample recognition project. The extracted fields are defined in the document type assigned at the <b>Classification</b> step. <b>Extraction</b> can use the sample document types <b>AR Life Insurance</b> , <b>AR Invoice</b> , and <b>AR Policy Statement</b> .
<i>Conditional branch:</i>		
The following steps run the Completion module. Each step is intended for particular operator activities. The steps use the <b>Advanced</b> sample recognition project.		
<b>Validate</b>	Completion (1)	The operator can fix the fields that have validation problems, and skip the rest.

Step name	Module name (trigger level)	Description
<b>SecureValidate</b>	Completion (1)	Same as <b>Validate</b> but the operator cannot view the page.
<b>Supervisor</b>	Completion (1)	The operator can do all above. This step is used as a final check for documents that still have work items before export.
<i>End of branch</i>		
<b>ImageConverter</b>	ImageConverter (1)	Converts single-page image files into a multi-page <i>PDF</i> file. Conversion is based on the <b>Page Images to PDF</b> sample profile.
<b>StandardExport</b>	StandardExport (7)	Creates a PDF file for each document and a summary <i>CSV</i> file that references each PDF file. Exports the output files to the file system. Export is based on the <b>MultiDocs to PDF and CSV</b> sample profile.
<b>Collector</b>	Collector (1)	Collects data for Production Auto Learning.

The `AdvancedMultiDocCapture.txt` and `BasicMultiDocCapture.txt` files located in the `...\<capture system name>\Readme\` directory on your local disk contain the detailed instructions on how to study, customize, and debug the `AdvancedMultiDocCapture` sample process.

## 16.4.2 AutoIndexing Process

`AutoIndexing.xpp` is a sample that demonstrates how bitonal and color image processing can be combined into a single **ImageProcessor** step by using profile scripting to execute a context-dependent set of image filters. This sample also teaches you how document types can be assigned using IA values and how the profile scripting created for the Completion module uses named queries to look up values and populate document fields. Finally, this sample demonstrates how to use Image Converter to generate output files for the specific export modules.

The `AutoIndexing` process includes the steps outlined in the following table.

**Table 16-14: AutoIndexing sample process: included steps**

Step name	Module name (trigger level)	Description
<b>ScanPlus</b>	ScanPlus (7)	Creates a batch and imports or scans pages into it. The step scans images from the \AutoIndexing subfolder located in the local folder ... \<capture system name>\Images\.
<b>ImageProcessor</b>	ImageProcessor (0)	Handles bitonal and color images using the profile scripting to execute a context-dependent set of image filters. Processing is based on the <b>Color and Bitonal Scripted Cleanup</b> sample profile. The script ensures that color processing is only done to color input images. After converting images to bitonal, the script applies barcode detection to all images.
<b>SplitAndDeleteBlanks</b>	Multi (7)	Splits a document wherever a barcode is detected, and deletes blank pages.
<b>DeleteEmptyDocs</b>	Multi (7)	Detects and deletes empty documents.
<b>Desktop</b>	Completion (1)	Added to demonstrate how scripting uses named queries to look up values and populate document fields. The step uses the <b>Default</b> sample recognition project and the <b>AutoIndexing</b> document type. The script uses the <b>LookupCustomerRecord</b> sample query to fetch data from the <b>AutoIndexing.xls</b> database.
<b>ImageConverter</b>	ImageConverter (1)	Converts single-page image files to multi-page TIFF files. Conversion is based on the <b>Page Images to Multipage TIFF</b> sample profile.

Step name	Module name (trigger level)	Description
<b>StandardExport</b>	StandardExport (1)	Exports a multi-page TIFF and a text file for each document. The output files are exported to the file system folder that uses the name of the batch. Export is based on the <code>AutoIndex to Multipage TIFF and Text</code> sample profile.
<b>DocumentumAdvancedExp</b>	OpenText Documentum Advanced Export (1)	Exports documents to Documentum if the Desktop operator has picked it for export.

The `AutoIndexing.txt` file located in the `... \<capture system name>\Readme\` directory on your local disk contains the detailed instructions on how to create and deploy profile scripting for the Completion module and how to configure all steps. It also describes the expected result for each step when debugging the `AutoIndexing` sample process.

### 16.4.3 BasicMultiDocCapture Process

`BasicMultiDocCapture.xpp` is a sample process that demonstrates multi-page document extraction and navigation, secure validation, and simple routing. The sample includes profile scripting that shows how to prepare resources for later use, how to use a named query for database lookup, how to pre-populate fields, how to open a popup dialog, how to change labels, and others.

The `BasicMultiDocCapture` process includes the steps outlined in the following table.

**Table 16-15: BasicMultiDocCapture sample process: included steps**

Step name	Module name (trigger level)	Description
<b>ScanPlus</b>	ScanPlus (7)	Creates a batch and imports or scans all pages into it. The step scans images from subfolders <code>\invoices</code> and <code>\life insurance</code> located in the local folder <code>... \&lt;capture system name&gt;\Images\</code> .

Step name	Module name (trigger level)	Description
<b>ImageProcessor</b>	ImageProcessor (0)	Removes overscanned and deskewed pages, detects patch codes, and normalizes images to 300 <i>DPI</i> . Processing is based on the <b>BasicCleanup</b> sample profile.
<b>ManualClassify</b>	Completion (2)	The operator can split and merge documents and assign the document types. The step uses the <b>Default</b> sample recognition project.
<i>Conditional branch:</i>		
<b>Extraction</b>	Extraction (1)	Extracts the multi-page structured documents. The step uses the <b>Default</b> sample recognition project. The extracted fields are defined in the document type assigned at the <b>ManualClassify</b> step. <b>Extraction</b> can use the sample document types <b>Life Insurance</b> and <b>Invoice</b> . For both document types, profile scripting provides extended functionality.
<b>KeyFormImage</b>	Completion (1)	The operator can populate index fields with values rubberbanded from the image. The step uses the <b>Default</b> sample recognition project.
<i>End of branch</i>		
<i>Conditional branch:</i>		
The following steps run the Completion module. Each step is intended for particular operator activities. The steps use the <b>Default</b> sample recognition project.		
<b>Validate</b>	Completion (1)	The operator can fix the fields that have validation issues, and skip the rest.
<b>SecureValidate</b>	Completion (1)	Same as <b>Validate</b> but the operator cannot view the page.

Step name	Module name (trigger level)	Description
<b>Supervisor</b>	Completion (1)	The operator can do all above. This step is used as a final check for documents that still have work items before export.
<i>End of branch</i>		
<b>ImageConverter</b>	ImageConverter (1)	Converts single-page image files into a multi-page <i>PDF</i> file. Conversion is based on the Page Images to PDF sample profile.
<b>StandardExport</b>	StandardExport (7)	Creates a PDF file for each document and a summary <i>CSV</i> file that references each PDF file. Exports the output files to the file system. Export is based on the MultiDocs to PDF and CSV sample profile.

The sample requires the following additional resources to be deployed on the server:

- The script file ProfileScripts.csproj located ... \ScriptSource\Profile Scripts\
- The VendorsDB database connection that uses files bin\VendorListDB.xls and bin\LabelsDB.xls.
- The named queries CheckVendorID, GetVendorList, and GetLabelList.

The BasicMultiDocCapture.txt file located in the ... \<capture system name> \Readme\ directory on your local disk contains the detailed instructions on how to study, customize, and debug the BasicMultiDocCapture sample process.

#### 16.4.4 BasicRescan Process

BasicRescan.xpp is a sample process that includes branching functionality for rescanning images that were flagged by the Completion module operator for quality assurance issues.

The BasicRescan process includes the steps outlined in the following table.

**Table 16-16: BasicRescan sample process: included steps**

Step name	Module name (trigger level)	Description
<b>Scan</b>	ScanPlus (0)	Creates a batch and imports or scans all pages into it. You can use any images from folder . . . \<capture system name>\Images\ for scanning.
<b>ImageProcessor</b>	Image Processor (0)	Improves the quality of scanned images and/or detects certain kinds of information in image data. The performed operations are defined in the attached profile. The step uses the <b>BasicCleanup</b> sample profile for image processing.
<b>ImageQualityCheck</b>	Completion (1)	Displays the scanned documents to the operator. The operator's task consists in validating the image quality and consistency of the scanned documents. Documents without flagged images proceed to the <b>StandardExport</b> step. Documents with detected issues are routed to the <b>Rescan</b> step for rescanning.
<b>Rescan</b>	RescanPlus (7)	Enables the RescanPlus operator to review the pages that are flagged by the Completion module operator for having quality or consistency issues. The operator can rescan poor-quality images, change the page orientation, insert missing images, rearrange order of documents in the batch, and delete pages. Updated images are automatically routed to the <b>ImageProcessor</b> step to complete the process flow.

Step name	Module name (trigger level)	Description
StandardExport	Standard Export (7)	Exports the validated pages as a single batch. The export formats and destinations are defined in the attached export profile. The step uses the CustomValuesOutputImage sample profile for export.

The BasicRescan.txt file located in the ... \<<capture system name>> \Readme \ directory on your local disk contains the detailed instructions on how to customize, deploy, and debug the BasicRescan sample process.

The corresponding module guides (as listed in *OpenText Intelligent Capture - Module Reference Help (ECPCORE-H-CMD)*) provide more information about configuring individual steps within a process after it has been compiled and installed.

## 16.4.5 eDocumentAndImageCapture Process

eDocumentAndImageCapture.xpp is a sample process similar to eDocumentCapture but in addition to electronic document handling it demonstrates how to work with a mixture of images, PDF files, and Microsoft Office documents (DOC, XLS, PPT), how to create searchable PDF files for archiving, and tune the Production Auto-Learning to collect mixed file types (PDF and TIFF).

The eDocumentAndImageCapture process includes the steps outlined in the following table.

**Table 16-17: eDocumentAndImageCapture sample process: included steps**

Step name	Module name (trigger level)	Description
StandardImport	Standard Import (7)	Creates a batch and imports files from folder c:\import \ into it. These files should be copied there from the local folder ... \<capture system name> \Images \ eDocumentAndImageInvoices. Processing is based on the eDocumentAndImageImport sample profile.

Step name	Module name (trigger level)	Description
<b>ScanPlus</b>	Scan Plus (7)	Creates a batch and imports or scans all pages into it. The step scans images from subfolder <code>\eDocumentAndImageInvoices</code> in the local folder <code>... \&lt;capture system name&gt; \Images\</code> . There is an image filter to import only image files (other files should be imported by <b>StandardImport</b> ).
<i>Conditional branch:</i>		
<b>ImageConverter_Split</b>	Image Converter (1)	Converts input Microsoft Office files into PDF format and splits multi-page PDF files to single-page PDF files. Besides, defines a type of content (text or mixed) in <code>Level0_OutputContentType</code> . Processing is based on the <b>Split to PDF pages</b> sample profile for electronic documents and <b>Split to TIFF pages</b> for image files.
<i>End of branch</i>		

Step name	Module name (trigger level)	Description
<b>StandardOCR</b>	Standard OCR (0)	Depending on the content type: extracts text content from text PDF files ( <b>Text</b> profile) or extracts text and OCR content from image-based or mixed PDF files ( <b>TextAndImage</b> profile). Saves extracted content into the <b>OCRDataCache</b> IA value and creates output searchable PDF files for images and mixed PDF files.
<b>Classification</b>	Classification (7)	Selects templates for input files using <b>OCRDataCache</b> content for text-based classification types, splits and merges document structure according to pre-index field value. The step uses <b>eDocumentAndImage</b> sample recognition project.
<i>Conditional branch:</i>		

Step name	Module name (trigger level)	Description
<b>Identification</b>	Identification (7)	Added for manual classification of input pages, validation of pre-index fields, and marking a page by the <b>ProcessAsImage</b> flag for performing OCR. If the <b>ProcessAndImage</b> page flag is set, then the flagged pages are routed to the Standard OCR module to repeat processing with <b>Image</b> profile. Unflagged pages are routed to the <b>Classification</b> step to be combined back into a single batch. The step uses the <b>eDocumentAndImage</b> sample recognition project.
<i>End of branch</i>		
<b>Extraction</b>	Extraction (1)	Extracts fields from OCRDataCache content of input pages according to their template's zones or Free Form rules. The step uses <b>eDocumentAndImage</b> sample recognition project. The extracted fields are defined in the <b>eDocumentAndImageInvoice</b> document type. Also, it shows textual anchor search for shifted fields and using OCR engine exceptions for barcode recognition.
<i>Conditional branch:</i>		

Step name	Module name (trigger level)	Description
<b>Validate</b>	Completion (1)	Operator fixes the fields that have validation issues and fulfills empty fields by basic rubberbanding based on the OCRDataCache content. Operator's input is used for Production Auto-Learning.
<i>End of branch</i>		
<b>Collector</b>	Collector (1)	Collects data for Production Auto-Learning.
<b>ImageConverter_Merge</b>	Image Converter (1)	Converts single-page searchable PDF files into a multi-page searchable PDF file. Conversion is based on the <b>Merge Pages to PDF</b> sample profile.
<b>StandardExport</b>	Standard Export (1)	Creates a text file for each document which contains some extracted fields. Exports the output text files and appropriate merged searchable PDF files to the file system into a separate folder for each template code. Export is based on the <b>eDocument to PDF and Text</b> sample profile.

The `eDocumentAndCapture.txt` file located in the `... \<capture system name> \Readme\` directory on your local disk contains the detailed instructions on how to study and customize the **eDocumentAndImageCapture** sample process.

## 16.4.6 eDocumentCapture Process

eDocumentCapture.xpp is a sample process that demonstrates handling of electronic documents based on their content type (textual, image, or mixed content).

The primary goals of electronic document handling are:

- eliminating OCR errors caused by rasterizing and performing OCR;
- improving performance by elimination of image rasterization from electronic documents and performing OCR.

The eDocumentCapture process includes the steps outlined in the following table.

**Table 16-18: eDocumentCapture sample process: included steps**

Step name	Module name (trigger level)	Description
ReadFromFolder	Standard Import (7)	Creates a batch and imports files from folder c:\import\ into it. These files should be copied there from the local folder ...\ <code>&lt;capture system name&gt;</code> \Images\ eDocumentInvoices. Processing is based on the <b>eDocument Import</b> sample profile.
SplitToPdfPages	Image Converter (1)	Converts input Microsoft Office files into <i>PDF</i> format and splits multi-page PDF files to single-page PDF files. Besides, defines a type of content (text or mixed) in <code>Level0_OutputContentType</code> . Processing is based on the <b>Split to PDF pages</b> sample profile.

Step name	Module name (trigger level)	Description
<b>ExtractFullPageContent</b>	Standard OCR (0)	Depending on the content type: extracts text content from text PDF files ( <b>Text</b> profile) or extracts text and OCR content from image-based or mixed PDF files ( <b>TextAndImage</b> profile). Saves extracted content into the <b>OCRDataCache</b> IA value.
<b>ClassifyPages</b>	Classification (7)	Selects templates for input files using <b>OCRDataCache</b> content for text-based classification types. Assigns unclassified pages as "unknown" without a document type. The step uses <b>eDocuments</b> sample recognition project.
<b>ExtractFields</b>	Extraction (1)	Extracts fields from <b>OCRDataCache</b> content of input pages according to their template's zones. The step uses <b>eDocuments</b> sample recognition project. The extracted fields are defined in the <b>eDocumentInvoice</b> document type.
<i>Conditional branch:</i>		
<b>Validate</b>	Completion (1)	Operator fixes the fields that have validation issues and skips the rest fields. Also, operator can use basic rubberbanding based on the <b>OCRDataCache</b> to improve accuracy and increase performance.

Step name	Module name (trigger level)	Description
<i>End of branch</i>		
SaveDocTypeFieldsToHtml	Standard Export (1)	Creates a HTML file for each document which contains some extracted fields. Exports the output files to the file system into a separate folder for each template code. Export is based on the <b>eDocument to HTML</b> sample profile.

The eDocumentCapture.txt file located in the `... \<capture system name> \Readme\` directory on your local disk contains the detailed instructions on how to study and customize the **eDocumentCapture** sample process.

### 16.4.7 ExpenseCollate Process

ExpenseCollate.xpp is a sample process that handles the expense documents submitted by employees. The documents can be scanned or imported using the ScanPlus module. Alternatively, you can use the Standard Import module to import documents from the scanner, from the specified network location, and/or from the mail server. Then the process extracts image data into fields, performs complex data handling using the .NET code, and exports the expense report to each employee in *CSV* and *PDF* format.

The purpose of this sample is to demonstrate how to create a batch using the Intelligent Capture Web Client (distributed capture of scanned pages). The sample also shows how to use the .NET Code module to set configuration values, read and write IA values, and move batch nodes. You can also learn how to debug the .NET code, how to use basic zonal extraction into fields, how to configure NuanceOCR to auto-rotate a page and create a searchable PDF per page, and how to combine PDF pages together and export them to CSV format.

The ExpenseCollate process includes the steps outlined in the following table.

**Table 16-19: ExpenseCollate sample process: included steps**

Step name	Module name (trigger level)	Description
<b>Scan</b>	ScanPlus (7)	Creates a batch and imports or scans pages into it. The sample images are located in the <code>\expense</code> subfolder within the local folder <code>...\<i>capture system name</i>\Images\</code> .
<b>StandardImport</b>	Standard Import (7)	Alternatively, creates a batch and imports pages from multiple sources specified in the sample import profiles: <b>EMISampleProfile</b> (mailbox), <b>MDWSampleProfile</b> (root folder), <b>WebClientSample</b> (distributed capture).
<b>ImageProcessor</b>	Image Processor (0)	Removes overscanned and deskewed pages, detects patch codes, and normalizes images to 300 <i>DPI</i> . The step uses the <code>BasicCleanup</code> sample profile for image processing.
<b>FullPageOCRToPDF</b>	NuanceOCR (0)	If an image in the <code>\expense</code> folder is rotated, this step applies auto-rotation. The step outputs a text-based PDF file per page.
<b>ZoneExtraction</b>	Extraction (1)	Extracts image data into fields that are defined in the <code>Expense Report</code> sample document type. Rotated images come from the <b>FullPageOCRToPDF</b> step and the rest come from <b>ImageProcessor</b> . Zonal extraction is configured in the <code>Default</code> recognition project.
<b>Desktop</b>	Completion (1)	Documents extracted with issues are routed to the Completion operator for manual processing. The operator's task consists in populating and validating all document fields.

Step name	Module name (trigger level)	Description
<b>CollateExpensePages</b>	.NET Code (7)	Rearranges the batch tree so there is one folder per employee, and calculates various custom values. The sample script code is located on the local disk in the following folder: <code>... \&lt;capture system name&gt; \ScriptCode\CodeModule\CodeModuleSample\ExpenseCollate</code> . You can use the pre-built <code>Custom.CodeModule.Sample.dll</code> as well.
<b>CombinePDF</b>	Image Converter (2)	Merges the text-based PDF files (per page) into one PDF at the folder level. The step uses the <code>CombinePagePDF</code> sample profile.
<b>StandardExport</b>	Standard Export (7)	Exports a CSV report for each employee to show their total expenses. The CSV file includes a file path of the combined PDF file containing all expense report pages. The step uses the <code>ExpenseReport to PDF and CSV</code> sample profile for export.

In addition to the basic logic of the `ExpenseCollate` sample process, Designer lets you install `ExpenseCollateAdvanced` sample. `ExpenseCollateAdvanced` demonstrates how to use conditional value assignment to set the document type to use, and uses the advanced AR Expense Report document type to show how to use tables.

The `ExpenseCollate.txt` and `ExpenseCollateAdvanced.txt` files located in the `... \<capture system name> \Readme \` directory on your local disk contain the detailed instructions on handling these processes. The `ExpenseCollate.txt` file provides information on how to create and deploy the .NET Code assembly and how to customize the recognition project and the document type. It also describes what results are expected for each step when debugging the `ExpenseCollate` process.

The corresponding module guides (as listed in *OpenText Intelligent Capture - Module Reference Help (ECPCORE-H-CMD)*) provide more information about configuring individual module steps within a process after it has been compiled and installed.

## 16.4.8 WSInputRescan Process

`WSInputRescan.xpp` is a sample process that demonstrates how to use the `WebServicesInput` module with the client-side scripting.

Initially, documents are scanned using `ScanPlus` and exported to `Documentum`.

If configured to use scripting, the **DocumentumExport** step:

- Remembers the `dm_document` objects that will be exported
- Exports documents to `Documentum`
- Allows you to manually specify rescan settings (such as pages, reasons, and other)
- Uses the `Web Services Input` module to make a web call for rescan

When triggered, the **WebServicesInput** step creates a batch and uses client-side scripting to convert the *SOAP XML* request to batch IA values. The **RescanPlus** step processes input images based on their rescan settings. The rescanned pages are exported to the original documents in `Documentum`.

The `WSInputRescan` process includes the steps outlined in “*WSInputRescan sample process: included steps*” on page 516.

**Table 16-20: WSInputRescan sample process: included steps**

Step name	Module name (trigger level)	Description
<b>ScanPlus</b>	ScanPlus (7)	Creates a batch and scans (or imports) pages into it. Input images are passed to <b>RescanPlus</b> without rescan settings.

Step name	Module name (trigger level)	Description
<b>WebServicesInput</b>	Web Services Input Rescan (custom wsinput_rescan.mdf, 7)	<p>Receives an XML (SOAP) request for rescan and creates a batch. The client-side scripting converts XML parameters <i>client_info_XML</i> and <i>pages</i> to IA values, creates batch tree nodes for levels 6 through 0, and adds images at level 0. . Input images are passed to <b>RescanPlus</b> with rescan settings.</p> <p>Client-side scripting is available in folder ... \&lt;capture system name&gt; \ScriptSource\Client-side Scripts\WSInput\Rescan\. The IARescanWS.wsd1 file defines the SOAP interface.</p>
<b>RescanPlus</b>	RescanPlus (7)	<p>Gets all images. For each page:</p> <ul style="list-style-type: none"> <li>• Performs or skips rescan according to the rescan reason</li> <li>• Assigns a processed image to <b>DocumentumExport</b> for export</li> <li>• Outputs the image to the <i>OutputImage</i> custom value</li> </ul>
<b>StandardExport</b>	Standard Export (7)	<p>Exports the page image (TIFF file) from <i>OutputImage</i> to a local folder. The step uses the <b>CustomValues OutputImage</b> sample profile for export.</p>

Step name	Module name (trigger level)	Description
DocumentumExport	OpenText Documentum Advanced Export (7)	Performs export to Documentum.  Client-side scripting is available in folder <code>... \&lt;capture system name&gt; \ScriptSource\Client-side Scripts\ DocumentumAdvancedExport\AdvancedRescan\</code> .

The pre-steps for using the `WSInputRescan` sample process are described in the `WSInputRescan.txt` file located in the `... \<capture system name>\Readme\` directory on your local disk.

The corresponding module guides (as listed in *OpenText Intelligent Capture - Module Reference Help (ECPCORE-H-CMD)*) provide more information about configuring individual module steps within a process after it has been compiled and installed.

For more information about configuring Documentum, see *OpenText Intelligent Capture - Specialized Export Modules Guide (ECPCORE-CEE)*. For an overview of the Documentum Advanced Export module functionality with the sample script, see *OpenText Intelligent Capture - Scripting Guide (ECPCORE-PSC)*.

## 16.4.9 WSOOutputScan Process

`WSOutputScan.xpp` is a sample process that uses the `WebServicesOutput` module. The `WebServicesOutput` scripting demonstrates how to populate SOAP fields with client-side scripting, as well as how to verify that IA values have been correctly populated. This sample script fills an array parameter of the request, where each array element is a structure with two fields, a string containing the PDA *XML* and a string containing the URL of the image.

The `WSOutputScan` process includes the steps outlined in “[WSOutputScan sample process: included steps](#)” on page 518.

**Table 16-21: WSOOutputScan sample process: included steps**

Step name	Module name (trigger level)	Description
ScanPlus	ScanPlus (7)	Creates a batch and imports or scans pages into it. You can scan any images from the local folder <code>... \&lt;capture system name&gt;\Images\</code> .

Step name	Module name (trigger level)	Description
SSOcr	NuanceOCR (0)	Performs an <i>OCR</i> based recognition on images. For each page, creates XML PDA and outputs the page image to the <i>OutputImage</i> custom value.
StandardExport	Standard Export (1)	Exports the page image (TIFF file) to a local folder path generated dynamically, using a regular expression. Stores the <i>URI</i> of the image in the <i>FullExportName</i> custom value.  The step uses the <b>CustomValues</b> <b>OutputImage</b> sample profile for export.
WebServicesOutput	WebServicesOutput (1)	Calls the Web Service (one call per document) and sends the operator name and an array of pages (including the page image URI and XML PDA for each page) populated by the script.  Client-side scripting is available in folder <code>... \&lt;capture system name&gt; \ScriptSource\Client-side Scripts\WSOutput \</code> . The <b>ExportImagesWithZones</b> .wsdl file defines the SOAP interface.

The `WSOutputScan.txt` file located in the `... \<capture system name>\Readme\` directory on your local disk contains the brief instructions on how to configure and use the `WSOutputScan` process.

The corresponding module guides (as listed in *OpenText Intelligent Capture - Module Reference Help (ECPCORE-H-CMD)*) provide more information about configuring individual module steps within a process after it has been compiled and installed.

## 16.5 Deployment Utility

The `DeploymentUtility.exe` tool can be used as an alternative to **System > Deployment** functionality for uploading capture system files to the selected capture server from the command line. This utility can also be used to download the capture system components from the specified capture server to the local folder on the client machine. It enables you to back up, update, publish, and move metadata from one Intelligent Capture system to another (for example, from a development system to a production system).



**Note:** Download works for all components, except XPP.



### Example 16-2: Command line format

```
DeploymentUtility.exe [-login:<username>,<password>@<IA_server> [-configurations:<list>
or all][-solution:<solution_name>][-customfiles:<comma separated list of files>][-
overwrieworkflow: <true|false>]][-argumentsfile:<file_name>][-solutionpath:<capture
solution path>][-downloadfromserver:<true|false>][-help]
```



“[Deployment Utility: Command line interface](#)” on page 520 describes the `DeploymentUtility.exe` command line interface.

**Table 16-22: Deployment Utility: Command line interface**

Argument	Description	Example
-help	Displays command line arguments help text.	<code>DeploymentUtility.exe -help</code>
-login	Argument to provide user name, password and the capture server to connect to.	

Argument	Description	Example
-configurations	<p>Argument to provide a list of comma separated configurations to deploy or download.</p> <p>If set to "all", deploys/downloads all configuration files.</p> <p>Values: DocumentTypes, ImageProcessing, NamedQuery, ExportProfile, OdbcConnection, OcrProfile, EmailConnection, CustomStyle, GlobalOptions, DesktopShortCutKeys, DocumentResource, ImageConversion, XPP, SystemStyle, DqlConnection, FileImportProfile, ScanImportProfile, EmailImportProfile, FileDirectoryConnection, AXConnection, CmisConnection, CSConnection, SnapMatchConnection, RecognitionModelData</p>	<p>All capture system files will be deployed to the ia75-w12r2-srv server:</p> <pre>-login:corp\user01, user01@ia75-w12r2-srv -configurations:all</pre> <p>Only document types, ODBC connections, and named queries defined in the capture system will be deployed to the ia75-w12r2-srv server:</p> <pre>-login:corp\user01, user01@ia75-w12r2-srv -configurations:Docu mentTypes, OdbcConnection, NamedQuery</pre>
-solution	Argument to provide the capture system (solution) name. Default value is default.	
-solutionpath	Argument to provide the path to the capture system on the client machine. Default value is the default path of the "default" capture system.	
-customfiles	Argument to provide a comma separated list of custom files (.NET assemblies and others) to deploy. Applicable for publishing only.	

Argument	Description	Example
-removecustomfiles	Argument to provide a comma separated list of custom files (.NET assemblies and others) to delete from the capture server. Ignored when -downloadfromserver is set to true.	
-overwriteworkflow	Argument to indicate whether to override the existing CaptureFlows (process) on the capture server. Applicable only when deploying components. Default value false.	
-overwriteworkflowwithbatches	Argument to specify whether to override existing workflows with batches on the IA server. Applicable only when deploying workflows and the -overwriteworkflow parameter is set to true. Default value false.	
-downloadfromserver	Argument to provide an option to download configurations from the capture server to the local file system. Default value false. XPP files are excluded from downloading.	<p>The configuration files will be saved to the \Default directory (for example, C:\Users\Administrator\Documents\<product_name_and_version>\Default\):</product_name_and_version></p> <pre>-login:corp\user01, user01@ia75-w12r2-srv -configurations:ImageProcessing -downloadfromserver:true</pre>
-<config>_filter	Argument to list configuration names separated by a semicolon (;). Optional configuration deployment parameters can be passed following the name separated by a comma (.). The currently supported configuration parameter is the workflow name for XPP.	<pre>XPP_filter:xpp1, wrkflw1;xpp2,wrkflw</pre> <p>where xpp1 is the name of xpp and wrkflw1 is the deployed name of xpp1</p>

---

Argument	Description	Example
-argumentsfile	Argument to provide a text file containing above arguments. An argument file is the alternative to providing arguments in command line.	-argumentsfile:C:\Temp\DeployConfig.txt



# Glossary

**ANSI**

American National Standards Institute

**API**

Application Programming Interface

**ASCII**

American Standard Code for Information Interchange

**BMP**

Bitmap file extension

**CSV**

Comma Separated Variable

**DLL**

Dynamic Link Library

**dpi**

Dots Per Inch

**DPP**

Dispatcher project file extension

**DQL**

Database Query Language

**HTML**

HyperText Markup Language

**HTTP**

Hypertext Transfer Protocol

**IAP**

InputAccel process file extension

**IP**

Internet Protocol

**IPP**

Integrated ProcessFlow Project

**MB**

megabyte

**MDF**

Module Definition File

**OCR**

Optical Character Recognition

**ODBC**

Open Database Connectivity

**PC**

Personal Computer

**PDF/A**

Portable Document Format Archive

**PDF417**

Portable Data File 417 barcode format

**PDF**

Portable Document Format

**PPT**

Microsoft Office PowerPoint file extension

**SOAP**

Service-Oriented Access Protocol

**SQL**

Structured Query Language

**SSL**

Secure Sockets Layer

**TIFF**

Tagged Image File Format

**TLS**

Transport Layer Security

**UIM**

Unified Index Modeling

**UNC**

Universal Naming Convention

**URI**

Uniform Resource Identifier

**XLS**

Microsoft Office Excel file extension

**XML**

Extensible Markup Language

