

# **OpenText™ Exstream™**

## **Creating Output**

Design and Production Documentation

Release 16.6.0

**OpenText™ Exstream**

**Creating Output**

Rev.: 2019-Apr-30

**This documentation has been created for software version 16.6.0.**

It is also valid for subsequent software versions as long as no new document version is shipped with the product or is published at <https://knowledge.opentext.com>.

**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 © 2019 Open Text. All rights reserved.**

Trademarks owned by Open Text.

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.

# Contents

Chapter 1: Overview of Delivering Documents to Multiple Channels .....	12
1.1 An Introduction to Delivery Methods .....	12
1.1.1 Electronic Output Drivers .....	13
1.1.2 Print Output Drivers .....	14
1.2 An Introduction to High-Volume Production .....	15
1.2.1 Overview of High-Volume Features in Exstream .....	15
1.2.2 Sorting and Bundling Capabilities .....	16
Chapter 2: Setting Up Output Properties .....	19
2.1 Output Drivers Available with Exstream .....	20
2.1.1 3211 Line Data .....	21
2.1.2 AFP .....	21
2.1.3 DOCX .....	22
2.1.4 EDGAR HTML .....	22
2.1.5 HTML .....	22
2.1.6 HTML (Email) .....	23
2.1.7 IJPDS (Inkjet) .....	24
2.1.8 Metacode .....	24
2.1.9 MIBF (Inkjet) .....	26
2.1.10 PCL .....	26
2.1.11 PDF .....	27
2.1.12 PDF/A .....	27
2.1.13 PDF/VT .....	28
2.1.14 PostScript .....	28
2.1.15 PowerPoint .....	29
2.1.16 PPML .....	29
2.1.17 RTF .....	29
2.1.18 TIFF .....	29
2.1.19 TOP .....	30
2.1.20 VDX .....	30
2.1.21 VIPP .....	30
2.1.22 VPS .....	30
2.1.23 XML .....	31
2.1.24 ZPL .....	33
2.2 Setting Up General Output Driver Configurations .....	33

2.2.1	Creating an Output Object .....	34
2.2.2	Setting the value of the queue selection variable .....	35
2.2.3	Managing output resolution in Exstream .....	37
2.2.4	Using Simplex and Duplex in Output .....	39
2.2.5	Specifying the Way Color is Produced in Output .....	40
2.2.6	Creating Pages as Images .....	44
2.2.7	Assigning a Connector to an Output Object for Testing and Production .....	45
2.2.8	Using Search Keys in Output .....	46
2.2.9	Specifying Unprintable Areas for Page Margins .....	48
2.2.10	Highlighting New Text and Images in Output .....	49
2.3	Setting Up Paper Bins .....	50
2.3.1	Setting Up a Paper Type .....	51
2.3.2	Specifying the Paper Types in Each Bin .....	53
2.3.3	Setting Up Paper Bins for Output .....	54
2.3.4	Controlling Paper Selection For a PostScript Printer .....	55
2.4	Setting Up Color Options for Output .....	59
2.4.1	Setting the Color Format .....	59
2.4.2	Setting Up Color Tables to Adjust Named Colors for Output .....	60
2.4.3	Identifying the Spot Colors Available for an Output Device .....	61
2.5	Setting Up 3211 Line Data Output .....	62
2.5.1	Setting Up Carriage Controls .....	63
2.5.2	Specifying the Maximum Number of Lines and Characters per Inch .....	63
2.5.3	Specifying Whether to Use the ASCII or EBCDIC Character Set .....	64
2.5.4	Specifying Record Length .....	64
2.6	Setting Up AFP Output .....	65
2.6.1	Setting Up Resource Names for AFP Output in the System Settings .....	66
2.6.2	Changing the Duplex Page Rotation .....	66
2.6.3	Using Knockout to Prevent the Mixing of Colors in Overlapping Objects .....	67
2.6.4	Creating an AFP Index .....	67
2.6.5	Specifying Where to Place BDT and EDT Fields in AFP Output .....	68
2.7	Setting Up DOCX Output .....	68
2.7.1	About DOCX Output .....	69
2.7.2	Application Planning Considerations for DOCX Output .....	70
2.7.3	Design Considerations for Controlling the Page Layout for DOCX Output .....	73
2.7.4	Designing Applications to Maximize the Editability of DOCX Output .....	73
2.7.5	Exstream Features Supported in DOCX Output .....	77
2.8	Setting Up EDGAR HTML Output .....	90
2.8.1	Showing HTML Revision Tags in an HTML Viewer for EDGAR HTML .....	91

2.9 Setting Up HTML Output .....	91
2.9.1 Creating Basic HTML Output .....	93
2.9.2 Creating MIME HTML Output To Include All Resources in a Single File .....	95
2.9.3 Specifying Where Styles are Included in HTML Output .....	96
2.9.4 Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output .....	97
2.9.5 Using Scalable Vector Graphic Format for Graphic Design Objects .....	99
2.9.6 Adding Hyperlinks at the Top of Each HTML Page .....	100
2.9.7 Adding Metadata (<meta> Elements) to HTML and HTML (Email) Output ..	100
2.9.8 Setting Up to Transform HTML Output .....	102
2.9.9 Avoiding File Name Conflicts with CSS Files When Using Multiple HTML Output Objects .....	105
2.10 Setting Up HTML (Email) Output .....	105
2.10.1 Creating Basic HTML (Email) Output .....	107
2.10.2 Specifying Where Styles Are Included in HTML (Email) Output .....	108
2.10.3 Creating HTML (Email) Output With an External Email Template .....	110
2.10.4 Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output .....	114
2.10.5 Adding Metadata (<meta> Elements) to HTML and HTML (Email) Output ..	115
2.10.6 Setting Up to Transform HTML Output .....	117
2.11 Setting Up IJPDS Output .....	119
2.11.1 Creating an IJPDS Index .....	120
2.11.2 Specifying a Linearization Table for Tone Correction .....	120
2.11.3 Setting Up an IJPDS Output Configuration .....	121
2.12 Setting Up Metacode Output .....	124
2.12.1 Setting Up for Metacode Output in the System Settings .....	125
2.12.2 Printing Metacode Output Offline (to a Tape Device) .....	126
2.12.3 Using Image Blocking for Metacode Output When Not Printing Offline ..	127
2.12.4 Setting Print Stream Options for Metacode Output .....	127
2.13 Setting Up MIBF Output .....	131
2.13.1 Specifying the Output Format Mode for MIBF Output .....	131
2.13.2 Specifying ICC Profiles .....	132
2.13.3 Setting Up an MIBF Printer Configuration .....	133
2.14 Setting Up PCL Output .....	135
2.14.1 Using the PCL Output Driver on z/OS .....	135
2.15 Setting Up PDF Output .....	136
2.15.1 About PDF Output .....	136
2.15.2 Applying a PDF Preset .....	137
2.15.3 Customizing the PDF Output Setup .....	138

2.15.4 Creating PDF AcroForms output .....	140
2.15.5 Adding Hyperlinks to PDF Output .....	141
2.15.6 Adding Metadata to PDF Output .....	142
2.16 Setting Up PDF/A Output .....	144
2.16.1 Specifying the Color Profile for PDF/A Output .....	145
2.16.2 Creating PDF AcroForms output .....	146
2.16.3 Adding Static Subject and Keyword Metadata to the Output File .....	147
2.17 Setting Up PDF/VT Output .....	148
2.17.1 Specifying the Color Profile for PDF/VT Output .....	149
2.17.2 Using Document Part Metadata for PDF/VT Output .....	150
2.18 Setting Up PostScript Output .....	153
2.18.1 Controlling the Type of Comments Sent to the Output Device .....	154
2.19 Setting Up PowerPoint Output .....	155
2.19.1 Creating PowerPoint Output as a Single File or with One File Per Page .....	155
2.20 Setting Up PPML Output .....	156
2.21 Setting Up RTF Output .....	156
2.21.1 Using Absolute Object Positioning .....	157
2.21.2 Setting Up Editable RTF Output .....	158
2.22 Setting Up TIFF Output .....	160
2.23 Setting Up TOP Output .....	160
2.24 Setting Up VDX Output .....	161
2.25 Setting Up VIPP Output .....	162
2.26 Setting Up VPS Output .....	162
2.27 Setting Up XML (Composed) Output .....	163
2.27.1 Applying an Encoding to XML (Composed) Output (DBCS) .....	164
2.28 Setting Up XML (Content) Output .....	165
2.28.1 Defining a Basic XML (Content) Output Object .....	165
2.28.2 Producing UTF-8 XML (Content) Output .....	169
2.29 Setting Up Multi-Channel XML Output .....	169
2.29.1 About Multi-Channel XML Output .....	170
2.29.2 Creating Basic Multi-Channel XML Output .....	172
2.29.3 Specifying How Table Cells are Separated in Plain Text Multi-Channel XML Output .....	176
2.29.4 Tagging Objects for Multi-Channel XML Output with Metadata .....	177
2.29.5 Setting Up to Create Transformed Output .....	182
2.29.6 Applying an Encoding to Multi-Channel XML Output (DBCS) .....	184

2.30 Setting Up ZPL Output .....	185
<b>Chapter 3: Optimizing output production and enhancing output files .....</b>	<b>186</b>
3.1 Optimizing the size and structure of the print stream .....	186
3.2 Optimizing text quality for output .....	188
3.2.1 Making sure fonts appear correctly in output .....	189
3.2.2 Enhancing character print quality .....	190
3.2.3 Text quality considerations in multiple queue production .....	192
3.3 Optimizing 3211 line data .....	193
3.4 Optimizing AFP output .....	194
3.4.1 How AFP resource files are created in Exstream .....	194
3.4.2 Engine switches unique to AFP output .....	195
3.5 Optimizing and enhancing HTML and HTML (Email) output .....	196
3.5.1 Designing for HTML and HTML (Email) output .....	196
3.5.2 Text Restrictions in HTML and HTML (Email) Output .....	197
3.5.3 Using tables in HTML and HTML (Email) output .....	198
3.5.4 Customizing the Application Title for HTML, EDGAR HTML, and HTML (Email) Output .....	198
3.6 Optimizing IJPDS output .....	199
3.6.1 Optimizing engine processing time for IJPDS output .....	199
3.6.2 Optimizing print quality for IJPDS output .....	200
3.6.3 Using registration and cue marks .....	201
3.7 Optimizing Metacode output .....	201
3.7.1 Designing for Metacode output .....	201
3.7.2 Creating reverse video text with named fonts .....	202
3.7.3 Using VSAM package files to enhance performance on z/OS .....	203
3.7.4 Building print resource files .....	203
3.7.5 Engine switches unique to Metacode .....	204
3.8 Optimizing MIBF output .....	204
3.8.1 Designing for MIBF output .....	204
3.8.2 Using tone correction curves to optimize ink usage .....	205
3.8.3 Producing rotated output for multiple-up sheets in MIBF output .....	205
3.9 Optimizing and enhancing PDF, PDF/A, or PDF/VT output .....	206
3.9.1 Compressing PDF files .....	206
3.9.2 Creating linearized PDF output .....	207
3.9.3 Using editable PDF forms in Exstream .....	207
3.9.4 Adding accessibility tags to PDF files for screen readers and text-to-speech converters .....	208

3.9.5 Optimizing Adobe Acrobat tool compatibility .....	210
3.9.6 Adding security to PDF files .....	210
3.9.7 Using digital IDs to digitally sign and authorize documents .....	213
3.9.8 Creating COLD PDF output .....	215
3.9.9 Controlling how bookmark entries are included in output .....	215
3.10 Optimizing PostScript output .....	216
3.10.1 Using PostScript commands supported in Exstream .....	217
3.10.2 Using named colors in PostScript and PostScript-based output .....	218
3.10.3 Using Ghostscript to view PostScript output .....	218
3.10.4 Using PostScript Encodings in DBCS Applications .....	218
3.11 Optimizing PowerPoint output .....	219
3.11.1 Designing for PowerPoint output .....	219
3.11.2 Text restrictions for PowerPoint output .....	220
3.11.3 Text size considerations for PowerPoint output .....	220
3.11.4 Using tables in PowerPoint output .....	221
3.12 Optimizing RTF output .....	222
3.12.1 Adding bookmarks to RTF output .....	222
3.12.2 Designing objects for editable RTF output .....	222
3.12.3 Defining the spelling dictionary language in RTF output .....	223
3.12.4 Viewing RTF output in other programs .....	225
3.13 Optimizing XML (Content) output .....	225
3.13.1 How XML (Content) output processes flow settings .....	225
3.13.2 How design objects are used in XML (Content) output .....	226
3.13.3 How tables are used in XML (Content) output .....	227
3.13.4 How charts and barcodes are processed in XML (Content) output .....	228
<b>Chapter 4: Managing Resources for the Print Stream .....</b>	<b>229</b>
4.1 Managing Font Resources .....	229
4.1.1 Referencing All Fonts .....	229
4.1.2 Embedding Fonts .....	230
4.1.3 Identifying the DBCS Font Characters to Include in the Package File .....	233
4.2 Managing Image Resources .....	235
4.2.1 Specifying Where to Place Static Images in the Print Stream .....	236
4.2.2 Converting Images for Optimized Viewing .....	237
4.2.3 Specifying How Image Resolution Settings Are Processed .....	238
4.2.4 Specifying the Internal Format to Use for Images .....	239
4.2.5 Using AFP Color Management Architecture Functionality with Image Resources in the Print Stream .....	241

4.2.6 Referencing External Images for PostScript and VIPP Output .....	243
4.2.7 Converting Charts and Shapes into Images (AFP Output Only) .....	244
4.2.8 Referencing and Storing Images for HTML, HTML (Email), and Multi-Channel XML Output .....	244
4.3 Using Inline Form Definition File in AFP Output .....	248
4.4 Using Overlays in Output .....	249
4.4.1 Specifying Overlay Settings .....	249
4.5 Managing Print Resources .....	249
4.5.1 Including Only Used Resources in Output .....	250
4.5.2 Managing Print Resources with a Resource Manager File .....	251
4.5.3 Storing Named Files on a PostScript-Based Output Device .....	255
<b>Chapter 5: Using Barcodes in Exstream .....</b>	<b>256</b>
5.1 Barcode Types .....	256
5.1.1 3 of 9 .....	257
5.1.2 Code 128-Based .....	257
5.1.3 DataMatrix 2D .....	259
5.1.4 EAN-8 .....	259
5.1.5 Four State .....	260
5.1.6 General Purpose .....	261
5.1.7 Interleaved 2 of 5 .....	261
5.1.8 Japanese Postal .....	262
5.1.9 Modified Plessey .....	263
5.1.10 OMR and GBR OMR .....	264
5.1.11 PDF 417 .....	265
5.1.12 POSTNET .....	265
5.1.13 QR Code .....	266
5.1.14 UPC .....	266
5.2 Creating Barcodes .....	267
5.2.1 Creating a Barcode Object in Design Manager .....	267
5.2.2 Creating a Barcode from Fonts in Designer .....	268
5.2.3 Importing a Scanned Barcode to Use in Designer .....	269
5.3 Specifying How a Barcode is Generated .....	269
5.3.1 Generating a Barcode with a Specified Font .....	270
5.3.2 Generating a Barcode by Drawing the Lines .....	270
5.4 Specifying the Placement and Appearance of a Barcode Object .....	271
5.4.1 Specifying Where the Barcode is Placed on the Page .....	272
5.4.2 Adjusting the Size of a Barcode .....	272

5.4.3 Rotating a Barcode .....	273
5.4.4 Customizing the Barcode Character Set .....	273
5.4.5 Specifying the Size of Bars and Spaces for Four State, Japanese Postal, and OMR Barcodes .....	274
5.4.6 Adding a Human-Readable Identifier to a Barcode .....	275
5.4.7 Using the BCOCA Standard (AFP Output Only) .....	277
5.4.8 Specifying 2D Barcode Properties .....	278
5.5 Specifying the Content of a Barcode Object .....	281
5.5.1 Content Type Options for Barcodes .....	282
5.6 Previewing and Printing a Barcode .....	286
5.7 Using Barcodes in Your Design .....	287
5.7.1 Adding a Barcode Object to a Page in Designer .....	287
5.7.2 Creating an Inventory or Pick List with Barcodes in a Table .....	288
<b>Chapter 6: Setting Up for High-Volume Production .....</b>	<b>290</b>
6.1 Using Inserter Objects to Add Messages and Inserts During Print Production .....	290
6.1.1 Configuring an Inserter Object .....	291
6.1.2 Including Inserts Only if the Language Matches the Customer .....	292
6.1.3 Using a Barcode to Control an Inserter Object .....	293
6.1.4 Specifying a Weight Selection Method for Carrier Envelopes .....	293
6.1.5 Specifying the Way an Output Queue Handles an Inserter Object .....	294
6.2 Using Banner Pages to Mark Breaks in Output .....	295
6.2.1 Creating a Banner Page .....	296
6.2.2 Specifying When to Add a Banner Page in Output .....	297
6.2.3 Specifying Where to Place a Banner Page in Output .....	297
6.2.4 Adding an Envelope Banner Page to Split Customer Documents into Separate Envelopes .....	298
6.2.5 Specifying the Way an Output Queue Handles Banner Pages .....	299
6.3 Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects .....	300
6.3.1 Creating a Multiple-Up Object .....	301
6.3.2 Setting Basic Options for a Multiple-Up Object in Design Manager .....	302
6.3.3 Specifying a Dynamic Paper Type for a Multiple-Up Object .....	302
6.3.4 Allowing Multiple Languages for Multiple-Up Objects .....	303
6.3.5 Adding Content to Multiple-Ups with Multiple-Up Frames in Designer .....	303
6.3.6 Using Bleeds to Prevent Unprinted Edges in Trimmed Multiple-Up Content .....	305
6.3.7 Using Imposition to Specify the Page Order for Pages in a Booklet .....	309
6.3.8 Setting the Margins for Pages in a Booklet Using the Booklet Creep Option .....	310
6.3.9 Specifying How to Order Customers on Multiple-Up Sheets .....	311

6.3.10 Specifying the Way an Output Queue Handles Multiple-Ups .....	314
6.3.11 System Variables You Can Use with Multiple-Ups .....	314
6.4 Controlling and Setting Up High Volume Production Options with Output Queues	315
6.4.1 Creating an Output Queue Object .....	316
6.4.2 Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages .....	318
6.4.3 Adding a Unique Name to Each Record in an Output File .....	321
6.4.4 Offsetting Stacks of Printed Pages with Jogging .....	322
6.4.5 Assigning a Connector to an Output Queue .....	322
6.4.6 Generating Queue Report Files for Tracking and Quality Control .....	323
6.4.7 Adding Information to the Output Manifest for Output Delivery .....	324
6.5 Adding Non-Printing Data or Production Device Controls with Search Keys .....	326
6.5.1 Using Application Search Keys .....	326
6.5.2 Using Locator Search Keys .....	332
<b>Chapter 7: Setting Up Output for Sorting and Bundling .....</b>	<b>335</b>
7.1 An Overview of the Sorting and Bundling Process .....	335
7.1.1 Timing Considerations for Variable Data .....	337
7.2 Setting Up for Pre-Sort Processing .....	339
7.2.1 Creating a Sample Sort Index File Layout .....	339
7.2.2 Creating a Sort Index File Object and Mapping the Data .....	340
7.2.3 Setting Up an Output Queue for Pre-Sort Processing .....	342
7.2.4 Using Bundling (Householding) to Group Customer Information .....	343
7.2.5 Setting Up Control Files for Pre-Sort Processing .....	345
7.3 Setting Up Post-Sort Processing .....	345
7.3.1 Setting Up an Output Queue for Post-Sort Processing .....	346
7.3.2 Creating a Post-Sort Initialization File .....	346
7.3.3 Setting Up a Control File for Post-Sort Processing .....	347
7.4 Running the Engine for Pre-Sort Processing .....	349
7.4.1 Running the Engine for Pre-Sort Processing on the z/OS Platform .....	350
7.5 Merging Sort Index Files from Multiple Pre-Sort Engine Runs .....	350
7.5.1 Using Batch Aggregation to Merge Sort Index Files from Multiple Runs of One Application .....	351
7.5.2 Using the Application Consolidator to Merge Sort Index Files from Different Applications .....	353
7.6 Sorting Files with an External Program .....	355
7.6.1 Sorting Considerations .....	356
7.7 Running the Engine for Post-Sort Processing .....	356

# Chapter 1: Overview of Delivering Documents to Multiple Channels

After you have designed your application and are ready for production, you are ready to set up output and high-volume components. Exstream supports many different output formats. From a single application, and in a single pass of the engine, you can build output files for delivery in nearly any print or electronic output format.

For enterprise document production, Exstream also provides production capabilities that maximize high-volume delivery efficiency by controlling postage weights, preparing data for postal sort, driving inserters, and householding documents. To streamline distribution, output can be routed to the appropriate queue based on the priority, number of pages, weight, or other rule-based specifications. Exstream imposes pages, creates audit (or report) files, and breaks the print stream into document and page subsets based on the queue.

This guide describes the output drivers that are available from Exstream and how to optimize your output. In addition, this guide describes the high-volume capabilities of Exstream and how to use them to support enterprise document production.

## 1.1 An Introduction to Delivery Methods

To deliver communications to your customers, you must produce output. With Exstream, you can create your customer communications for multiple outputs simultaneously using a single application design.

Exstream creates all of the resources needed to produce output in many different formats and to many different devices, so you design your application only one time. No additional changes are needed to the original design to begin producing output to additional formats, as long as the output Page Description Language (PDL) you want to use supports the design elements used.

If you have licensed the High-Volume delivery module, you can also create print streams for electronic and print output simultaneously. For example, you can create both print statements and electronic statements in one engine run. This will allow you to mail hard copies of a customer's statement to a customer, including relevant marketing materials, and then you can also upload PDF versions of the same statement to your website.

Based on the target output, Exstream builds all the necessary resources for the most efficient output processing possible. Exstream automatically creates the required images, fonts, and overlays (forms). Resources can either be created and loaded on printers manually or included in the print stream. If the resources are included in the print stream, Exstream will delete them from the printer upon print completion. Paper bin selection and paper tray control is available for printers with multiple paper types.

### 1.1.1 Electronic Output Drivers

Many businesses are moving primarily to electronic output for sending business communication and for archival purposes. Depending on the documents you produce, you might want to create electronic output for the following reasons:

- Web publishing
- Sending email
- Archiving
- Front office editing
- Creating business presentations

The following electronic output formats are supported in Exstream:

- DLF (Live documents)
- DOCX
- EDGAR HTML
- HTML
- HTML (email)
- MPW (Empower documents)
- PDF
- PDF/A
- PDF/VT
- PowerPoint
- RTF
- TIFF
- XML (composed)
- XML (content)
- XML/JSON (data)
- XML (multi-channel)

For more information about each output driver, see “[Output Drivers Available with Exstream](#)” on [page 20](#).

You must have licensed the appropriate module for the output driver associated with each output format you want to produce in Exstream.

For information about the modules required for the various output drivers in Exstream, see *Getting Started* in the Exstream Design and Production documentation.

## 1.1.2 Print Output Drivers

Exstream supports many output formats that can be printed on the print devices your business might already use. With Exstream, depending on the output format, you can produce output for multiple output devices from the same design.

Refer to your output device documentation for information about the output formats that your output device supports.

The following print output formats are supported in Exstream:

- 3211 line data
- AFP
- IJPDS
- Metacode
- MIFB
- PCL
- PostScript
- PPML
- TOP
- VDX
- VIPP
- VPS
- ZPL

For more information about each output format, see “[Output Drivers Available with Exstream](#)” on [page 20](#).

You must have licensed the appropriate module for the output driver associated with each output format you want to produce in Exstream.

For information about the modules required for the various output drivers in Exstream, see *Getting Started* in the Exstream Design and Production documentation.

## 1.2 An Introduction to High-Volume Production

The high-volume capabilities with Exstream let you process large quantities of customer documents through a high-volume printing and mailing facility, support the audit and distribution process for Internet delivery, and simultaneously produce output for different delivery mediums.

The high-volume features in Exstream let you complete the following tasks:

- Add pre-printed marketing material.
- Include return envelopes.
- Add banner pages.
- Configure barcodes.
- Use system queue variables to control output.
- Print only certain customer documents from an existing output file.
- Consolidate output from different sort runs.
- Consolidate output from different applications.
- Create booklets and brochures.

### 1.2.1 Overview of High-Volume Features in Exstream

With high-volume delivery, you can build output files for delivery through multiple print and electronic channels in a single pass of the engine. To streamline distribution, output can be routed to different queues based on the number of pages, weight, or other rule-based specifications. Exstream provides many options that help you set up the most efficient applications and reduce mailing and production costs, including setting up for sorting and bundling.

You must have licensed the High-Volume Delivery module to use high-volume capabilities in Exstream.

The objects specific to high-volume production are described in detail in this guide. The following table gives a brief overview of these objects:

#### High-volume objects available in Exstream

High-volume object	Description
Banner pages	<p>With banner pages, you can mark breaks in the output and include current or summary information about the production run. You can add banner pages at the beginning or at the end of a processing event, such as before or after each customer. This information functions as an aid to splitting and sorting the finished output.</p> <p>For more information about banner pages, see <a href="#">"Using Banner Pages to Mark Breaks in Output" on page 295</a>.</p>
Barcodes	<p>Barcodes let you complete a variety of tasks in production, including controlling inserters, tracking equipment, and marking output for mailing purposes. Exstream supports many barcode types, all of which can be customized, though they must follow necessary constraints for the equipment used to read them.</p> <p>For more information about barcodes, see <a href="#">"Using Barcodes in Exstream" on page 256</a>.</p>
Inserters	<p>With inserters, you can control production equipment called inserter devices so that pre-printed messages or envelopes are added to output at print time. You can also set up an inserter object in Exstream so that if an inserter device runs out of pre-printed messages, you can instruct the Exstream engine to create a replacement message as a graphic image on a page.</p> <p>For more information about inserters, see <a href="#">"Using Inserter Objects to Add Messages and Inserts During Print Production" on page 290</a>.</p>
Output queues	<p>In Exstream, an output queue functions as a container to which you assign an output driver and the objects you use for high-volume production, such as inserter objects and multiple-ups. You also set up an output queue to configure high-volume production settings, including where to add breaks in the output and setting up queue reports for tracking and quality control.</p> <p>For more information about output queues, see <a href="#">"Controlling and Setting Up High Volume Production Options with Output Queues" on page 315</a>.</p>
Multiple-Ups	<p>Multiple-ups let you design several logical pages and produce those pages so that they print on one sheet, whether you are producing electronic or print output. Multiple-ups are most commonly used for print output to maximize print efficiency and to create booklets. You can also use multiple-ups in electronic output for archiving purposes or to easily send to multiple locations for printing.</p> <p>For more information about multiple-ups, see <a href="#">"Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects" on page 300</a>.</p>
Search keys	<p>Search keys are versatile features in Exstream that let you add comments, searchable tags, instructions to print devices, and more to both electronic and print output. For example, you can use search keys to add comments to PDF output, or you can add information to control stapling functionality with an AFP output device.</p> <p>For more information about search keys, see <a href="#">"Adding Non-Printing Data or Production Device Controls with Search Keys" on page 326</a>.</p>

## 1.2.2 Sorting and Bundling Capabilities

Exstream provides the opportunity to set up options for sorting data and to place multiple customer communications to the same address into one envelope (also called "householding"). After sorting the data stream with a third-party program, you can then complete the sorting and bundling process in Exstream. This process, called sorting and bundling, provides more

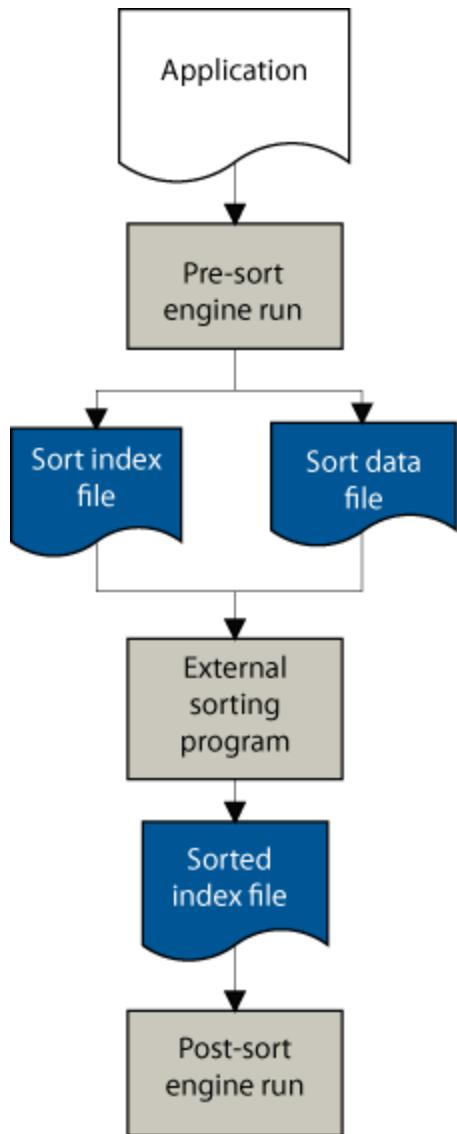
flexibility in high-volume production processing and allows you to simplify processes and improve overall production performance.

The sorting and bundling process includes the following steps:

1. Setting up for pre-sort processing
2. Setting up for post-sort processing
3. Running the engine for pre-sort processing
4. Merging sort index files from multiple pre-sort engine runs (optional)
5. Sorting files with an external program
6. Running the engine for post-sort processing

The following graphic provides a basic overview of the sorting and bundling process:

#### Sorting and bundling process



For more information about sorting and bundling, see [“Setting Up Output for Sorting and Bundling” on page 335](#).

# Chapter 2: Setting Up Output Properties

This chapter gives a brief overview of each output driver available from Exstream and provides the information you need to set up basic output properties.

To set up for output, you create output objects. Output objects are Library objects to which you assign your output driver. After you assign an output driver to an output object, you can customize the settings specific to output production.

This chapter discusses the following topics:

- [“Output Drivers Available with Exstream” on the next page](#)
- [“Setting Up General Output Driver Configurations” on page 33](#)
- [“Setting Up Paper Bins” on page 50](#)
- [“Setting Up Color Options for Output” on page 59](#)
- [“Setting Up 3211 Line Data Output” on page 62](#)
- [“Setting Up AFP Output” on page 65](#)
- [“Setting Up DOCX Output” on page 68](#)
- [“Setting Up EDGAR HTML Output” on page 90](#)
- [“Setting Up HTML Output” on page 91](#)
- [“Setting Up HTML \(Email\) Output” on page 105](#)
- [“Setting Up IJPDS Output” on page 119](#)
- [“Setting Up Metacode Output” on page 124](#)
- [“Setting Up MIBF Output” on page 131](#)
- [“Setting Up Multi-Channel XML Output” on page 169](#)
- [“Setting Up PCL Output” on page 135](#)
- [“Setting Up PDF Output” on page 136](#)
- [“Setting Up PDF/A Output” on page 144](#)
- [“Setting Up PDF/VT Output” on page 148](#)
- [“Setting Up PostScript Output” on page 153](#)
- [“Setting Up PowerPoint Output” on page 155](#)
- [“Setting Up PPML Output” on page 156](#)

- “Setting Up RTF Output” on page 156
- “Setting Up TIFF Output” on page 160
- “Setting Up TOP Output” on page 160
- “Setting Up VDX Output” on page 161
- “Setting Up VIPP Output” on page 162
- “Setting Up VPS Output” on page 162
- “Setting Up XML (Composed) Output” on page 163
- “Setting Up XML (Content) Output” on page 165
- “Setting Up ZPL Output” on page 185

For information about optimizing the print stream, optimizing print quality, and other features to enhance output quality, see “[Optimizing output production and enhancing output files](#)” on [page 186](#).

For information about managing output resources, see “[Managing Resources for the Print Stream](#)” on [page 229](#).

## 2.1 Output Drivers Available with Exstream

Exstream offers many output drivers that let you produce many different output formats. The following sections describe the output drivers and give an introduction to their capabilities.

- “3211 Line Data” on the next page
- “AFP” on the next page
- “DOCX” on page 22
- “EDGAR HTML” on page 22
- “HTML” on page 22
- “HTML (Email)” on page 23
- “IJPDS (Inkjet)” on page 24
- “Metacode” on page 24
- “MIFB (Inkjet)” on page 26
- “PCL” on page 26
- “PDF” on page 27
- “PDF/A” on page 27

- “[PDF/VT](#)” on page 28
- “[PostScript](#)” on page 28
- “[PowerPoint](#)” on page 29
- “[PPML](#)” on page 29
- “[RTF](#)” on page 29
- “[TIFF](#)” on page 29
- “[TOP](#)” on page 30
- “[VDX](#)” on page 30
- “[VIPP](#)” on page 30
- “[VPS](#)” on page 30
- “[XML](#)” on page 31
- “[ZPL](#)” on page 33

For information about the modules required for the various output drivers in Exstream, see *Getting Started* in the Exstream Design and Production documentation.

### 2.1.1 3211 Line Data

The 3211 line data output driver produces the necessary resources to support the print stream for 3211 impact line printers. Exstream supports many features of 3211 line data output, including record separation, text overflow, ANSI or machine carriage controls, ASCII or EBCDIC output character sets, number of lines per inch, number of characters per inch, blank line removal, and line overprinting.

Line printers produce output with one font, printing only from the top to the bottom of the page and one line at a time. This technology is frequently used for high-volume printing.

### 2.1.2 AFP

The AFP output driver produces the necessary resources to support the print stream for IBM's Advanced Function Presentation (AFP) format printers.

Exstream supports many features of AFP output, including the following:

- Image Output Content Architecture (IOCA), Bar Code Object Content Architecture (BCOCA), and Image Format 1 (IM1) formats
- Raster font creation and inclusion
- Graphic Object Content Architecture (GOCA) shading and drawing components

- Tagged Logical Elements (TLEs)
- No Operation (NOP) record creation
- Full-color, AFP OCA, or black-and-white output
- Three levels of optimization

### 2.1.3 DOCX

The DOCX output driver produces Open Office XML files that can be viewed and edited in Microsoft Word 2007 and later. In Exstream, DOCX output is produced as content that can be easily distributed and opened by recipients with Microsoft Word or so that recipients can make edits or changes to the content to meet your business needs.

### 2.1.4 EDGAR HTML

The EDGAR HTML output driver produces a subset of HTML output that is used to file forms with the Electronic Data Gathering, Analysis, and Retrieval (EDGAR) system used by the United States Securities and Exchange Commission (SEC). EDGAR performs automated collection, validation, indexing, acceptance, and forwarding of submissions by companies and others who are required by law to file forms with the SEC.

The SEC Filing module offered by Exstream limits standard HTML features to fit SEC specifications.

For more information about EDGAR, see <http://edgar.sec.gov/edgar/aboutedgar.htm>.

### 2.1.5 HTML

The HTML output driver produces HTML files that use Cascading Style Sheets (CSS) and GIF or Scalable Vector Graphics (SVG) images. These files are used for web-based communications.

For more information about using cascading style sheets (CSS) to control the final appearance of container design output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Exstream provides HTML output that supports text and images as output for presentation on the Internet, intranets, or other electronic networks. With HTML output, you can produce web pages and other browser-based communications. You can deploy Exstream-generated HTML pages on the web, in eLearning sessions, for online help programs, and in other communications that use standard web browsers.

When you deliver container designs in HTML output, you can specify an XSL file to transform the HTML output during production. Transforming the HTML lets you customize or add content

to the output. For example, you can include JavaScript or other custom content that you would normally include as a standard part of a web page.

The HTML content that you produce from Exstream can be HTML5, HTML 4.01 Transitional, or MIME HTML. HTML5 (the default selection for a new application) lets you produce output to which you can, using external systems, add any of the new features that are specified by the HTML5 recommendation. However, you can specify HTML 4.01 Transitional output in order to support legacy systems. You can specify MIME HTML output if you want to include all of the resources for the HTML output in a single file.

**Note:** Because there are differences in the way some web browsers display different versions of HTML, some web browsers might display HTML5 output and HTML 4.01 Transitional output that were produced from the same design in an application with slightly different positioning of objects or spacing between objects.

HTML5 uses Cascading Style Sheets, level 3 (CSS3) for styles and object positioning, and HTML 4.01 Transitional uses Cascading Style Sheets, level 2 (CSS2) for styles and, optionally, object positioning. Also keep in mind that HTML5 output provides better separation of presentation and content than HTML 4.01 Transitional output. With HTML5 output, all styles are defined in an external CSS file or within the `style` element in the HTML itself, depending on how the HTML output object is configured. Individual elements use class selectors to apply styles. This method for defining styles differs from HTML 4.01 Transitional output, which in many cases uses the `style` attribute of individual elements to define and apply styles. The separation of styles in HTML5 output lets you more easily change styles to adjust the look and feel of your existing output, without repackaging your application or producing new output.

For more information about selecting HTML5 or HTML 4.01 Transitional output, see “[Creating Basic HTML Output](#)” on page 93.

For more information about producing MIME HTML output, see “[Creating MIME HTML Output To Include All Resources in a Single File](#)” on page 95.

HTML 4.01 Transitional output from Exstream is compatible with extensible HTML (XHTML) only when delivering container designs in the output.

For more information about container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## 2.1.6 HTML (Email)

The HTML (email) output driver produces HTML files that are used for email-based communications. This driver lets you produce HTML emails and (optionally) a plain text version of the same output.

You can produce HTML (email) output only when delivering container designs in the output.

For more information about container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

When you are delivering container designs in HTML (email) output, you can specify an XSL file to transform HTML (email) output during production. Transforming the HTML (email) lets you customize or add content to the output. For example, you can include custom content that you would normally include as a standard part of an email.

## 2.1.7 IJPDS (Inkjet)

The IJPDS output driver produces the necessary resources to support the print stream for proprietary Inkjet Printer Data Stream (IJPDS) file format for Kodak digital printers. IJPDS output supports black-and-white, highlight color, and full-color modes for high-volume, transaction-oriented production. Keep in mind that for DBCS applications, IJPDS does not support rotated text. The engine moves all rotated text in your design to a horizontal position during the production run for a DBCS application.

The IJPDS driver lets you use Unicode character sets to reach an expanded audience with your communications. Inkjet printers also provide you with many color manipulation options and support high-speed, full-color production.

IJPDS is a proprietary HP Scitex Corporation file format for inkjet print streams.

An IJPDS file produced in Exstream contains the following information:

- Control data and commands that identify the characteristics of the data to be printed and the position of the printed image on the document
- Image data, consisting of the text and graphics to be printed

## 2.1.8 Metacode

The Metacode output driver produces the necessary resources to support the print stream for Xerox high-speed laser printers. The Metacode output driver supports on-the-fly image and font creation; highlight color, halftone, and black-and-white color modes; and optimization. Exstream also supports kerning and tracking for fonts and enhanced shading and area patterns.

Metacode is widely used on mainframe platforms, such as z/OS, for high-speed, high-volume production runs. The z/OS-based version of Metacode processes differ in setup and production details from other output formats. The following sections summarize key terms and acronyms that appear on the Exstream graphical user interface for Metacode output.

### Metacode Print Streams

Metacode data and output commands can be in either EBCDIC or ASCII format. To help compress print file size, this language uses hexadecimal values to specify commands and x/y coordinates on a page.

Other features of this output driver include the following:

- Resources are typically stored on the printer.
- No subsystem monitors resource usage.
- Images appear at the top of the file so that they can be placed as needed on pages.
- Font sizes generally do not exceed 60 points.

## Print Stream Types

Some types of Metacode print streams that appear on Exstream dialog boxes and tabs include the following:

- **Metacode**—An all-points addressable dynamic page composition print stream
- **Line Conditioned Data Stream (LCDS)**—An extended line print stream with bin selection capabilities
- **Xerox Escape Sequence (XES)**—A distributed printer escape sequence language
- **Interpress**—A print stream with Page Description Language (PDL), a precursor to PostScript

## Acronyms Commonly Used with Metacode

As you set properties for Metacode output, you will encounter the following acronyms. Most of these refer to resource files required for traditional Metacode production.

### Metacode acronyms

Acronym	Description
DJDE (Dynamic Job Description Entry)	Metacode uses a native ASCII character set for text, along with binary commands and embedded DJDE records. A non-printing DJDE record can be placed within existing line data printer files to add special print features, such as the format of the page and duplex information.
IDEN	The IDEN command is a character sequence that identifies and invokes DJDE records for the print stream.
JCL (Job Control Language)	JCL is batch-oriented command language used to identify jobs and their requirements to the operating system. JCL includes job cards, Data Definitions (DDs) that define files, null statements ( // ) that mark the end of a job when they appear alone on a line, and comment statements that begin with // *. You use this language with batch processing.
JDE (Job Descriptor Entry)	The information in the JDE defines the formatting of a particular job. You can have more than one JDE per JSL file.

#### Metacode acronyms, continued

Acronym	Description
JDL (Job Description Library)	The JDL controls specific formatting elements for the formatting of a job or a series of jobs. There is only one JDL per JSI file.
JSI (Job Source Language)	The JSI is a file containing JDEs and JDLS.

### 2.1.9 MIBF (Inkjet)

The MIBF output driver produces the necessary resources to support the print stream for the proprietary Memory Image Bitmap File (MIBF) format for Miyakoshi Corporation's inkjet printers. The MIBF output driver supports high-speed, full-color production and high-quality plane separation for CMYK JPEG and CMYK TIFF images.

The MIBF output driver lets you use Unicode character sets to reach an expanded audience with your communications. Inkjet printers also provide you with many color manipulation options.

Masking lets you alter the appearance of an object by hiding or showing parts of the object. Exstream determines when masking is needed for MIBF output and implements this feature by default. The following objects can be masked with MIBF output.

#### Masking specifications

Object	Can be masked?
Text	Yes
Top-of-file images	Yes
On-page images	No
Overlays	No

### 2.1.10 PCL

The PCL output driver produces the necessary resources to support the print stream for the proprietary Printer Command Language (PCL) file format for HP PCL4 and PCL5 printers. PCL is a page description language developed by Hewlett-Packard. This ASCII-based standard supports low- to mid-volume output runs. The PCL output driver supports black-and-white, grayscale, highlight color, and full-color modes for low- to mid-volume production.

Exstream supports PCL4 and PCL5e standards in black and white and in color.

Especially well-suited to printing over a Local Area Network (LAN) environment, PCL is known for producing static documents in a business office setting. The language used by PCL is

common to all HP printers, minimizing support problems. The PCL output driver also does the following:

- Uses fonts embedded in the print stream or installed on the printer (sometimes on cartridges), expanding the font options for your output
- Employs escape sequences for printer instructions, enabling you to print special characters

## 2.1.11 PDF

The PDF output driver produces fully-composed Portable Document Format (PDF) files that are automatically bookmarked and are searchable using the Adobe Acrobat Reader. The PDF output driver supports PDF security features, binary compression, and base 14 or TrueType fonts. You can add accessibility tags to these files to control and optimize the way the document is read by a screen reader or text-to-speech converter.

Exstream supports PDF version 1.3 and later for standard PDF output.

When you produce PDF-based output, you can do the following:

- You can send or post your PDF documents on the web for customer access.
- A growing number of high-volume print vendors can use your PDF files to complete production runs.
- Customers can easily view PDF output by either downloading the free Acrobat Reader from the Adobe website or purchasing the full Acrobat program. Customers can view, print, or change PDF documents depending on the Acrobat program they use.
- You can place varying levels of security on documents, including encryption, password protection, and editing restrictions.

## 2.1.12 PDF/A

The PDF/A output driver produces a subset of PDF output that is used specifically for archiving documents.

Exstream supports the production of PDF documents that conform to the following levels of PDF/A compliance:

- **PDF/A-1a**—This level of compliance ensures that the logical structure and content text stream of a document is preserved in the natural reading order. Preserving the logical structure ensures that the file will be displayed correctly on multiple viewing devices. This level also ensures that text can be extracted. Use PDF/A-1a if you must produce output in accordance with Section 508 with the US Rehabilitation Act. Accessibility tags are automatically enabled for PDF/A-1a output. PDF/A-1a produced in Exstream meets PDF 1.4 standards.

- **PDF/A-1b**—This level of compliance meets the minimum requirements for the PDF/A standard. PDF/A-1b output preserves the visual appearance of archived documents only with the original structure of the file. This level of compliance is not optimized for viewing on multiple devices. Use PDF/A-1b if you do not anticipate the need to extract text. PDF/A-1b output does not necessarily meet the requirements of Section 508 of the US Rehabilitation Act. Accessibility tags are optional for PDF/A-1b output. PDF/A-1b produced in Exstream meets PDF 1.4 standards.
- **PDF/A-2a**—This level of compliance is an ISO-standardized version that is suitable for long-term archival material. This update to the PDF/A standard supports PDF 1.7 standards when creating archival documents, so you can include digital signatures in accordance with the PDF Advanced Electronic Signatures (PAdES) standard, ADO stream objects, and JPEG2000 (\*.jp2) images. Accessibility tags are required for PDF/A-2a output. One advantage of PDF/A-2a is a smaller archival file size. To optimize the file size in Exstream, use JPEG2000 files in the design and select **Max** from the **Compression level** box on the PDF/A-2a output object properties.

### 2.1.13 PDF/VT

The PDF/VT output driver produces a subset of [PDF](#) output that is used specifically for high volume print and electronic output. It is a format that is optimized for variable and transactional (VT) printing in a variety of environments, from desktop printing to digital production presses. It represents the fully assembled output stream of a variable data printing (VDP) job, and provides additional control for printing processes, such as paper tray selection and paper bin configuration.

Exstream supports the production of PDF/VT-1 documents, which are PDF/X-4 compliant. PDF/VT-1 is designed for single-file exchange, which means that all of the layout data and variable data are embedded in one self-contained file. PDF/VT files that are produced in Exstream meet PDF 1.6 standards.

**Note:** Exstream PDF/VT output does not support using accessibility tags, encryption, permissions, or digital signatures.

### 2.1.14 PostScript

The PostScript output driver produces the necessary resources to support the print stream for Level 2 and Level 3 PostScript printers. PostScript output automatically builds forms and images and places them at the top of the print stream so they can be pre-ripped and referenced. PostScript output supports Type42 (available in PostScript v.2.013 and later) and Adobe Type1 fonts; full-color, highlight color, grayscale, and black-and-white color modes; and paper bin selection.

PostScript is a machine-independent output driver that supports macros and commands. Its user-friendly codes control formatting options such as page layout and scale outline fonts. PostScript offers font rendering and graphics that remain faithful to the screen.

## 2.1.15 PowerPoint

The PowerPoint output driver produces output specifically formatted for the Microsoft PowerPoint 2002 or 2003 business presentation software. In Exstream, PowerPoint output is produced as modified HTML with PowerPoint commands for design objects such as lines, shapes, and text boxes.

## 2.1.16 PPML

The PPML output driver produces an XML-based structure for Personalized Print Markup Language (PPML) output that encapsulates all the objects needed to send the print stream to the printer, which can hold the data in memory to reference as needed for each customer.

Exstream supports PPML version 2.0.

PPML output enables a printer to act like a browser. This means that the printer can hold data in memory and reference it as required. The PPML language uses Extensible Markup Language (XML) as a structural base and can even encapsulate objects from different formats (such as PDF and JPEG) in an XML-based wrapper.

PPML is not optimized for the z/OS platform.

## 2.1.17 RTF

The RTF output driver produces Rich Text Format (RTF) files that are ready for viewing and editing in word processing programs. These files are used most often for communications that customers can revise in word processing programs such as Microsoft Word. Exstream supports RTF output for Microsoft Word 97 and later.

## 2.1.18 TIFF

The TIFF output driver produces Tagged Image Format File (TIFF) graphic file output. TIFF output is viewable on most graphic viewers and maintains a high degree of graphic quality.

TIFF output converts each page in the design to a black-and-white bitmap file, compressed with TIFF Group 4 compression, the standard format for faxing or for storing documents in archival and retrieval systems.

## 2.1.19 TOP

The TOP output driver produces the necessary resources to support the print stream for TruePress Optimized PostScript (TOP) printers. TOP is an extension to the PostScript language. This primarily print-media language supports raster image processing (RIP) of pages and uses pre-rasterized elements to speed the printing process.

## 2.1.20 VDX

The VDX output driver produces Variable Data Exchange (VDX) output. VDX output builds on the strengths and implementation base of PPML. A VDX output file is a PDF file with an embedded PPML layout file. Exstream embeds the PPML in the PDF file, which is called thick or strict VDX. As a result, VDX has many of the same options supported for PDF, but all printer product intent commands are handled using PPML data.

To view VDX files on-screen, use the NexPress VDX Viewer plug-in to Adobe Acrobat. The plug-in opens .vdx documents and displays all the records associated with a document as bookmarks. To view the PDF objects in the plug-in, select **Show Original PDF data** from the **VDXViewer** submenu.

## 2.1.21 VIPP

The VIPP output driver produces the necessary resources to support the print stream for Xerox printers that use Variable Data Intelligent PostScript PrintWare (VIPP), an extension of PostScript developed for variable-data applications. It is optimized for speed and overlay caching for high-volume variable full color output. With VIPP output, Exstream can use objects that have been through raster image processing and stored in the cache memory to produce high-volume color output at high speeds, particularly with the Xerox iGen3 color printer.

For z/OS platforms, Exstream supports only "thick" VIPP output.

For more information about creating "thick" VIPP output, see "[Optimizing the size and structure of the print stream](#)" on page 186.

## 2.1.22 VPS

Variable Print Specification (VPS) output produces the necessary resources to support the print stream for Xerox printers that use VPS, an extension of PostScript developed by Creo that provides a structured definition of the documents being processed. VPS supports raster image processing of pages at the rate required for digital color printing, using pre-rasterized elements to speed the production of personalized business documents.

VPS is an extension to the PostScript language. This primarily print-media language supports raster image processing (ripping) of pages, using pre-rasterized elements to speed the production of personalized business documents.

For z/OS platforms, Exstream supports only "thick" VPS output.

For more information about creating "thick" VPS output, see "[Optimizing the size and structure of the print stream](#)" on page 186.

## 2.1.23 XML

Exstream supports both XML input and output. Only XML output is discussed in this guide. XML input requires the XML/JSON Input module.

For more information about the XML/JSON Input module, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

Exstream Design and Production offers the following four modules for XML output:

- **XML (content)**
- **XML (composed)**
- **XML/JSON (data)**
- **XML (multi-channel)**

When you license either the XML (content) module or the XML (multi-channel) module, the Content.xsd schema is included as part of your installation. The schema provides information that Exstream uses to create the framework of the file and to validate your output. Review this schema to help you understand the structure of the XML created by the XML (content) and XML (multi-channel) output drivers. For Multi-Channel XML, understanding the schema is essential for writing the XSL file you use to produce transformed output from Multi-Channel XML output.

This schema can be viewed in word processing programs such as Notepad, in Internet browsers such as Internet Explorer, or in XML editors such as oXygen.

### XML (Composed)

The XML (composed) output driver produces fully-composed XML files based on the Document Type Definition (DTD) for the Exstream Exchange Format (DXF). DXF is based on the W3C standard Extensible Stylesheet Language-Formatting Objects (XSL-FO). This file format simplifies importing and exporting pages between Exstream and business programs manufactured by other vendors. The XML files you create with the XML (composed) output driver include all the page layout, formatting, and all objects sent to the print stream (for example, images and pages). These files are ready for web presentation.

Unlike XML (content) output, which is variable-based only, XML (composed) output contains all the objects sent to the print stream (including shapes, images, pages, and so on). This makes the output file very large.

You can use XML (composed) output to do the following:

- Interface with pre-existing XML-based web-presentation systems.
- Run through routines written in Extensible Stylesheet Language: Transformation (XSLT) to produce HTML or other electronic output.
- Update XML-literate databases directly.
- Produce output that can be imported into QuarkXpress (through a QuarkXTension plug-in).

For more information about the Quark converter, see *Importing Designs* in the Exstream Design and Production documentation.

## XML (content)

The XML (content) output driver lets you create XML output that documents the logical content of your design. Logical content describes the structure and textual content of a document, but not the layout or formatting (for example, color and font). You can use the XML (content) output driver to interface with pre-existing systems that read and mine information from an XML source. If you want to see the layout and formatting of your design in XML, you must use the XML (composed) output driver.

XML (content) is most often used for archival and internal statement regeneration programs.

## XML (data)

The XML/JSON (data) output driver lets you create XML or JSON report files. These files contain only the variable mapping that exists at the application or queue level, not the page or document content.

For information on XML/JSON (data) output, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

## XML (multi-channel)

The XML (multi-channel) output driver lets you create Multi-Channel XML output, which is XML that can be transformed for delivery to different media channels (for example, Short Message Service (SMS) messages, social media posts, and VoiceXML-based speech applications).

To transform the Multi-Channel XML output from Exstream into output that is ready for these media channels, you provide an Extensible Stylesheet Language (XSL) file. After Exstream creates Multi-Channel XML output, the output file is sent along with the XSL file to the embedded XSL Transformation (XSLT) engine during the production run. The transformed output produced from the XSLT engine can be used for many different media formats, as determined by the XSL file you provide.

The embedded XSLT engine in Exstream supports XSLT version 1.0.

## 2.1.24 ZPL

The ZPL output driver produces the necessary resources to support the print stream for Zebra printers that use ZPL II commands. Exstream produces a black-and-white image using the Zebra compression algorithm, and includes the necessary ZPL II commands to produce the image from a compatible printer.

When you prepare to produce ZPL output, keep in mind that you cannot use multiple-up (MUP) objects with ZPL output.

## 2.2 Setting Up General Output Driver Configurations

Output objects are Library objects to which you assign your output driver and with which you customize the production settings for output. These settings are used to control the print stream that is sent to your output device or viewing software. You can use an output object to create a print stream with any of the output drivers you have licensed, whether or not an output device is present.

In Exstream, some basic output settings are the same or similar for many output drivers.

This section discusses the following topics:

- [“Creating an Output Object” on the next page](#)
- [“Setting the value of the queue selection variable” on page 35](#)
- [“Managing output resolution in Exstream” on page 37](#)
- [“Using Simplex and Duplex in Output” on page 39](#)
- [“Specifying the Way Color is Produced in Output” on page 40](#)
- [“Creating Pages as Images” on page 44](#)
- [“Assigning a Connector to an Output Object for Testing and Production” on page 45](#)
- [“Using Search Keys in Output” on page 46](#)
- [“Specifying Unprintable Areas for Page Margins” on page 48](#)
- [“Highlighting New Text and Images in Output” on page 49](#)

Not all options are available for all output drivers.

For more information about available options for setting up each output driver, see the individual output sections.

## 2.2.1 Creating an Output Object

1. To select an application mode, complete one of the following tasks:

To	Do this
Create a new output object in SBCS mode	In Design Manager, in the Library, right-click the <b>Outputs</b> heading and select <b>New Output</b> . The <b>New Output</b> dialog box opens.
Create a new output object in DBCS mode	In Design Manager, in the Library, right-click the <b>Outputs</b> heading and select <b>New Output</b> . The <b>New Output</b> dialog box opens.
Create a new output object in SBCS/DBCS mode	a. In Design Manager, in the Library, expand the <b>Outputs</b> heading. b. In the Library, right-click the type of output object you wish to create. For example, select <b>SBCS outputs</b> for a new SBCS output object. The <b>New Output</b> dialog box opens.

2. In the **Name** box, enter a name for the output device.

3. In the **Description** box, enter a description (optional).

4. Click **Finish**.

The output object opens in the Property Panel for you to define.

5. In the **Model** box, you can enter descriptive information that can help you differentiate output objects. For example, you might use characteristics such as printer family, model, capabilities, or location.

6. From the **Driver** drop-down list, select the output driver you want to use.

7. For some output drivers, you must select a subtype:

Output driver	Options available
PCL	<ul style="list-style-type: none"><li>• Level 4</li><li>• Level 5e</li></ul>
PDF/A	<ul style="list-style-type: none"><li>• 1a</li><li>• 1b</li><li>• 2a</li></ul> <p>For more information on PDF/A, see "<a href="#">Setting Up PDF/A Output</a>" on page 144.</p>
PostScript	<ul style="list-style-type: none"><li>• Level 2</li><li>• Level 3</li></ul> <p>Exstream does not support the /MediaClass parameter available with Level 3 PostScript printers.</p>

8. Save the output object.

**Note:** If you choose to create a new output object by selecting **Insert> Environment > Delivery > Output Device**, you must specify the application mode from the **Application mode** drop-down list in the **New Output** dialog box.

## 2.2.2 Setting the value of the queue selection variable

You can use the SYS\_CustomerQueues system variable to specify the list of output queue names that is enabled for a customer. During the engine run, when customer data is processed, the output queues that are specified by the SYS\_CustomerQueues variable will be flagged for inclusion. It is important to note, however, that when the engine processes output queue rules, a queue that has been included by the variable could be excluded by a rule, and would therefore not be used. Additionally, queue names that are included by a rule but that have not been specified by the SYS\_CustomerQueues variable will be disabled.

## Using the variable with Exstream Document Generation on Demand

In the Exstream Document Generation on Demand environment, the SYS\_CustomerQueues variable makes it easy to dynamically specify which output queues you want to use when you generate a document. The value of the variable is supplied in the Exstream Document Generation environment either by a query parameter in the document generation HTTP request,

or by the output types that are specified in the HTTP Accept header of a document generation request.

**Important:** For documents that will be produced using Exstream Document Generation, do not map the SYS\_CustomerQueues variable.

- To specify the output types in the document generation HTTP request, use the following query:

```
?format=<output_type>
```

**Example:**

```
?format=PDF
```

- To specify the output types in the HTTP Accept header of the request, use one of the following MIME types:

Supported MIME types and Exstream output drivers

Output type	MIME type	Exstream output driver
PDF	application/pdf	PDF
DOCX	application/vnd.openxmlformats-officedocument.wordprocessingml.document	Word (*.docx)

For the latest information about which output queue types can be used with Exstream Document Generation on Demand, see the online help that is available from within the Exstream Document Generation on Demand interface.

## Using the variable with the classic Exstream engine

Although the SYS\_CustomerQueues system variable was designed primarily for applications that will be used with Exstream Document Generation, this variable can also be added to the control file for any Design and Production application. To specify the list of output queue names that is enabled for a customer, use the SYS\_CustomerQueues system variable in the VARSET switch in the control file for your application.

When you add the VARSET switch to a control file, use the following syntax for this variable:

```
-VARSET=SYS_CustomerQueues,<output_queue_name>,PREINIT
```

You can include as many queue names as needed, by specifying multiple VARSET switches in your control file. For example:

```
-VARSET=SYS_CustomerQueues,PDF_Queue_Main,PREINIT
```

```
-VARSET=SYS_CustomerQueues,Acceptance_Letter_PDF_Queue,PREINIT
```

```
-VARSET=SYS_CustomerQueues,DOCX_Queue,PREINIT
```

The SYS\_CustomerQueues system variable can be used to specify all output queue types for applications that are fulfilled using the classic Exstream engine.

## 2.2.3 Managing output resolution in Exstream

You can customize the output resolution settings for output to ensure that output is clear and readable. The output resolution specifies the sharpness of text and graphics, measured in dots per inch (dpi). If you are generating printed output, use a dpi at or lower than the maximum resolution supported by the output device you are using for the best processing performance.

You can set the output resolution for the following output drivers:

- AFP
- IJPDS
- Metacode
- MIFB
- PCL
- PDF
- PDF/A
- PDF/VT
- PostScript
- PPML
- TOP
- TIFF
- VDX
- VIPP
- VPS
- XML (composed)
- XML (multi-channel)
- ZPL

When you set the output resolution, keep in mind the following information:

- **Design resolution settings**—For the best results, set the design resolution to be the same resolution as the output resolution. By using the same resolution as the output device, you can help prevent the output from appearing differently than the design (for example, preventing bold text from being overprinted).

For more information about setting the design resolution, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Image resolution settings**—When you import images into Designer, for the best results, match the resolution to the output resolution. If the image resolution settings are different from the output resolution settings, you will get larger files and can experience other problems (for example, unpredictable sizing and a reduction in image quality).

For more information about image resolution, see “[Specifying How Image Resolution Settings Are Processed](#)” on page 238.

- **Multiple output resolution settings**—Significant differences between the output resolutions for multiple output drivers during the same production run can affect pagination. For example, suppose an application produces output to both PostScript and PDF output. If you set the PostScript output resolution to 240 dpi and PDF output resolution to 600 dpi, the PDF output might push a few rows of a table to an extra flow page, resulting in an increased page count compared to the PostScript document. To make sure the pagination is the same for both outputs, set the same output resolution on both output objects.
- **Resolution settings and file size for electronic output**—Resolution settings directly affect the size of an electronic file. If you produce PDF output with a 600 dpi resolution setting, the file will be much larger than a file produced with a 300 dpi resolution setting.

## Specifying the output resolution

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Resolution** list, select the dpi setting you want to use from the available options.  
The options in the **Resolution** list vary depending on the type of output you select.

## Specifying a custom output resolution

If the output resolutions available in the **Resolution** list do not meet your business needs, you can specify a custom output resolution for some output drivers.

You can specify a custom resolution for the following output drivers:

- PDF
- PostScript
- PPML
- TIFF
- TOP
- VDX

- VIPP
- VPS
- XML (composed)
- XML (multi-channel)
- ZPL

To specify a custom output resolution:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. In the **Resolution** list, select **Custom**.
3. In the **Custom** box, enter the custom output resolution you want to use. The minimum valid resolution you can set is 72, and the maximum valid resolution is 2400.

## 2.2.4 Using Simplex and Duplex in Output

You can set up output as simplex, duplex, or both simplex and duplex (also known as mixed-duplex output). When you choose to use both simplex and duplex pages in the print stream, the engine uses page and document properties to specify which pages to print as duplex. Pages that you design as duplex automatically change the output device into duplex mode; simplex pages change the output device into simplex mode. In addition, system variables that track actual page counts also take into account which pages are simplex and which pages are duplex.

You can set simplex and duplex options for the following output drivers:

- AFP
- IJPDS
- Metacode
- MIBF
- PCL
- PDF
- PDF/A
- PDF/VT
- PostScript
- PPML
- TOP
- VDX

- VIPP
- VPS

## Setting Up Simplex and Duplex Options

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Simplex/duplex** list, select one of the following:
  - **Simplex**—Prints only on one side of a page
  - **Duplex**—Prints on both sides of a page
  - **Simplex and Duplex**—Uses both simplex and duplex options (sometimes called multi-plex output). The engine composes the print stream as if it were simplex, so no blank pages are inserted. The **Simplex and Duplex** option operates at an individual page level. To control duplexing at the document level, select an option in the **Duplex Method when Printing on a Mixed-plex Output** list in the document properties.
4. Save the output object.

For more information about document and page properties, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 2.2.5 Specifying the Way Color is Produced in Output

Depending on the output driver you select, you can produce output in full color, with only select colors, in black and white, or in shades of gray. Some output drivers have additional options for producing color, including AFP, Metacode, and PCL.

You can specify how color is produced in output for the following output drivers:

- AFP
- IJPDS
- Metacode
- MIFB
- PCL
- PDF
- PDF/A

- PDF/VT
- PostScript
- PPML
- TOP
- VDX
- VIPP
- VPS

The following table demonstrates the availability of options in the **Color mode** list, based on the output driver you are using:

Content type availability

Output drivers	Full color	Black and white	Highlight	Gray scale	Gray scale, black text	AFP OCA (color output)	AFP OCA (black and white Output)	Use H and K values for non-raster fills
AFP	X	X	X			X	X	
IJPDS	X		X	X				
Metacode		X	X					X
MIFB	X		X	X				
PCL	X	X	X	X	X			
PDF	X	X	X	X				
PDF/A	X	X		X				
PDF/VT	X	X	X	X				
PostScript	X	X	X	X				
PPML	X	X	X	X				
TOP	X	X						
VDX	X	X	X	X				
VIPP	X	X	X	X				
VPS	X	X	X	X				

For more information about setting up color, see “[Setting Up Color Options for Output](#)” on page 59.

For more information about using color in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To specify how color is produced in output:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Basic** tab.
3. Depending on how you want to use color in output, complete one of the tasks in the following table:

To	Do this
Produce output in full color	In the <b>Color mode</b> list, select <b>Full color</b> .  For printed output, if you are using a printer that does not print in full color, the different colors are simulated by patterns. These patterns are determined by the output device.
Produce output in full color using OCA technology (AFP output only)	In the <b>Color mode</b> list, select <b>AFP OCA (color Output)</b> .
Produce output with highlight colors	<ol style="list-style-type: none"><li>a. In the <b>Color mode</b> list, select <b>Highlight</b>.</li><li>b. To specify a color other than black or white for all images, click the adjacent color well.</li><li>c. Select the color you want to use.</li><li>d. Click <b>OK</b>.</li></ol>
Produce output with multiple highlight colors (AFP output only)	<ol style="list-style-type: none"><li>a. In the <b>Color mode</b> list, select <b>Highlight</b>.</li><li>b. Select the <b>Use multiple highlight colors</b> check box.</li><li>c. To specify a color other than black or white for all images, click the adjacent color wells.</li><li>d. Select the color you want to use.</li><li>e. Click <b>OK</b>.</li><li>f. If your AFP printer supports CMOCA architecture, you can select the <b>Use CMOCA to mix and shade highlight colors</b> check box to mix and shade highlight colors to create new highlight colors or different intensities of one highlight color for text and graphics.</li></ol> <p>The <b>Use CMOCA to mix and shade highlight colors</b> is inactive if you select the <b>Convert charts and shapes into images</b> check box on the <b>Resource Management</b> tab.</p>

To	Do this
Produce output in black and white or shades of gray	<p>a. In the <b>Color mode</b> list, select either <b>Black and white</b> or <b>Gray scale</b>.</p> <p>b. To enhance image quality, change the halftone settings. A halftone image appears to have depth based on the size, shape, and frequency of dots. Click the <b>Advanced</b> tab and select one of the following options in the <b>Halftone</b> list:</p> <ul style="list-style-type: none"> <li>• <b>No halftone images</b>—Images are not halftoned.</li> <li>• <b>Controlled by image</b>—Images are halftoned using the settings on the <b>Image Color Management</b> dialog box.</li> <li>• <b>All images must be halftoned</b>—All images have the same halftone properties. When you select this option, the settings in output properties take precedence over the properties in the <b>Image Color Management</b> dialog box. If you select this option, complete step c through step e.</li> </ul> <p>c. Click  to adjust the halftone settings. The <b>Halftone Setting</b> dialog box opens.</p> <p>d. In the <b>Shape</b> list, to select the shape of the dots used to create the halftone, select an option.</p> <p>e. If you select <b>Default</b>, <b>Elliptical</b>, or <b>Linear</b> in the <b>Shape</b> list, you can specify a number to control the angle of the axis for the dots in images in the <b>Angle</b> box. You can enter a number between 0 and 180. Make sure the number you use is compatible with the setup of your output device.</p> <p>f. If you select any option in the <b>Shape</b> list except <b>Default</b>, specify a grain size. In the <b>Grain</b> box, enter a number between 1 and 15 to specify a grain size for the dots in images.</p> <p>g. Click <b>OK</b>.</p>
Produce output in black and white using Object Content Architecture (OCA) technology (AFP output only)	In the <b>Color mode</b> list, select the <b>AFP OCA (black and white Output)</b> option.
Produce output in shades of gray with black text (PCL output only)	In the <b>Color mode</b> list, select <b>Gray scale, black text</b> .
Instruct the Exstream engine to convert the RGB value to an H and K value, look up the value on the list, and then use the index to specify the exact hue color (Metacode output only)	<p>Select the <b>Use H and K values for non-raster fills</b> check box.</p> <p>When you select this option, the Exstream engine places the following two items at the top of the print stream:</p> <ul style="list-style-type: none"> <li>• <b>Palette information</b>—Created based on the color you select for the <b>Highlight color</b></li> <li>• <b>Color resource list</b>—Contains all the available H and K color combinations for the palette</li> </ul>

4. Save the output object.

**Note:** PANTONE Colors are not supported in AFP, HTML, HTML (email), IJPDS, Metacode, MIFB, PCL, or RTF output. If you use a PANTONE Color in a design for output that does not support PANTONE Colors, Exstream will substitute an equivalent color value.

## 2.2.6 Creating Pages as Images

In some cases, you might want to create each page as an image. Design Manager uses a TIFF engine to send each page into the print stream as an image. Exstream imports and converts the PDF file (sized 8.5 x 11 inches) into a TIFF page image (sized 5.5 x 8.5 inches, based on the design page). Because image scaling is not supported for TIFF output, the page image cannot be scaled.

If you choose to create each page as an image, you can produce output only in black and white.

Creating each page as an image increases processing time and disk requirements, but use this option for the best results in the following situations:

- If commands or settings in a print stream do not operate with your output or archival system
- If you receive unexpected results when you select **Portrait Reversed** or **Landscape Reversed** on the page properties

If you are creating PDF output, you should use this option only if you do not want to use accessibility tags or to produce searchable text.

You can create pages as images for the following output drivers:

- AFP
- Metacode
- PCL
- PDF
- PDF/VT
- PostScript

To creating each page as an image:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Image each page** check box.
4. Save the output object.

The **Color mode** list automatically changes to **Black and white**.

For more information about the color mode, see “[Specifying the Way Color is Produced in Output](#)” on page 40.

## 2.2.7 Assigning a Connector to an Output Object for Testing and Production

You assign a connector to an output object so a DDA output file can perform post processing to manipulate output before it reaches the print stream. You are not required to assign a connector to an output unless you want your routine to buffer, encrypt, and manipulate in other ways engine-produced output in manageable pieces and, if needed, return manipulated data back to the engine.

This option is available for all output drivers.

To assign a connector to an output object:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Advanced** tab.
3. In the **Test post processor** box, click  .  
The **Select Connector** dialog box opens.
4. Highlight the connector object you want to assign to all test runs that use this output object.  
Click  to browse in the **Folders** dialog box to open a specific folder in the Library that contains the connector you want to assign.
5. Click **OK**.
6. Next to the **Production post processor** box, click  .  
The **Select Connector** dialog box opens.
7. Select the connector object you want to assign to this output object. You can use the same connector object (that is, the same routine) for both test and production, or if you use different routines for the two types of engine runs, you can use a different connector objects..
8. Click **OK**.
9. Save the output object.

For more information about DLL routines and using connectors, see *Configuring Connectors* in the Exstream Design and Production documentation.

## 2.2.8 Using Search Keys in Output

Search keys are versatile features in Exstream that let you add data to the print stream to both electronic and print output. Data you can add includes comments, searchable tags, instructions to print devices, and more.

Since you can customize the output driver, value, timing, and other properties, search keys can be used to define unique methods of counting or as a way of inserting output device or post-processing commands (for example, specifying the stapling options).

For more information on search keys, see [“Adding Non-Printing Data or Production Device Controls with Search Keys” on page 326](#).

The following table explains search key options for various output drivers:

Limitations for search keys on output drivers

Output driver type	Search key limitations
3211 Line Data	Search keys not supported
AFP	DBCS characters not supported
DOCX	Search keys not supported
EDGAR HTML	DBCS characters not supported
HTML	DBCS characters not supported
HTML (email)	Search keys not supported
IJPDS	Search keys not supported
Metacode	DBCS characters not supported
MIFB	Search keys not supported
PCL	DBCS characters not supported
PDF	No limitations
PDF/A	No limitations
PDF/VT	No limitations
PostScript	DBCS characters not supported
PowerPoint	DBCS characters not supported
PPML	DBCS characters not supported

## Limitations for search keys on output drivers, continued

Output driver type	Search key limitations
RTF	No limitations
TIFF	DBCS characters not supported
TOP	DBCS characters not supported
VDX	No limitations
VIPP	DBCS characters not supported
VPS	DBCS characters not supported
XML (content)	To use DBCS characters in search keys, you must specify a Unicode encoding.
XML (composed)	To use DBCS characters in search keys, you must specify a Unicode encoding.
XML (multi-channel)	DBCS characters not supported
ZPL	Search keys not supported

As you use search keys, keep in mind that DBCS characters appear differently in the print streams for the following outputs:

- **HTML:** Double-byte characters are represented by their decimal values and are preceded by &#.
- **PDF:** Double-byte characters are represented by their hexadecimal values (big-endian encoding) and are preceded by a byte-order mark.
- **RTF:** Double-byte characters are represented by their decimal values and are preceded by \u.
- **VDX:** Double-byte characters are represented by their hexadecimal values (big-endian encoding) and are preceded by a byte-order mark.

## Specifying the Type of Search Keys to Use in the Output

To specify the type of search keys to use in the output:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Advanced** tab.
3. In the **Search key** list, select one of the following options:
  - **As is**—Adds a record with no comment or NOP characters. This method can be used to add search data such as special indexing or Table of Contents tags. Use this method

to enter custom printer commands and secondary device commands (for example, for staplers).

- **NOP comments**—Prefixes the search key with the appropriate No Operation (NOP) comment value for the driver (such as %). This method is useful for adding information that can be used for indexing, archiving, future machine instructions, or a very large instruction word (VLIW).
  - **TLEs**—(AFP output only) Adds Tagged Logical Elements (TLEs). TLEs serve as begin and end records in AFP output. These can be array variables so that each element is included as a separate entry in the print stream.
  - **NOP comments and TLEs**—(AFP output only) Allows the engine to write search keys to both TLEs and NOP comments in AFP output.
  - **PDF/VT metadata**—(PDF/VT output only) Adds document part metadata (DPM) information to PDF/VT output. This method is useful for adding information that can be used for production (such as sorting and bundling) or information about the recipient (such as the corresponding ZIP code).
  - **All**—Search keys can be a mix of all the available types.
4. Save the output object.

## 2.2.9 Specifying Unprintable Areas for Page Margins

For printed output, you can set the print stream so that you do not print on page edges. You might want to set these unprintable areas because of your output device limitations, because you are using a pre-printed or non-standard stock, or because of post-processing trimming.

Keep in mind that all desktop printers have an unprintable area, and the output is offset by the printer automatically. A centralized production printer does not have the same restrictions. If you want to print from edge to edge when you use a desktop printer, consult your desktop printer manufacturer documentation for information.

You can create specify unprintable areas for page margins for the following output drivers:

- AFP
- Metacode
- PCL
- PostScript
- PPML
- TOP
- VIPP

To specify unprintable areas for page margins:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Unprintable area offsets** area, enter the values of the unprintable areas on the edges of your pages. If the page is duplex, you can define offsets for both the front and the back of the page.
4. For each side of a printed sheet, enter the measurements of the **x (short side)** and **y (long side)** in the **Front** and **Back** boxes.
5. Save the output object.

## 2.2.10 Highlighting New Text and Images in Output

If revision tracking is turned on, you can select the **Show Blacklines** check box to highlight added text or images in your output with a black underline. The **Show Blacklines** feature shows only added text, not deleted text. For example, if you delete four paragraphs of text and add one, the only revision tracking you see is a black underline highlighting the one added paragraph.

For more information on revision tracking, see *Designing Customer Communications* in the Exstream Design and Production documentation.

You can highlight new text and images for the following output drivers:

- AFP
- HTML
- IJPDS
- Metacode
- MIFB
- PDF
- PDF/VT
- PostScript
- PowerPoint
- PPML
- RTF
- TIFF
- TOP
- VDX

- VIPP
- VPS
- XML (composed)

To highlight new text and images in output:

1. Make sure revision tracking is turned on.
2. In Design Manager, from the Library, drag the output object to the Property Panel.
3. Click the **Advanced** tab.
4. Select the **Show Blacklines** check box.
5. Save the output object.

**Note:** For images, the **Show Blacklines** option works only with embedded images in text boxes.

## 2.3 Setting Up Paper Bins

For print output, you can store and use multiple paper bin configurations. Each configuration is a separate bin contents object in the Library.

You can set up paper bins for the following output drivers:

- AFP
- Metacode
- PCL
- PDF/VT
- PostScript
- PPML
- TOP
- VIPP
- VPS

If you set up paper bins on an output object, use the SETPRINTERBINCONTENTS engine switch to specify which of the bin contents configurations you want. If you have multiple bin configurations for the output device and you do not use the switch, then the engine selects the first bin contents object listed in the bin contents selection area of the **Paper Bins** tab.

For more information about engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

This section discusses the following topics:

- “[Setting Up a Paper Type](#)” below
- “[Specifying the Paper Types in Each Bin](#)” on page 53
- “[Setting Up Paper Bins for Output](#)” on page 54
- “[Controlling Paper Selection For a PostScript Printer](#)” on page 55

### 2.3.1 Setting Up a Paper Type

Before you can set up paper bins, you must set up paper type objects to define the size, weight, and color of paper stock, as well as media names for printer control. Setting up paper type characteristics promotes efficient printing and allows you tight control over the appearance of your output. Paper types do not always define physical paper. Paper types can also define the screen properties for electronic output. Before creating a page template, you must have created a paper type.

For more information about setting up a paper type, see *System Administration* in the Exstream Design and Production documentation.

For some output drivers, you must complete the following tasks:

- “[Specifying Media Names](#)” below
- “[Specifying an AFP Copygroup Name](#)” on the next page

Review the documentation for your output device to determine whether you must use these options.

### Specifying Media Names

Media names define special settings for your printing system and the paper stock you use for a particular print run. After you create a paper type, you can set options in the Property Panel that let you control paper bin switches, printing offsets, and paper orientation.

To specify media names for the following output drivers:

1. In Design Manager, from the Library, drag a paper type object to the Property Panel.
2. Depending on your output driver, set the following options in the **Media name(s)** area:

For this output driver	Do this
Metacode	<p>In the <b>Metacode stock name</b> box, specify the stock name or stock reference to use on the FEED command in the DJDE record.</p> <p>If you do not specify a Metacode stock name, the output bin positions create FEED= commands. The paper type in the first bin creates the FEED=MAIN command, the second bin becomes FEED=AUX, the third bin becomes FEED=3, and additional bins become FEED=n.</p>
PCL	<p>In the <b>PCL tray command</b> box, specify the paper tray to use.</p>
PDF/VT	<p>In the <b>PDF/VT media name</b> box, specify the name (or source) of the media to use for PDF/VT output.</p> <p><b>Note:</b> If you specify the PDF/VT media name, and you are also using <a href="#">document part metadata (DPM) search keys</a> for PDF/VT output, OpenText recommends that you select the <b>Insert media name as is</b> check box of the PDF/VT output object.</p>
PostScript	<p>In the <b>PostScript media name</b> box, specify the name of the media type to use with the <code>setpagedevice</code> command.</p>
VDX	<p>In the <b>VDX Media Type</b> list, specify the Job Definition Format (JDF) to supply media intent for the print job. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Body</b></li> <li>• <b>Cover</b></li> <li>• <b>Insert</b></li> </ul> <p>For the bin selection to work properly, you must map the media intents from the JDF to a media substrate selection at the press.</p>

3. Save the paper type object.

For the output device to recognize the commands in the **Media name(s)** area, you must select the **Use control bin selection** check box on the **Paper Bins** tab of the output properties.

## Specifying an AFP Copygroup Name

In AFP, copygroup commands instruct an AFP printer which paper bin to use when generating AFP output. Copygroup commands are typically necessary when a production run contains multiple paper types that can include different page orientations, duplex operations, or multiple-up printing. For each paper type you use with AFP output, you can specify a copygroup name to identify the paper type in the AFP print stream. The copygroup name you specify is included as a part of the Invoke Media Map (IMM) command to request the paper type in the AFP formdef.

Sometimes an AFP printer must use the same paper type regardless of the different pages used in a design. For example, the printer might contain only one paper tray or the printer might require a consistent feed of paper. In these cases, you can also specify that Exstream automatically add an additional suffix to the AFP copygroup name to inform the printer of any page rotation necessary for orientation or duplexing.

AFP copygroup names are limited to eight characters. If you want to include suffixes, keep in mind that you must also reserve character space for the suffix characters in the copygroup name.

You must also make sure that the paper bins you specify are available from the printer, otherwise the IMM records are not placed in the AFP stream.

To specify an AFP copygroup name:

1. In the **AFP Copygroup** name box, specify a name to represent the paper type in the IMM command.
2. To append a suffix to the copygroup name to represent the orientation of the page, select the **Add orientation suffix (P, L, I, J)** check box. When this check box is selected, one of the following suffixes is automatically appended to the copygroup name, depending on the orientation of the page: P (portrait), L (landscape), I (portrait reversed), or J (landscape reversed).
3. To append a suffix to the copygroup name to represent whether the page is simplex or duplex, select the **Add simplex/duplex suffix (S, D)** check box. When this check box is selected, one of the following suffixes is automatically appended to the copygroup name, depending on whether the page is simplex or duplex: S (simplex) or D (duplex).
4. From the Menu bar, select **File > Save**.

### 2.3.2 Specifying the Paper Types in Each Bin

In the Library, you define which paper types you have in your output devices by defining bin contents objects. These objects can save time when setting up and running different applications.

Before you can create a bin contents object, you or your system administrator must have created at least one paper type.

For more information about creating a paper type, see *System Administration* in the Exstream Design and Production documentation.

**Note:** If your output device uses roll-fed paper, you do not need to create a bin contents object.

To specify the bin contents for your output:

1. In Design Manager, from the Library, right-click the **Bin Contents** heading and select **New Bin Contents**.

The **New Bin Contents** dialog box opens.

2. In the **Name** box, enter a name for the bin contents object.
3. In the **Description** box, enter a description (optional).
4. Click **Finish**.

The new object opens in the Property Panel for you to define.

5. From the **Paper types** drop-down list, select the number of available bins you want to use. You can define up to eight different bins using one bin contents object.
6. From the paper type selection boxes, select the paper type you want to use for each bin.
7. Save the bin contents object.

### 2.3.3 Setting Up Paper Bins for Output

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Paper Bins** tab.
3. Select the **Use control bin selection** check box to make the bin contents selection area active.
4. In the bin contents selection area, add or remove bin contents objects:

To	Do this
Add a bin contents object	<ol style="list-style-type: none"><li>a. Click .</li><li>The <b>Select Bin Contents</b> dialog box opens.</li><li>b. Select a bin contents object.</li><li>c. Click OK.</li></ol>
Remove a bin contents object	<ol style="list-style-type: none"><li>a. Select a bin contents object from the list.</li><li>b. Click .</li></ol>

5. To change the order of the bin contents, do one of the following:

To	Do this
Move a bin contents object up in the list	Select a bin contents object from the list and click  .
Move a bin contents object down in the list	Select a bin contents object from the list and click  .

6. Save the output object.

### 2.3.4 Controlling Paper Selection For a PostScript Printer

If you are using PostScript print devices, you can control paper selection using Exstream tools to coordinate communication between the Exstream output and the PostScript printer.

When you generate output for a PostScript printer, the print stream includes a description of the paper that is used by the application. When the print stream is delivered to the printer, the printer uses the paper information to search through the paper trays and select the paper tray that contains the closest matching paper for the output.

To accurately select the closest matching paper stored within the printer, different PostScript printers require different information. For example, a printer with fewer paper bins might only need to know the differences in height and weight, while a printer with more paper bins might require additional information (such as height, width, weight, paper color, and so on) to distinguish the difference between the paper stored in each paper bin. To ensure that the PostScript printer makes an accurate paper selection, you can control what paper information is provided in the print stream.

To control paper selection for a PostScript printer, use one of the following methods to determine how paper information is added to the print stream:

To	Do this
Input information from the printer into the print stream to help the printer identify a specific paper tray	Use a PPD file to control paper selection. PPD files can be used with only PostScript printers.
Send information from the Exstream paper type into the print stream to help the printer search for the closest matching paper tray	Use a Media Map to control paper selection. Media Maps can be used with only PostScript Level 2 and PPML printers.

This section discusses the following topics:

- “[Using a PPD File to Control Paper Selection](#)” below
- “[Applying Media Maps to Control Paper Selection](#)” on the next page

## Using a PPD File to Control Paper Selection

All PostScript printers have a PostScript Printer Definition (PPD) file that contains the information needed to accurately identify the printer's paper bins and the types of paper stored in each bin. From Exstream, you can use a PPD file to connect an Exstream paper bin object with the exact values needed to associate output with a specific paper bin in the printer. For example, if a PostScript printer contains multiple bins with various types of paper, you can use the PPD file to embed the exact paper bin values from the printer with the paper bin objects set up in Exstream to ensure that, during production, the printer can accurately associate the correct paper bin with the content being printed.

Using a PPD file can also be useful if you are delivering content to a print vendor. For example, the vendor can provide you with the appropriate PPD file and provide the exact settings necessary for your specific printing needs to ease the transition from design to print.

Keywords represent a specific control within the printer. From Exstream, you can use the following keywords from a PPD file to control paper selection:

Keyword	Typical use
InputSlot	The options for this keyword are typically used to identify a printer tray.
MediaType	The options for this keyword are typically used to identify a paper size and/or color.
MediaWeight	The options for this keyword are typically used to identify the weight of the paper.

Keep in mind the following:

- Before you can apply a PPD file, you must have already set up a paper bin object.  
For more information about setting up a paper bin object, see “[Setting Up Paper Bins](#)” on [page 50](#).
- A PPD file is used only as a reference to allow Exstream to extract and package the appropriate keyword values. When you use a PPD file to define bin settings, the value of the keyword options you select will be included in the PostScript print stream (not the PPD file itself).
- Each PPD file can contain different keywords and options.
- You can review the PostScript value associated with any keyword option by reviewing the **PostScript command** area of the **Media Settings** dialog box as you select keyword options. The PostScript value that appears in the **PostScript command** area is the PostScript value that will be included in the PostScript print stream. If nothing appears in the **PostScript command** area, then there is no additional PostScript associated with the

selected option and the option name will be used as the value to be included in the PostScript print stream.

- You cannot edit a PPD file or its content from within Exstream. If a change is necessary in the PPD file, you must make the change directly in the PPD file.

## Applying a PPD File to Control Paper Selection

1. In Design Manager, in the Library, go to **Environment > Delivery > Bin Contents**.
2. Drag the appropriate bin contents object to the Property Panel.
3. In the **PostScript Printer Definition (PPD)** box, click  and select the PPD file you want to use.
4. For each paper bin, complete the following steps:
  - a. Click **Media Settings**.  
The **Media Settings** dialog box opens.
  - b. Click .
  - c. The **Media Settings Properties** dialog box opens.
  - d. From the **Keyword** drop-down list, select the PPD file keyword you want to define.
  - e. From the **Option** drop-down list, select the keyword option that best applies to the selected paper bin.  
The PostScript syntax for the option appears in the **Postscript command** area. If nothing appears in the **PostScript command** area, then there is no additional PostScript associated with the selected option.
  - f. Repeat step c through step d for each keyword that you want to define.
  - g. Click **OK**.  
The **Media Settings Properties** dialog box closes.
  - h. Click **OK**.  
The **Media Settings** dialog box closes and the PPD media selections are applied to the paper bin.
5. From the Menu bar, select **File > Save**.

## Applying Media Maps to Control Paper Selection

As part of creating an application in Exstream, you set key values (such as media names or paper definitions). You can pass these values into the print stream to help control paper selection. A media map allows you to identify which values from the application should be

mapped from an application into the print stream. This process allows you to select which values are needed by your print device. Media maps can be used with PostScript Level 2 and PPML printers only.

From Exstream, you can use a media map to include the following paper type information in the PostScript print stream:

- PostScript media name
- Width and height
- Color
- Weight

For more information about creating and defining a paper type, see *System Administration* in the Exstream Design and Production documentation.

When you print with multiple paper types, make sure that you specify the applicable PostScript media options. If you do not specify the necessary media options, the printer might not recognize paper change commands and could print in the wrong area of the page.

To apply media maps to control paper selection:

1. In Design Manager, in the Library, navigate to **Environment > Delivery > Output**.
2. Drag the appropriate output object to the Property Panel.
3. Click the **Paper Bins** tab.
4. Select the following media map options as needed to deliver the paper type information in to the print stream:

To	Do this
Include the paper type form name from the <b>PostScript media name</b> box on the paper type	Select the <b>Map media name</b> check box.
Include the paper size form name from the <b>Width</b> and <b>Height</b> boxes on the paper type	Select the <b>Map media size</b> check box.
Include the paper color form name from the <b>Color</b> box on the paper type	Select the <b>Map media color</b> check box.
Include the paper weight form name from the <b>Weight</b> box on the paper type	Select the <b>Map media weight</b> check box.

5. Save the output object.

**Note:** The `/MediaClass` parameter is not available for use with Level 3 PostScript printers.

## 2.4 Setting Up Color Options for Output

In Exstream, you can set up how some output drivers use color by specifying the color format. You can also control the color in your output using color tables and spot colors.

This section discusses the following topics:

- [“Setting the Color Format” below](#)
- [“Setting Up Color Tables to Adjust Named Colors for Output” on the next page](#)
- [“Identifying the Spot Colors Available for an Output Device” on page 61](#)

You can set up the color options in this section for the following output drivers:

- AFP
- PDF
- PDF/A
- PDF/VT
- PostScript
- PPML
- TOP
- VDX
- VIPP
- VPS

### 2.4.1 Setting the Color Format

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Color** tab.
3. From the **Color model** drop-down list, specify the image colors to use in the print stream. The color option you select must be made available for use by the system administrator in the **System Settings**. You can select one of the following options:
  - **RGB**

- **CYMK**

4. Save the output object.

For more information about the color options in the **System Settings**, see *System Administration* in the Exstream Design and Production documentation.

## 2.4.2 Setting Up Color Tables to Adjust Named Colors for Output

If you use custom colors (also known as named colors), you might want to set up color tables so that you can control exactly how colors appear in output. Color tables can offset the difference between the colors you want and the colors produced by printer inks or the viewing software. Before you create a color table, make sure the colors you want to adjust are already configured in one or more color families in the Library.

For more information about color families, see *System Administration* in the Exstream Design and Production documentation.

For more information about color, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### Setting Up a Color Table

1. In Design Manager, in the Library, right-click the **Color Tables** heading and select **New Color Table**.

The **New Color Table** dialog box opens.

2. In the **Name** box, enter a name for the color table.
3. In the **Description** box, enter a description (optional).
4. Click **Finish**.

The new color table opens in the Property Panel for you to define.

5. In the **Adjusted named colors** box, click .

The **Color** dialog box opens.

6. From the **Standard color palette** area, select the color you want to use.
7. Click **OK**.

8. In the **Adjusted value** area, click the color well.

The **Color** dialog box opens.

9. Modify the color values of the named color based on the original values.

10. Click **OK**.

You can see the settings for the named color in the **Named color to adjust** area.

11. Save the color table.

**Tip:** CMYK color models are generally used for printed output, while RGB color models are generally used for output going to electronic devices, such as webpages.

## Adding a Color Table to an Output Object

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Color** tab.
3. In the **Color adjustment table** box, click  . The options available depend on the color tables you have previously defined. To review the properties of the selected color table, click .
4. Save the output object.

### 2.4.3 Identifying the Spot Colors Available for an Output Device

Spot colors are pre-mixed colors that can be applied directly by the printer instead of printing layers of colors. Spot colors offer some colors and specialties (including fluorescent and metallic) that are harder to achieve with process color methods. To ensure that an output device accurately applies spot colors, you must identify which spot colors are available for a specific output device.

If the appropriate spot colors are not available for an output device, Exstream issues an error message and uses normal CMYK color processing to apply the spot color as it is defined in the Design Manager Library.

You must have created spot colors in the Design Manager Library before you can identify the spot colors available for an output device.

For more information about creating spot colors, see *System Administration* in the Exstream Design and Production documentation.

To identify the spot colors available for an output device:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Colors** tab.
3. Select the **Use spot colors** check box to make the options in the **Spot colors** area active

so you can define the spot colors for the output device.

4. In the **Number** box, enter the number of spot colors on the output device. Valid values are between 1 and 8. The value you enter in the **Number** box determines the number of **Spot** boxes that are active.
5. Next to each **Spot** box, click a color well.

The **Color** dialog box opens.

6. From the **Color model** drop-down list, if you have multiple spot colors, select the appropriate spot color.
7. In the **Standard color palette** area, select the base spot color.
8. Click **OK**.

You are returned to the Property Panel. The name of the spot color appears in the box adjacent to the color well.

9. From the Menu bar, select **File > Save**.

For information on the **Color** dialog box, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For information on spot colors, see *System Administration* in the Exstream Design and Production documentation.

## 2.5 Setting Up 3211 Line Data Output

This section discusses the following topics:

- “[Setting Up Carriage Controls](#)” on the next page
- “[Specifying the Maximum Number of Lines and Characters per Inch](#)” on the next page
- “[Specifying Whether to Use the ASCII or EBCDIC Character Set](#)” on page 64
- “[Specifying Record Length](#)” on page 64

In addition to the topics discussed in this section, you can also use the following basic setup options for 3211 line data output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>

## 2.5.1 Setting Up Carriage Controls

In 3211 line data output, the carriage control character acts as a vertical tab command. Carriage controls let you do the following:

- Position the paper at the start of a new page.
- Position the paper at a specified line on the page.
- Control skipping to the next line.

To set up carriage controls:

1. In Design Manager, from the Library, drag a 3211 line data output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Carriage controls** drop-down list, select one of the following options:
  - **ANSI**—The control consists of a single character that is a prefix for the print line.
  - **Machine**—The control consists of literal values, not symbols. They are not represented as characters in any encoding and, therefore, machine controls cannot be translated.
4. To remove all blank lines by using carriage controls to skip lines and break pages, select the **Remove blank lines (using carriage controls)** check box.
5. If you want to use carriage controls to print over text, select the **Allow overprinting lines** check box. For example, to create bold or strike-through formatting, you must select this option. The ability to use this feature is dependent on your printer.
6. Save the output object.

## 2.5.2 Specifying the Maximum Number of Lines and Characters per Inch

1. In Design Manager, from the Library, drag a 3211 line data output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Lines per inch** box, enter the maximum number of lines per vertical inch on each page (taking into account the font used and output settings). For example, if you set 4 lines per inch on an 8 inch page, 32 lines are printed.
4. In the **Characters per inch** box, enter the number of characters per horizontal inch (taking into account the font used). This option is used to determine placement of text on the page.

5. From the **Text overflow** drop-down list, you can specify how to handle text that overflows the maximum length per line. The maximum line length is determined by characters per inch times the width of the page.

Select one of the following options:

- **Truncate right side of text**
- **Shift text to fit**

6. Save the output object.

### 2.5.3 Specifying Whether to Use the ASCII or EBCDIC Character Set

1. In Design Manager, from the Library, drag a 3211 line data output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Character sets** drop-down list, select one of the following:
  - **ASCII**—Uses the American Standard Code for Information Interchange character set. Binary 7-bit code, defined by ASCII, uses 128 combinations of numbers to represent characters. It can be used for text, drawing, and printer controls. Eight-bit bytes (the standard storage unit) provide a total of 256 characters. The extra bit in this standard code is reserved for foreign language and graphic symbols.
  - **EBCDIC**—Uses the Extended Binary-Coded Decimal Interchange Code character set. An IBM standard for representing characters used on IBM compatible mainframes.
4. Save the output object.

### 2.5.4 Specifying Record Length

Depending on your business needs, you can choose to print records with fixed lengths or variable lengths.

To specify the record length:

1. In Design Manager, from the Library, drag a 3211 line data output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Record Separation** drop-down list, select one of the following options:

- **Variable length**—The length of a record is determined by its value.
  - **Fixed length**—Enter a number in the adjacent box to specify the length of a record.
4. Save the output object.

## 2.6 Setting Up AFP Output

This section discusses the following topics:

- “[Setting Up Resource Names for AFP Output in the System Settings](#)” on the next page
- “[Changing the Duplex Page Rotation](#)” on the next page
- “[Using Knockout to Prevent the Mixing of Colors in Overlapping Objects](#)” on page 67
- “[Creating an AFP Index](#)” on page 67
- “[Specifying Where to Place BDT and EDT Fields in AFP Output](#)” on page 68

In addition to the topics discussed in this section, you also use the following basic setup options for AFP output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying how to set up unprintable areas in the page margins	<a href="#">“Specifying Unprintable Areas for Page Margins” on page 48</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Setting up paper bin options	<a href="#">“Setting Up Paper Bins” on page 50</a>
Setting up color formatting options	<a href="#">“Setting Up Color Options for Output” on page 59</a>

## 2.6.1 Setting Up Resource Names for AFP Output in the System Settings

Before you create AFP output, the system administrator must specify certain options in the **System Settings** on the **Production Print** tab:

1. In Design Manager, in the Library, navigate to **Environment > System Settings**.
2. Drag the **System Settings** to the Property Panel.
3. Click the **Production Print** tab.
4. To set up resource names for images and fonts in AFP output, do the following:

To	Do this
Specify a character prefix for images in AFP output	In the <b>Resource naming convention</b> area, in the <b>Image</b> box, enter the character prefix you want to use. You can use either one or two characters.
Specify a character prefix for fonts in AFP output	In the <b>Resource naming convention</b> area, in the <b>Font</b> box, enter the character prefix you want to use. You can use either one or two characters.

5. Save the **System Settings**.

## 2.6.2 Changing the Duplex Page Rotation

When you create duplex AFP output, Exstream creates duplex pages as normal, or flipped along the long edge (or the y-axis). If you use the tumble feature, the printer flips the duplex side on the short (or x-axis) edge of the page.

To flip duplex pages:

1. In Design Manager, from the Library, drag an AFP object to the Property Panel.
2. Click the **Resource Management** tab.
3. Select the **Create and include Formdef in print file** check box.
4. Click the **Basic** tab.
5. From the **Simplex/duplex** drop-down list, select **Duplex**.
6. Select the **Tumble (MMC)** check box.
7. Save the output object.

For more information about creating an in-line formdef, see “[Using Inline Form Definition File in AFP Output](#)” on page 248.

## 2.6.3 Using Knockout to Prevent the Mixing of Colors in Overlapping Objects

If design objects overlap in your design, the color of the objects on top will mix with the color of the objects that they overlap. If you want to prevent the colors of the objects from mixing, you can use the knockout feature.

To prevent overlapping objects from mixing:

1. In Design Manager, from the Library, drag an AFP output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Use knockout for overlapping objects** check box.
4. Save the output object.

The **Use knockout for overlapping objects** check box is inactive if you select either **Medium** or **High** from the **Print stream optimization** drop-down list or if you select **Black and white** from the **Color mode** drop-down list.

For more information about print stream optimization, see “[Optimizing the size and structure of the print stream](#)” on page 186.

For more information about color mode, see “[Specifying the Way Color is Produced in Output](#)” on page 40.

## 2.6.4 Creating an AFP Index

If your organization uses an archival system that requires an index file, you can generate the index file during the engine run.

To create an AFP index:

1. In Design Manager, from the Library, drag an AFP output object to the Property Panel.
2. Select the **Create AFP Index** check box.
3. In the **Local index** box, specify the name and location of the test index file created by the engine.
4. In the **Production index** box, specify the name and location of the index file created by the engine for the production run.
5. Save the output object.

## 2.6.5 Specifying Where to Place BDT and EDT Fields in AFP Output

You can specify where to place Begin Document (BDT) and End Document (EDT) structured fields in the AFP print stream, depending on whether you want to create one large output file or to place a break between each customer record.

To specify where to place BDT and EDT structured fields:

1. In Design Manager, from the Library, drag an AFP output object to the Property Panel.
2. Click the **Advanced** tab.
3. To specify where to place the document breaks in the output file, select one of the following options from the **BDT/EDT placement** drop-down list:
  - **Around entire file**—Create one long "document" containing the entire output.
  - **Around each customer**—Place a "break" between each customer record, so each customer record is considered a separate document by the printer.
4. Save the output object.

## 2.7 Setting Up DOCX Output

This section discusses the following topics:

- [“About DOCX Output” on the next page](#)
- [“Application Planning Considerations for DOCX Output” on page 70](#)
- [“Designing Applications to Maximize the Editability of DOCX Output” on page 73](#)
- [“Exstream Features Supported in DOCX Output” on page 77](#)

In addition to the topics discussed in this section, you can also use the following basic setup options for DOCX output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>

## 2.7.1 About DOCX Output

DOCX output is output that can be opened or edited using Microsoft Word 2007 or later. DOCX output from Exstream is best suited for simple, text-heavy communications that recipients can edit or revise (such as contracts or correspondence). For example, suppose you are creating a contract that can later be distributed by individual insurers. From Exstream, you can design the layout of the content, including objects such as sections and paragraphs to organize agreements or clauses, or tables of contents (TOCs) to help quickly locate information. When the contract is generated as DOCX output and distributed to service representatives, they can edit the contract to include information based on individual customer negotiations or to include information about additional services that were not included in the original contract.

DOCX output is not intended for use on all document designs, so it is important to review DOCX considerations before selecting or creating a document to generate as DOCX output. DOCX output can be opened and edited by any recipient. The editability of DOCX output makes the format useful for documents that contain large amounts of text that require editing by recipients (such as contracts or correspondence). However, to preserve the editing capabilities within DOCX output, some conversion changes might be applied to the placement and layout of content that is generated as DOCX output. These conversion changes are necessary because of differences between how content is designed in Exstream and how content is designed in Microsoft Word. Because of the needed conversions, you might not receive the best results if you try to send documents that use blocked page layouts (such as newsletters) or complicated tables (such as bank statements) to DOCX output, since these designs can restrict the editability of the output. To help you get the best results with DOCX output, the content in the following sections provide best practices for designing applications for use with DOCX output.

To generate or open DOCX output, you must meet the following requirements:

### DOCX requirements

Requirement	Description
Operating system	DOCX output can be opened in the following operating systems: <ul style="list-style-type: none"><li>• AIX</li><li>• HPUX</li><li>• Linux</li><li>• Sun</li><li>• Windows 7</li><li>• Windows Server 2003 SP2</li><li>• Windows Server 2008 R2</li><li>• Windows Server 2008 SP2</li></ul>
Java	Java version 1.6 or later  Make sure that the Java file ( <code>jvm.dll</code> ) is included in the system path of the computer.
Viewer	Microsoft Word 2007 or later

## 2.7.2 Application Planning Considerations for DOCX Output

To get the most from DOCX output, it is important to understand that there are differences between how content is designed in Exstream and how content is presented in a DOCX editor; these differences influence the way you must create and design an application for use with DOCX output. Before you get started with DOCX output, consider the following:

- What is the page layout of the document you are delivering to DOCX output?
- How will the output be used after it is delivered to document recipients?

The topics in the following sections discuss some of the long-term implications of the decisions you make for these two questions during the planning and design phases of an application intended for use with DOCX output. These sections also provide best practice information and considerations to help you create the most efficient, versatile design possible.

This section discusses the following topics:

- “[DOCX Page Layout and Appearance Considerations](#)” below
- “[DOCX Document Recipient Considerations](#)” on the next page

For information about other design questions you should ask when planning an application, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### DOCX Page Layout and Appearance Considerations

In DOCX output, content is meant to flow directly from one page to the next. This single, flowing design can be very different from a traditional Exstream design, which allows you to design full pages and control the position of each object on a page. Because of these differences, the layout you design in Exstream will not be identical to the output you receive with DOCX output.

These differences in how the page layout of a document is handled can also influence other planning considerations. Keep in mind the following considerations as you design the flow of content in an Exstream application intended for use with DOCX output:

#### Page layout considerations

DOCX output behavior	How the behavior impacts your planning
Content delivered to DOCX output might appear different from the same content delivered to other output formats	If you use multiple output formats in combination with DOCX output, you should not plan for the DOCX output to be identical to output from other formats.  Since each output can compose content differently on the page, the same content can appear different when delivered to multiple output types. This means that the DOCX output might not look identical to output generated using a different output format (such as PDF). If you compare DOCX output with another output, there can be differences in the page layout, page breaks, word wrapping, or other aspects of appearance depending on how you set up your design.

### Page layout considerations, continued

DOCX output behavior	How the behavior impacts your planning
DOCX output is intended for content with a single linear flow	<p>Before you select or create a document for use with DOCX output, you should consider whether the document uses a format that will be compatible with the design capabilities of Microsoft Word.</p> <p>Designs in Exstream allow you to create page-by-page designs and complex flowing documents. However, designs in DOCX are a single continuous flow of content from one page to the next, which doesn't allow for complex flowing or page layout designs.</p> <p>This means that DOCX output is most useful when creating flowing documents (such as contracts and correspondence) that contain large amounts of text. However, this also means that DOCX output might not be as editable when used with documents that use blocked page layouts (such as newsletters) or complicated tables (such as bank statements), since these designs can restrict the editability of the output.</p>
DOCX output has no equivalents for the complex relativity and embedding options available from Exstream	<p>If you want to keep objects together or keep objects grouped together in DOCX output, you should consider whether the option you use can be replicated in Microsoft Word.</p> <p>While Exstream allows complex relativity and embedding options to keep content together, DOCX is more limited in the ways it can keep objects together on the page. If your design includes relativity settings or embedded objects that are not supported or that are structured differently in DOCX, then DOCX output that is produced from the design might appear different from your intended design, or in some cases, it might not be fully editable.</p> <p>For example, in Exstream, you can set objects to be relative to the top, left, right, or bottom of another object. In DOCX output, design objects work together as if they are relative only to the object(s) above. Creating a design in Exstream which requires multiple directions of relativity can restrict the editability of content when changes are made in DOCX output. As another example, in Exstream, you can embed a text box within another text box, which is not supported in DOCX. Including such a structure in your design can similarly restrict the editability of the content in DOCX output.</p> <p>Even though you should not use complex relativity settings or certain types of embedded objects in designs intended for DOCX output, Exstream provides several other features that allow you to keep objects together in the output.</p> <p>For more information about how to design your content to keep objects together in DOCX output, see "<a href="#">Keeping Objects Together In DOCX Output</a>" on page 75</p>

## DOCX Document Recipient Considerations

Before you select or design an application for DOCX output, you should consider how the final output will be used after it is distributed, since DOCX output can be opened and changed by any recipient. The methods you use to design content in Exstream can directly affect the editability of the content within DOCX output. To enhance the document recipient experience, you must customize your design to specifically meet the editing needs of the document recipient.

Before you set up an application for use with DOCX output, consider the following questions:

### Document recipient considerations

Planning consideration	How this impacts your design strategy
Is the DOCX output intended for recipient who will edit the content, or is it intended as a static document that can be opened and read in Microsoft Word?	<p>The document design must meet specific requirements in order for the DOCX output to maintain editability or fidelity to the original design.</p> <p>To preserve editability, when content is delivered to DOCX output, the engine automatically converts or modifies much of the design layout to allow content to conform to the design needs and limitations of Microsoft Word. For example, since messages and paragraph objects do not exist in Microsoft Word, this content is arranged in DOCX output to be inline with the main flow of content.</p> <p>Microsoft Word is a generally available program and DOCX output can be useful for ease of access. However, because of the changes needed to support Exstream content in Microsoft Word, you should consider whether fidelity to the original design is important. If you want a static, uneditable deliverable in a format that is easily accessible to document recipients, consider using an output that is capable of preserving the original Exstream design (such as PDF output).</p>
How much text will be edited by document recipients in the DOCX output? For example, will document recipients be adding new sections to a contract or will document recipients be removing content that doesn't pertain to them?	<p>You might need to make different design choices in Exstream based on the final editing needs of the document recipient. For example, in Exstream there are multiple ways to add text to a design (such as text boxes, messages, and paragraphs). However, each of these objects is reproduced differently and will have a different level of editability in DOCX output, depending on the setup of your design and the layout of your page. To get the best results when creating text intended for DOCX output, you should have an idea of how much of the text content will be changed by recipients after the DOCX output is distributed.</p> <p>For more information about using text in DOCX output, see <a href="#">"Designing Editable Text for DOCX Output" on page 75</a></p> <p>Another key consideration is that in DOCX output there are no restrictions to the amount of changes or types of content that can be added to the document after it has been distributed. Since the document has no editing restrictions, DOCX output is a useful format if output is distributed to content experts (such as insurance or marketing departments) for planned changes to content. However, if you want more control over the content that can be added to a document after it is distributed (such as pre-selecting images or pre-writing the content that can be selected by a recipient to customize the document), consider using the interactive (Live) capabilities of Exstream.</p> <p>For more information about using Live output, see <i>Designing for LiveEditor</i> in the Exstream Design and Production documentation.</p>
Will document recipients be updating content that was formerly controlled by variables in Exstream (such as counts, automated totals, and customer names)?	<p>After content is generated as DOCX output, the content is static and limited to the editing capabilities and controls available in Microsoft Word. In Exstream there are multiple tools available that allow you to dynamically change content only at the time the content is generated as output, including customer data, rules, and variables. However, changes made by a user editing DOCX output will not cause updates or changes to any of these settings in Exstream. For example, suppose you use customer data to fill in customer deadlines in a simple table. After the table is generated as DOCX output, any changes a user makes to table content will not affect the customer data.</p> <p>In Exstream, variables allows you to dynamically import values from customer data (such as customer names or account balances) directly into a design. However, after the content is produced as DOCX output, variables are no longer connected to the customer data and the variable values become static values in the output.</p> <p>If you use variables to count or calculate specific values, any change the document recipient makes to these values in the DOCX output will no longer update any connected values for counts or totals. The DOCX recipient will be required to validate any changes they make and manually update any connected counts and totals.</p>

## 2.7.3 Design Considerations for Controlling the Page Layout for DOCX Output

To achieve the page layout you want for DOCX output, it is important to understand the relationship between the page objects you use in Exstream and the page layout settings in DOCX output. In Exstream, you can use several different tools to specify the layout of the pages in the document (such as paper types, page templates, and design frames). DOCX output, however, doesn't use the same page-by-page layouts you use to create content in Exstream. For example, in Exstream, you can create each page separately, whereas in Microsoft Word content is intended to flow continuously from one page to the next. By taking into account the differences discussed in this section, you can help ensure that your DOCX output contains the desired page layout.

To control the page layout used by the DOCX output, keep in mind the following guidelines:

To control this in DOCX output	Do this in your Exstream design
Page size and orientation	<ul style="list-style-type: none"><li>Microsoft Word cannot replicate page orientation changes as they are defined in Exstream. Do not mix paper types or page templates with different sizes and orientations in content that is intended for DOCX output.</li><li>You must set the page orientation to match the page size. For example, if you want to use a portrait page, create an 8 1/2 x 11 page with portrait orientation. Likewise, if you want to use a landscape page, create an 11 x 8 1/2 page with landscape orientation. Keep in mind that if you mix page sizes with opposite orientations, content can become truncated if the DOCX output is printed from Microsoft Word.</li></ul> <p><b>Note:</b> DOCX output does not support reversed landscape or reversed portrait orientations</p>
Page margins	Use flow frames to achieve the page margins you want in DOCX output. The distance from the edge of frames in Exstream designs can be used to specify each margin measurement applied to DOCX output. For example, the distance from the left-most edge of a frame to the left-most edge of the page is used as the left margin. The same process applies to each page margin.
Page columns	Page columns can be controlled in a design either by text boxes or by flow frames, depending on which design objects you use to contain text and where these objects are placed in the design.  For more information about deciding which design objects are best for the type of text content in your particular design, see " <a href="#">Designing Editable Text for DOCX Output</a> " on page 75.

## 2.7.4 Designing Applications to Maximize the Editability of DOCX Output

DOCX output can be edited by document recipients; therefore, you must take into account some special design considerations in order to maximize the editability of the generated DOCX output. Because of the constraints of Microsoft Word, there are more design considerations for DOCX output than for most other output formats in Exstream. For example, Microsoft Word

does not have the capability to manage advanced tables or customer data in the same way you are able to in Exstream. You must also consider how the document will be used by document recipients. For example, if document recipients will need to make extensive edits to the content, you might benefit from using a particular design object in Exstream to create the design.

In addition, for most print and electronic outputs available in Exstream, your Design and Production setup exclusively controls the placement of all objects in your design. However, with DOCX, the final rendering is dependent on the software you use to open the DOCX file (such as Microsoft Word). This means that content placement (such as flow, page breaks, the number of lines of text on a page, and whether a table can fit on one page or split to two pages) can differ between Exstream and the DOCX output.

This section discusses the following topics:

- “[Controlling the Flow of Content in DOCX Output](#)” below
- “[Keeping Objects Together In DOCX Output](#)” on the next page
- “[Designing Editable Text for DOCX Output](#)” on the next page
- “[Designing Tables for DOCX Output](#)” on page 77

## Controlling the Flow of Content in DOCX Output

Content flows differently in DOCX output than it does in Exstream.

For maximum editability, use the following best practices:

- **Use a single flow area that continues uniformly from one page to the next**—Exstream supports multiple flows and allows you to split content across multiple pages. For example, suppose you use complex flow to allow content to skip from page 1 to page 4 in the output. When the design is produced as DOCX output, the content is initially placed on the correct pages. However, if content is added to page 1, it will no longer flow to page 4. Because of the continuous flow in DOCX, if a document recipient makes changes that overflow page 1, the content will automatically continue on to page 2. A single flow that continues from one page to the next allows users to edit large amounts of content without impacting the remaining layout of the document.

For more information about setting up a flow page, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Use a single flow frame per page**—Unlike Exstream, DOCX does not have the ability to manage separate flows of content. All content in Microsoft Word is managed in a single continuous flow of content. When you design content for DOCX output, a single flow frame per page ensures that content flows as expected when the content is generated as DOCX output and when document recipients make changes to the content. Using multiple flow frames per page can yield unexpected results.

For more information about designing flow frames, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## Keeping Objects Together In DOCX Output

In a design, it is often important that specific objects are grouped together so that they always appear on the same page. For maximum editability, use the following best practices to keep objects together in DOCX output:

- **When using relativity, use only "relative to above" settings**—Because DOCX content does not have the advanced design capabilities available in Exstream, DOCX output is not capable of complex "left," "right," and "below" relativity settings. "Relative to above" settings allow you to keep an object positioned in the output relative to the object above it.
- **When embedding objects in a design, use inline or floating embedding instead of linked embedding**—Microsoft Word has limited support for embedded objects from Exstream. For best results, you should use inline or floating embedded objects, which by default have the same behaviors as embedded objects in Microsoft Word. If you do use linked embedding, keep in mind that DOCX output supports embedded objects that are linked to text box content, but not embedded objects linked to table cell content.
- **If you are embedding objects in text, it is best to place the embedded object on a separate line from the text**—The word wrap settings you can apply to embedded objects in Exstream do not carry over to the DOCX output. When embedded objects are separated from the text, the embedded object and text are imported as expected. However, if you include embedded objects within or adjacent to text, the image and text will overlap when generated as DOCX output, and document recipients must manually apply word wrapping to the embedded object in Microsoft Word.

For more information about using relativity, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about embedding objects, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## Designing Editable Text for DOCX Output

Exstream provides multiple objects that allow you to add text to your design; however, each object can be handled differently (depending on your design) when delivered to DOCX output. The main reason for differences in text behavior is that the main body text in DOCX output is placed directly onto a page, whereas in Exstream, all text is contained within objects (such text boxes, paragraph objects, and messages).

Use the following guidelines to help you choose the objects that are best suited to contain the text in your design:

- **In general, paragraph objects are the best design objects to use for text intended for DOCX output**—Paragraph objects in Exstream are generally used to build complex documents that contain large amounts of text (such as contracts and policy documents). Paragraph objects from Exstream are transformed into the main body text of DOCX output. This placement allows document recipients to add large amounts of text to the output without impacting the layout or flow of the DOCX output.

- **Text boxes can behave differently in DOCX output depending on how they are included in the Exstream design**—Text boxes can be transformed to be either main body text or text boxes in DOCX output, depending on the relativity of your content or where the text box is placed in the design. Text box content will either be placed into the main flow of content or will be placed as a Microsoft Word text box.

For more information about controlling how relative content or embedded objects are handled in DOCX output, see “[Keeping Objects Together In DOCX Output](#)” on the previous page.

- **Text boxes in Microsoft Word can limit editing capability**—While text boxes in Exstream are allowed to grow or shrink to fit content, text boxes in DOCX output must be manually resized to fit content. This behavior makes text boxes well suited for content that requires minor changes (such as name changes or word replacements) that do not have a major impact on the amount of text within the text box. However, if users must make major changes to text box content (such as adding or removing text paragraphs), then document recipients might be required to resize text boxes. Resizing a text box in Microsoft Word can also alter the way other content flows around the text box.
- **Not all message types are supported by Microsoft Word**—DOCX output offers the following support for message objects:
  - Text messages are supported and are placed into the main body flow in DOCX output.
  - Graphic messages are supported and are placed into the main body flow in DOCX output.
  - Insert messages are not supported in DOCX output, since insert messages are intended to be preprinted content that is inserted into the output after it has been generated.
  - Graphic or insert messages are supported if you included a graphic message design within the graphic or insert message. The graphic message design is placed into the main body flow.
- **For multi-column output, you should use paragraph objects to contain text**—Because paragraph object content is always placed into the main flow of text, paragraph objects give the best results with multi-column content. The column settings are controlled by the flow frame that accepts the paragraph objects. You should avoid using text boxes for multi-column output since text boxes can be placed either into the main flow or placed as DOCX text boxes, depending on your design. Columns are replicated in DOCX output for text boxes only if the text box content will be applied to the main flow of the content. If the text box will remain as a text box in the DOCX output, Microsoft Word does not support columns in text boxes, and the text box will be generated as a single column of text.

Keep in mind that the placement of text can also be affected by the layout of frames, columns, and relativity settings in your document design.

For more information about understanding how the layout of your design impacts DOCX output, see “[Design Considerations for Controlling the Page Layout for DOCX Output](#)” on page 73.

For more information about using relativity to keep objects together in DOCX output, see “[Keeping Objects Together In DOCX Output](#)” on the previous page.

## Designing Tables for DOCX Output

DOCX output does not have the same complex table capabilities as Exstream. Although DOCX output can support tables, it is important to understand the best practices needed to successfully deliver table content to DOCX output. Use the following best practices to ensure that tables included in DOCX output remain accessible and editable:

- **For best results, use simple tables when possible**—DOCX output is not intended for documents containing complex tables (such as bank statements or financial reports). Simple tables translate to DOCX output better than other table types because all simple table design and formatting options are supported in DOCX output.
- **Automated rows and columns are supported but do not continue to automate after they are generated as output**—You can use automated rows and columns when building a table in Exstream to allow content to grow based on customer information. After the content is generated, however, rows that were automated in Exstream are treated as normal rows and cells in DOCX output. Also, because these cells are also no longer connected to a customer database after they are generated as output, any changes made to the content of these rows in DOCX output will not appear in the customer database.

### 2.7.5 Exstream Features Supported in DOCX Output

When designing content intended for DOCX output, keep in mind that DOCX output might not have an equivalent for all Exstream options. For example, Microsoft Word does not offer a way to support advanced table automation and setup in the same way Exstream can. To ensure the best results when creating DOCX output, you should review the following topics to determine the level of support DOCX output offers for the design features you want to use.

In the following tables, features are identified as having one of the following support levels:

- **Supported**—Supported options are reproduced accurately in DOCX output.
- **Partially supported**—Partially supported options are reproduced with some minor modifications because of differences in the features available in Exstream and DOCX output. Options marked as partially supported are accompanied by support notes to help you understand the difference between how the feature is handled in Exstream and in DOCX output.
- **Not supported**—Unsupported options are not reproduced in DOCX output. Options marked as not supported are accompanied by support notes to help you understand how these features are handled if you choose to leave these features in an application delivered to DOCX output.

This section discusses the following topics:

- “[DOCX Output Support for Exstream Text Formatting](#)” below
- “[DOCX Output Support for Exstream Text Boxes](#)” on page 80
- “[DOCX Output Support for Exstream Paragraph Objects](#)” on page 81
- “[DOCX Output Support for Exstream Message Objects](#)” on page 82
- “[DOCX Output Support for Exstream Images, Charts, and Shapes](#)” on page 83
- “[DOCX Output Support for Exstream Tables](#)” on page 85
- “[DOCX Output Support for Exstream Hyperlinks](#)” on page 89

## DOCX Output Support for Exstream Text Formatting

DOCX output support for Exstream text formatting includes the following:

Text formatting feature	Support level	Notes
Alignment	Supported	—
Bullets	Supported	—
Breaks (column, line, page, non-breaking space)	Partially supported	In DOCX output, a column break is supported only in text included in the main body flow. If you use column breaks in paragraph objects or text boxes, make sure that these objects are designed to be the main body flow in DOCX output.  All other breaks are fully supported.
Cross-references	Supported	—
Fonts	Supported	—
Font color	Supported	—
Font size	Supported	—
Footnotes	Partially supported	<ul style="list-style-type: none"><li>• In DOCX output, footnotes are always placed at the foot of the document body, regardless of where they appear in the Exstream design.</li><li>• DOCX does not support footnote frames. If you include footnote frames in content delivered to DOCX, the frames are ignored.</li><li>• In DOCX output, footnotes can be placed only on the current page.</li></ul>
Hyphenation	Supported	—

Text formatting feature	Support level	Notes
Hyperlinks	Supported	<p>In DOCX output, hyperlinks are supported. The link addresses can be static or dynamic (set by variable values).</p> <p>For more information about using hyperlinks in DOCX output, see “<a href="#">DOCX Output Support for Exstream Hyperlinks</a>” on page 89</p>
Indenting	Supported	—
Indexes	Partially supported	<ul style="list-style-type: none"> <li>• In Exstream, the content for index markers is automatically generated by the engine, based on the text content to which the index marker is connected. However, when the content is generated as DOCX output, the content of the index marker and the text content are no longer connected and the index marker content will no longer be updated automatically when the text content is changed.</li> <li>• When indexes are generated in DOCX output, they will appear as blank. Document recipients must manually update the index in order for the index to appear in the DOCX output.</li> <li>• DOCX output supports only the default formatting for indexes.</li> <li>• In DOCX output, indexes do not display duplicate page numbers for the same index entry.</li> </ul>
Kerning	Supported	—
Leader dots	Partially supported	<p>DOCX output supports the following leader dot formats:</p> <ul style="list-style-type: none"> <li>• Period (.)</li> <li>• Underscore (_)</li> <li>• Hyphen (-)</li> </ul> <p>If you use a leader dot format that is not supported, the leader dots are generated using the period (.) format.</p>
Line spacing	Partially supported	<p>Line spacing is calculated differently in Exstream and DOCX output; therefore, when you generate DOCX output, there might be some minor spacing differences from what was designed in Exstream.</p>
Page numbering	Partially supported	<ul style="list-style-type: none"> <li>• In DOCX output, document level page counts are not supported.</li> <li>• In DOCX output, page counts are always reset at the beginning of a new customer.</li> </ul>
Paragraph types	Partially supported	<p>In DOCX output, headers and footers are supported, provided that they are the very first or the very last entry in the design (respectively).</p> <p>However, DOCX output does not have the capability to recreate repeating headers or repeating footers. If the text paragraph uses a repeating paragraph type, the paragraph is generated as regular text.</p>

Text formatting feature	Support level	Notes
Tables of contents (TOCs)	Partially supported	<ul style="list-style-type: none"><li>In Exstream, the content for TOC markers is automatically generated by the engine, based on the text content to which the TOC marker is connected. However, when the content is generated as DOCX output, the content of the TOC marker and the text content are no longer connected; therefore, the TOC marker content will no longer be updated automatically when the text content is changed.</li><li>DOCX output supports only one table of contents per document.</li><li>DOCX output does not support sub-customer tables of contents.</li><li>DOCX output does not support wrapping text in a table of contents.</li></ul>
Tab stops	Partially supported	DOCX output does not support custom tab stops with leader dots.
Vertical writing	Supported	—

For more information about formatting paragraph objects in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Output Support for Exstream Text Boxes

DOCX output support for Exstream text boxes includes the following, listed by interface option:

Text box formatting option	Support level	Notes
Autofit text	Supported	—
Autosize height, Autosize width	Partially supported	In DOCX output, auto-sizing width can only auto-size a text box to the width of the main body of text, even if the text box being auto-sized is in a header or footer.
Columns	Partially supported	DOCX output does not support columns with the column style of <b>Spans columns</b> .
Fill	Partially supported	DOCX output does not support hatched fill patterns. If you apply a hatched fill pattern and generate DOCX output, the fill will be generated as a solid pattern.
Frame style	Partially supported	<p>DOCX output does not have an equivalent for some frame styles available in Exstream.</p> <p>DOCX output supports the following frame styles:</p> <ul style="list-style-type: none"><li>Frame</li><li>Top Line</li><li>Bottom Line</li><li>Top and Bottom</li><li>Double</li></ul> <p>If the design contains an unsupported frame style, then the text box is generated in DOCX output with the default frame style of <b>Frame</b>.</p>

Text box formatting option	Support level	Notes
<b>Grow ( Down)</b>	Supported	In DOCX output, text boxes can grow downward.
<b>Grow ( Up)</b>	Not supported	DOCX does not have the ability for a text box to grow upward. If you include this option in your design, the text box will be static in the DOCX output.
<b>Left margin, Right margin, Top margin, Bottom margin</b>	Partially supported	Margin settings are supported in DOCX output. However, DOCX allows a maximum margin of 31 points in a text box. While you can design a text box in Exstream with a larger margin, the text box margins will be produced in DOCX output at the 31 point maximum.
<b>Convert polygon to text</b>	Not supported	DOCX does not support shapes that have been converted into text boxes. If you include a shape that has been converted in to a text box in content delivered to DOCX output, the text box will be drawn as a default rectangular text box in the DOCX output
<b>Shadow</b>	Not supported	DOCX output handles shadows on text boxes differently than Exstream. If you deliver a text box with a shadow applied to DOCX output, the shadow is removed.
<b>Can split and flow</b>	Supported	—
<b>Widow control</b>	Partially supported	Widow/orphan controls work differently in DOCX output than in Exstream. Widow/orphan settings are kept, but are limited to the capabilities of DOCX output.  In Exstream, widow/orphan controls allow you to set the number of lines that must appear at the end of one page or at the beginning of the next page. The same setting in DOCX output specifies whether the first line of a paragraph can be placed at the end of a page.
<b>Wrap</b>	Partially supported	Word wrapping controls work differently in DOCX output than in Exstream. In Exstream, you can specify how the text within the text box flows around other objects in the page. In DOCX output, wrapping controls how the text outside of the text box flows around the text box itself.  Wrap settings are kept, but are limited to the capabilities of DOCX output.

For more information about formatting text boxes in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Output Support for Exstream Paragraph Objects

DOCX output support for Exstream paragraph objects includes the following, listed by interface option:

Paragraph object formatting option	Support level	Notes
<b>Flow margin</b>	Supported	—
<b>Keep with next paragraph</b>	Supported	—

Paragraph object formatting option	Support level	Notes
<b>Keep with previous paragraph</b>	Partially supported	DOCX does not have a "keep with previous" setting. However, to preserve the paragraph grouping, any paragraph that is placed before a paragraph with a <b>Keep with previous paragraph</b> setting in an Exstream application will be set to "keep with next" in the DOCX output. This allows you to keep the same functionality even though the settings are slightly different.
<b>Link to URL (When included in electronic output)</b>	Supported	—
<b>Renumber text when placed in frame</b>	Supported	—
<b>Can split across frames</b>	Partially supported	In Exstream, if the <b>Can split across frames</b> check box is cleared (the paragraph is set not to split), the text within the paragraph object will have the following attributes when generated as DOCX output: <ul style="list-style-type: none"> <li>To prevent individual text paragraphs from splitting, each line of text is set to "keep lines together."</li> <li>To prevent groups of text paragraphs from splitting, each text paragraph is set to "keep with next."</li> </ul>
<b>Widow/orphan control</b>	Partially supported	Widow/orphan controls work differently in DOCX output than in Exstream. Widow/orphan settings are kept, but are limited to the capabilities of DOCX output.  In Exstream, widow/orphan controls allow you to set the number of lines that must appear at the end of one page or at the beginning on the next page. The same setting in DOCX output controls whether the first line of a paragraph can be placed at the end of a page.

For more information about formatting paragraph objects in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Output Support for Exstream Message Objects

DOCX output support for Exstream message objects includes the following:

Message object type/feature	Support level	Notes
Text messages	Supported	—
Graphic messages	Supported	—
Insert messages	Not supported	Since insert messages are intended to be pre-printed content that is inserted into the final output only after it has been generated, DOCX output does not support insert messages.

Message object type/feature	Support level	Notes
Graphic or insert messages	Partially supported	<p>Graphic or insert messages allow you to design a message with both an insert message (pre-printed message) and a graphic message in case the pre-printed messages are not available.</p> <p>If you include a graphic or insert message in DOCX output, the DOCX output uses the graphic message design you included in the graphic or insert message.</p>
Columns	Supported	—

For more information about managing message content in a design, see *Managing Marketing Messages* in the Exstream Design and Production documentation.

## DOCX Output Support for Exstream Images, Charts, and Shapes

DOCX output support for Exstream images, charts, and shapes includes the following:

Object/feature	Support level	Notes
Charts	Partially supported	<p>Charts are automatically converted into images when generated as DOCX output. You control the resolution and format applied to these images by defining options on the <b>Resource Management</b> tab of the DOCX output object.</p> <p>For more information about defining the image resolution and format of images delivered to DOCX output, see <a href="#">"Defining the Image Resolution and Format for Images in DOCX Output" on the next page</a>.</p>
Barcodes	Partially supported	<p>Barcodes are automatically converted into images when generated as DOCX output. You control the resolution and format applied to these images by defining options on the <b>Resource Management</b> tab of the DOCX output object.</p> <p>Keep in mind that some barcodes can be designed with dynamic content that is intended to change in Exstream based on customer data. After the barcode is converted to an image for DOCX output, this information will no longer change as recipients make changes to the document.</p> <p>For more information about defining the image resolution and format of images delivered to DOCX output, see <a href="#">"Defining the Image Resolution and Format for Images in DOCX Output" on the next page</a>.</p>
Images	Partially supported	<p>You must select an image format to apply to images that use formats that are not supported in DOCX output.</p> <p>For more information about defining the image resolution and format of images delivered to DOCX output, see <a href="#">"Defining the Image Resolution and Format for Images in DOCX Output" on the next page</a>.</p>
Shapes	Supported	—
Shadows on all objects	Not supported	If you deliver an object with a shadow to DOCX output, the shadow is removed from the object when it is produced as DOCX output.

For more information about formatting images, charts, and shapes in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## Defining the Image Resolution and Format for Images in DOCX Output

Depending on the images you use in your application, you can control how images are converted to a DOCX-supported format.

DOCX output supports the following image formats:

- JPEG
- GIF
- PNG
- TIFF

Setting the image resolution and format to use with DOCX output is also useful if your design contains objects that are converted into images in order to be included in the DOCX output, such as charts. Depending on the images in your design, you can choose to apply the conversion settings to all images or only to those images that are in a format that is not supported in DOCX output.

To define the image conversion formats for DOCX output:

1. In Design Manager, in the Library, go to one of the following locations:
  - **Environment > Delivery > Outputs > SBCS outputs**
  - **Environment > Delivery > Outputs > DBCS outputs**
2. Drag an output object to the Property Panel.  
The output properties open in the Property Panel for you to define.
3. Click the **Resource Management** tab.
4. From the **Image conversion action** drop-down list, select one of the following options to define which images receive the conversion settings:

To	Do this
Control the resolution and formatting for only the images in formats that are not supported by DOCX output	Select <b>Only convert if original format is unsupported.</b>
Control the resolution and formatting for all images delivered to DOCX output	Select <b>Convert all images using settings below.</b>

5. From the **Specific resolution (DPI)** drop-down list, select the DPI setting you want to use from the available options, or select **Custom** and enter a specific resolution in the adjacent box.

6. From the **Format** drop down list, select the image format you want to apply to the images in DOCX output.
7. From the Menu bar, select **File > Save**.

## DOCX Output Support for Exstream Tables

This section discusses the following topics:

- “[DOCX Output Support for General Table Properties](#)” below
- “[DOCX Output Support for Row Properties](#)” on the next page
- “[DOCX Output Support for Column Properties](#)” on page 87
- “[DOCX Support for Cell Properties](#)” on page 88

### DOCX Output Support for General Table Properties

DOCX output support for general table properties includes the following, listed by interface option:

Table option	Support level	Note
Add legend boxes to rows	Not supported	Legend boxes are removed from content delivered to DOCX output.
Alternate fill in table rows	Supported	—
Autofit width	Supported	—
Add automated page totals	Partially supported	Automated page totals are initially generated in DOCX output with the appropriate values. However, since DOCX does not have an equivalent setting, these values are static in the output and will not be updated as changes are made.
Enable data section properties	Partially supported	You can use data sections in tables delivered to DOCX output. Support for data sections is also dependent on the row types included in the section.
Frame style	Supported	All frame styles are supported in DOCX output.
Keep groups together	Not supported	DOCX output does not have a setting that enables you to keep groups together. <b>Keep groups together</b> settings are removed from content that is delivered to DOCX output.
Remove empty rows	Supported	—
Split if too wide	Not supported	DOCX output does not have a setting that enables a table to split if too wide. <b>Split if too wide</b> settings are removed from content that is delivered to DOCX output.
Target width	Supported	Generating a table targeting a specific table width is supported in DOCX output.

For more information about formatting tables in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Output Support for Row Properties

DOCX output support for row properties includes the following, listed by interface option:

Row option/feature	Support level	Notes
Row type: <b>Header (H)</b>	Supported	—
Row type: <b>Header – If table at top of flow frame (Hpf)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be treated like a standard row.
Row type: <b>Header - If table is not at top of flow frame (Hmf)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be completely removed from the table.
Row type: <b>Repeating header (H+)</b>	Partially supported	A repeating header that is not contained within a table section is generated as a repeating header in Microsoft Word.  However, a repeating header that is contained within a table section is generated as a standard row in DOCX output.
Row type: <b>Repeating header – Except first page (H-)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be completely removed from the table.
Row type: <b>Standard row – Not automated</b>	Supported	—
Row type: <b>Automated row – Repeating on repeat criteria (R)</b>	Partially supported	Automated rows are initially generated in DOCX output with the appropriate values. However, after the rows are generated as DOCX output, the rows will not retain their automated properties and will behave as standard rows.
Row type: <b>Divider row – Repeating on defined criteria (D+)</b>	Partially supported	Divider rows are generated containing the appropriate data. However, after the rows are generated as DOCX output, the rows will not retain their automated properties and will behave as standard rows.
Row type: <b>Section header – First occurrence on page (Hp)</b>	Partially supported	Section header rows are generated containing the appropriate data. However, after the rows are generated as DOCX output, the rows will not retain their automated properties and will behave as standard rows.
Row type: <b>Section header – Not first occurrence on page (Hm)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be completely removed from the table.
Row type: <b>Footer (F)</b>	Supported	—
Row type: <b>Repeating footer (F+)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be completely removed from the table.
Row type: <b>Repeating footer – Except last page (F-)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be completely removed from the table.

Row option/feature	Support level	Notes
Row type: <b>Sub-section footer (Fs)</b>	Not supported	DOCX does not have an equivalent feature. If this row type is included in any table delivered to DOCX output, the row will be completely removed from the table.
<b>Auto width</b>	Supported	—
<b>Autosize height during design</b>	Supported	Auto-sizing row height is supported in DOCX output.
<b>Line above, Line below</b>	Supported	The border types of <b>Line above</b> and <b>Line below</b> are supported in DOCX output.
<b>Cell widths are adjustable</b>	Supported	—
<b>Fixed height in Engine</b>	Supported	—
<b>Group with previous row</b>	Not supported	Since DOCX does not have an equivalent option, rows are not grouped in DOCX output.
<b>Include legend box</b>	Not supported	DOCX does not have an equivalent feature. If legend boxes are included in any table delivered to DOCX output, the legend box will be completely removed from the table.
<b>Number of serpentine, Include serpentine</b>	Not supported	Serpentine cells are not supported in DOCX output. If you deliver a table containing serpentine cells to DOCX output, the serpentine cells are generated as regular cells.
<b>Can split</b>	Supported	—
<b>Table section (set of rows)</b>	Partially supported	Table sections are generated containing the appropriate data. However, after a section is generated as DOCX output, the section will not retain any automated properties and all rows within the section will behave as standard rows.

For more information about formatting tables in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Output Support for Column Properties

DOCX output support for column properties includes the following, listed by interface option:

Column option	Support level	Notes
<b>Autofit width</b>	Supported	—
<b>Line left, Line right</b>	Supported	The border types of <b>Line left</b> and <b>Line right</b> are supported in DOCX output.
<b>Maximum width</b>	Supported	—
<b>Minimum width</b>	Supported	—
<b>Width</b>	Supported	—

For more information about formatting tables in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Support for Cell Properties

DOCX output support for cell properties includes the following, listed by interface option:

Cell option	Support level	Notes
Autofit text	Supported	—
Autojoin cells	Partially supported	Cells are able to auto-join only in the engine. After the content is generated as DOCX output, the cells will no longer have the ability to auto-join.
Background	Supported	—
Background style	Not supported	All cells are generated in DOCX output with the <b>Rectangle</b> background style.
Columns	Not supported	DOCX output does not have a equivalent to columns within a table cell. A cell containing columns is generated as a cell with a single column of text in DOCX output.
Conditional colors	Partially supported	Cell content is generated using the appropriate conditional colors. However, after the content is generated as DOCX output, the cells will not retain any automated properties and all conditional colors within the cell become static and will no longer change as edits are made to the content.
Grid lines	Supported	—
Join cells	Supported	Joined/merged cells are supported in DOCX output.
Kerning	Supported	—
Left margin, Right margin, Top margin, Bottom margin	Supported	All margin options are supported in DOCX output.
Widow control	Supported	—
Wrap	Supported	Text wrapping in cells is supported in DOCX output.
X/Y Font scaling	Not supported	—

For more information about formatting tables in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## DOCX Output Support for Exstream Hyperlinks

DOCX output supports the inclusion of hyperlinks in the output file. You can design hyperlinks on pages for specific text or a paragraph within a text box. You can also assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The link addresses can be static or dynamic (set by variable values).

The following table includes guidelines for following hyperlinks. For more information about following hyperlinks, refer to the documentation for the software that an end user will use to view DOCX output.

To follow a hyperlink address in DOCX output, an end user must complete one of the following tasks:

To	Do this
To follow a hyperlink on text or other objects within a text box or table	Hover the pointer over the text or object with a hyperlink and use <b>CTRL + Click</b> .
To follow a hyperlink on objects that are placed directly on a page (not within a text box or table)	Hover the pointer over an object with a hyperlink, right-click, and select <b>Open Hyperlink</b> .

Keep in mind that an end user can view a hyperlink address in a tooltip only if the text or object is within a text box or table. To view a hyperlink address in a tooltip, an end user must complete one of the following tasks:

To	Do this
To view the hyperlink address for an object that is in a text box	<ol style="list-style-type: none"><li>Select the text box that includes text or an object with a hyperlink.</li><li>Hover the pointer over the text or object with a hyperlink.</li></ol>
To view the hyperlink address for an object that is in a table	<ol style="list-style-type: none"><li>Select the table that includes text or an object with a hyperlink.</li><li>Hover the pointer over the text or object with a hyperlink.</li></ol>

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

## 2.8 Setting Up EDGAR HTML Output

This section discusses the following topic:

- [“Showing HTML Revision Tags in an HTML Viewer for EDGAR HTML” on the next page](#)

In addition to the topic discussed in this section, you can also use the following basic setup options for EDGAR HTML output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Creating MIME HTML output	<a href="#">“Creating MIME HTML Output To Include All Resources in a Single File” on page 95</a>
Adding hyperlinks at the top of each HTML page	<a href="#">“Adding Hyperlinks at the Top of Each HTML Page” on page 100</a>

EDGAR HTML supports the inclusion of hyperlinks in the output file. You can design hyperlinks on pages for specific text or a paragraph within a text box. You can also assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The link addresses can be static or dynamic (set by variable values). An end user will use a single click to follow hyperlink addresses in EDGAR HTML output.

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

EDGAR HTML supports the inclusion of metadata tags on the application object in the output file. You can add metadata to EDGAR HTML output following the same steps as you would for other objects in Design Manager and Designer. Because of limitations with EDGAR HTML, only metadata tags on application objects appear in the output.

For more information on creating and adding metadata in Design Manager and Designer, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Because of the specifications for HTML files submitted to the SEC with EDGAR, some HTML output options are not applicable for EDGAR HTML output.

The limitations are as follows:

- **Single-file HTML**—You cannot create single-file HTML.
- **Embedded tables**—You cannot use embedded tables.
- **Graphics**—EDGAR HTML does not support SVG format. Images are created as external GIF images and referenced as necessary.
- **Image pass-through**—The only image pass-through available for EDGAR HTML is JPEG.
- **Style sheets**—Style sheets are not available for EDGAR HTML. You must define all attributes for each object individually, including recurring objects. Both `<style>` and `<span>` HTML tags are restricted.
- **HREF on images**—Certain HREFs are allowed, but these HREFs are limited by EDGAR specifications.

## 2.8.1 Showing HTML Revision Tags in an HTML Viewer for EDGAR HTML

1. In Design Manager, from the Library, drag an EDGAR HTML output object to the Property Panel.
2. Click the **Advanced** tab.
3. Select the **Show HTML revision tags** check box. If you do not select this check box, you can see revision tags (`<r></r>`) only in the source code.
4. Save the output object.

## 2.9 Setting Up HTML Output

When you add an HTML output object or output queue to your application, Exstream produces a set of files in the output directory that you specify in the output object properties. This includes the following files:

- **HTML output file (.html)**—The source HTML file
- **GIF files (.gif) or SVG (.svg) files**—These files contain the graphic design objects (such as charts, lines, and shapes) that you added to your design using the drawing tools in Designer. The engine converts these design objects into separate image files in the format specified in the output object properties, and then references those image files in the HTML output, along with any images you imported at run time in their original format (such as JPEG or PNG).

- **External Cascading Style Sheets files (.css)**—These files contain the styles that are used in the output. You receive multiple CSS files if you are delivering container designs in the output and you are using additional cascading style sheets to control the final appearance of your container design.

**Note:** Exstream uses some tags with HTML output that are unnecessary for EDGAR HTML, including <style> and <span>. Because Exstream removes these tags for EDGAR HTML output, the appearance of the output between HTML output and EDGAR HTML output will be different.

The following table identifies the ways in which you can customize your HTML output:

For information about	See
Selecting the HTML version to use when producing output	<ul style="list-style-type: none"><li>• <b>Producing HTML 4.01 Transitional or HTML5</b>—“<a href="#">Creating Basic HTML Output</a>” on the next page</li><li>• <b>Producing archival (MIME) HTML</b>—“<a href="#">Creating MIME HTML Output To Include All Resources in a Single File</a>” on page 95</li></ul>
Formatting the HTML output using cascading style sheets	<ul style="list-style-type: none"><li>• <b>Using cascading style sheets in Exstream</b>—<i>Designing Customer Communications</i> in the Exstream Design and Production documentation</li><li>• <b>Specifying whether styles are included inline or at the top of the file</b>—“<a href="#">Specifying Where Styles are Included in HTML Output</a>” on page 96</li><li>• <b>Adding class attributes to objects</b>—<i>Designing Customer Communications</i> in the Exstream Design and Production documentation</li><li>• <b>Adding a prefix to CSS file names when you have multiple HTML output queues</b>—“<a href="#">Avoiding File Name Conflicts with CSS Files When Using Multiple HTML Output Objects</a>” on page 105</li></ul>
Selecting which fonts to display if the fonts used in the design are not supported by the browser	“ <a href="#">Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output</a> ” on page 114
Applying a background color to the HTML output	“ <a href="#">Creating Output</a> ” in the Exstream Design and Production documentation
Including hyperlinks to sections within the HTML document at the top of the HTML file	“ <a href="#">Adding Hyperlinks at the Top of Each HTML Page</a> ” on page 100
Changing the file format that the engine produces for each graphic design objects (such as a chart, line, or polygon) in your design from GIF to SVG format	“ <a href="#">Using Scalable Vector Graphic Format for Graphic Design Objects</a> ” on page 99
<p><b>Note:</b> This option applies only to HTML 4.01 Transitional output. For HTML5 output, the engine produces only SVG graphics.</p>	

For information about	See
Adding metadata (a <code>meta</code> element) to the <code>&lt;head&gt;</code> section of the HTML file	<a href="#">"Adding Metadata (&lt;meta&gt; Elements) to HTML and HTML (Email) Output" on page 115</a>
Using an XSL file to perform post-processing tasks to transform the HTML output for final presentation	<a href="#">"Setting Up to Transform HTML Output" on page 117</a>
Using a DDA output file to perform post-processing tasks to manipulate output before it reaches the print stream	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Highlighting tracked changes in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>

## Related information

*Designing Customer Communications* in the Exstream Design and Production documentation

["Creating Basic HTML Output" below](#)

### 2.9.1 Creating Basic HTML Output

When you create basic HTML output, you must specify which version of HTML that you want to produce and whether to deliver standard designs or container designs in the output.

To create basic HTML output:

1. If you do not know the application mode of the database in which you are working, review the **Application mode** drop-down list on the **Workflow** tab of the **System Settings** object.  
For information about the application mode of a database, see *System Administration* in the Exstream Design and Production documentation.
2. Based on the application mode of the database in which you are working, do one of the following:

For this application mode	Do this
SBCS	<ol style="list-style-type: none"><li>In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs</b>.</li><li>Right-click the <b>Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li></ol>
DBCS	<ol style="list-style-type: none"><li>In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs</b>.</li><li>Right-click the <b>Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li></ol>

For this application mode	Do this
SBCS/DBCS	<ul style="list-style-type: none"> <li>• To create SBCS output, complete the following steps:           <ul style="list-style-type: none"> <li>i. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs &gt; SBCS Outputs</b>.</li> <li>ii. Right-click the <b>SBCS Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li> </ul> </li> <li>• To create DBCS output, complete the following steps:           <ul style="list-style-type: none"> <li>i. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs &gt; DBCS Outputs</b>.</li> <li>ii. Right-click the <b>DBCS Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li> </ul> </li> </ul>

3. In the **Name** box, enter a name for the output object.

4. In the **Description** box, enter a description (optional).

5. Click **Finish**.

The output object opens in the Property Panel for you to define.

6. From the **Driver** drop-down list, select **HTML**.

7. From the **Version** drop-down list, select one of the following options, depending on the version of the HTML output that you want to produce:

To	Do this
Produce HTML5 output	Select <b>5</b> .
Produce HTML 4.01 Transitional output, which might be suitable for legacy applications	Select <b>4.01 Transitional</b> .
Produce MIME HTML (MHTML) output, which allows you to include all of the resources for the HTML output in a single file  For more information about creating MHTML, see " <a href="#">Creating MIME HTML Output To Include All Resources in a Single File</a> " on the next page.	Select <b>MIME</b> .  <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <b>Note:</b> If you are creating output from a standard design on a z/OS system, you must create single-file MIME HTML (MHTML) output.       </div>

For more information about the differences among versions of HTML produced by Exstream, see "[HTML](#)" on page 22.

8. Select the type of designs that you want to deliver in the output:

To	Do this
Deliver standard designs in the output (default)	Clear the <b>Use container design</b> check box.
Deliver container designs in the output	a. Select the <b>Use container design</b> check box. b. In the <b>Container design</b> label box, click  and select the name of the container design that you want to deliver in the output.

For more information about container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## 2.9.2 Creating MIME HTML Output To Include All Resources in a Single File

Exstream lets you create one output file that includes all of the resources for the HTML output in a single file. This output is known as MIME HTML, or MHTML. If you are creating output from a standard design on a z/OS system, you must create single-file HTML output.

Keep in mind that many web browsers do not natively support viewing MHTML.

**Note:** If you are creating output from a standard design on a z/OS system, you must create MHTML output.

To create MHTML:

1. In Design Manager, from the Library, create an HTML output object, or drag an existing HTML output object to the Property Panel.

For more information about creating an HTML output object, see “[Creating Basic HTML Output](#)” on page 93.

2. Click the **Basic** tab.
3. From the **Version** drop-down list, select **MIME**.
4. From the Menu bar, select **Edit > Save**.

**Note:** When you select **MIME** from the **Version** drop-down list, and you select the **Create external cascading style sheet** check box, the output is still a single MHTML file.

However, the single output file includes the CSS as a separate file part with a content type of `text/css` instead of including it directly within the HTML.

For more information about creating external style sheets or including styles within HTML, see “[Specifying Where Styles are Included in HTML Output](#)” below.

## 2.9.3 Specifying Where Styles are Included in HTML Output

When you create HTML output, you can either use an external Cascading Style Sheet (CSS) file for styles used in the HTML, or you can include style information at the top of each HTML file, within the `<head>` element.

Using an external style sheet makes it very easy to change the appearance of your HTML output outside of Exstream, but selecting this option also means that you must manage multiple files as part of your workflow, since all of the data defining your application is not included in one self-contained package file. Consider the pros and cons of both CSS placement options, and then choose the one that best suits your needs.

You specify your CSS placement preference in the properties of the HTML output object, as described in the following procedure. If you are delivering container designs in the output, and you are using custom CSS style sheets in addition to the CSS style sheets that Exstream automatically creates to manage the styles that you applied in Designer, the CSS placement option you select applies to the custom CSS style sheets as well as the automatically generated ones.

For more information about using cascading style sheets to control the appearance and layout of HMTL and HTML (email) container designs , see *Designing Customer Communications* in the Exstream Design and Production documentation.

Note that if you choose the external CSS placement option for HTML 4.01 Transitional output, the engine does not remove all inline styles from the HTML code, as it does for HTML5 output.

**Note:** If you use HTML output in conjunction with a Dynamic Data Access (DDA) routine, you must include styles within the HTML.

For more information about DDA, see *Configuring Connectors* in the Exstream Design and Production documentation.

To select whether to create an external CSS file or to include style sheet information at the top of the file:

1. In Design Manager, from the Library, drag an HTML output object to the Property Panel.
2. Click the **Basic** tab.

3. Depending on where you want styles to be included in the output, select or clear the **Create external cascading style sheet** check box:

To	Do this
Create an external CSS file with the HTML output	Select the <b>Create external cascading style sheet</b> check box.
Include styles at the top of each HTML file	<p>Clear the <b>Create external cascading style sheet</b> check box.</p> <p><b>Important:</b> If you clear the <b>Create external cascading style sheet</b> check box for HTML 4.01 Transitional output, some styles will be placed at the top of your .html output files and some will remain inline.</p> <p>When you include style information at the top of the HTML file, the engine uses a temporary file to generate HTML 4.01 Transitional or HTML5 output. If you are producing HTML 4.01 Transitional or HTML5 output on a z/OS platform, you must allocate a temporary file (DD : TEMP) before running the engine. The space allocated for the temporary file must be at least as large as the entire HTML file, including style information.</p>

4. From the Menu bar, select **Edit > Save**.

**Note:** When you select **MIME** from the **Version** drop-down list, and you select the **Create external cascading style sheet** check box, the output is still a single MHTML file. However, the single output file includes the CSS as a separate file part with a content type of text/css instead of including it directly within the HTML.

For more information about creating MIME HTML output, see “[Creating MIME HTML Output To Include All Resources in a Single File](#)” on page 95

## 2.9.4 Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output

If you are creating designs for HTML, HTML (email), or Multi-Channel XML outputs, it is important to remember that browsers, email clients, and XML display systems support a limited list of fonts. If your design uses fonts that are not supported by a particular display system, the display system will automatically replace these fonts with supported fonts.

For more precise control over the appearance of your output, it is a best practice to specify HTML fallback fonts that will be substituted for any unsupported design fonts in your output.

Exstream Design and Production allows you to specify a secondary font family for each design font using the **HTML fallback font options** section of the font properties. You can also specify a generic font family to use if both the design font and the secondary font family are not supported.

## Enabling HTML Fallback Font Options

**Important:** You must enable HTML fallback font options individually for each font in your design.

To enable HTML fallback font options for a design font:

1. In Design Manager, in the Library, go to **Environment > Design > Fonts**.
2. Select the font for which you want to specify fallback font options and drag it to the Property Panel.
3. In the **HTML fallback font options** section of the font properties, select the **Use HTML fallback font family** check box.

**Note:** By default, when you import a new font, the **Use HTML fallback font family** check box *is not* selected. This means that Exstream does not specify a fallback font family for that font and the display system selects a substitute font for any unsupported fonts.

However, when you upgrade your design database from a previous version of Exstream, by default the check box *is* selected and the **HTML generic font family** list is set to **Preserve existing** for all existing design fonts to preserve previous functionality.

4. In the **HTML secondary font family** box, enter the name of the font family to use if the design font is not supported.

Note that this box is disabled if **Preserve existing** is selected in the **HTML generic font family** list.

5. In the **HTML generic font family** list, select one of the available options to specify the generic font family to use if both the design font and the secondary font family are not supported:

- Select **Preserve existing** to preserve the appearance of fonts in your output as they appeared in previous versions of Exstream. Keep in mind that selecting this option disables the **HTML secondary family** box; however, if you previously entered a secondary font family name, this information is not discarded.
- Select one of the following options to instruct the display system to substitute any unsupported design font with a suitable font from the selected generic font family.
  - **Sans-serif**
  - **Serif**
  - **Cursive**

- **Fantasy**
- **Monospace**

## 2.9.5 Using Scalable Vector Graphic Format for Graphic Design Objects

When you produce HTML output, the engine creates a separate image file for each graphic design object (such as a chart, line, or polygon) in either GIF or Scalable Vector Graphic (SVG) format. The engine places these image files, along with any external images that you imported at run time, into the same destination directory as your HTML files. Keep in mind that the images that you import into your design are referenced in the HTML output files in their original format (such as JPEG or PNG).

When you produce HTML 4.01 Transitional output, the engine uses GIF format by default for the images created from graphic design objects, but you can optionally use Scalable Vector Graphic (SVG) format for compatibility with older versions of Exstream or based on your business needs. When you produce HTML5 output, you must use SVG format.

**Note:** If you use SVG format for graphic design objects, and you are viewing HTML output in a version of Microsoft Internet Explorer earlier than 9, you must have a browser plug-in that lets you view SVG format; otherwise, the images do not appear as designed. You can download and install Adobe SVG Viewer for free from Adobe. Most web browsers natively support the GIF image format, and the latest versions of most web browsers natively support the SVG image format.

To use SVG format for graphic design objects:

1. In Design Manager, from the Library, drag an HTML output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Use SVG for graphics** check box.

**Note:** When you are producing HTML5 output, you must use SVG format for graphic design objects. If you select HTML version 5 as the output driver, the **Use SVG for graphics** check box is selected by default and you cannot clear it.

4. From the **Object Positioning** drop-down list, specify how objects are positioned in the output:
  - **Absolute positioning with CSS styles**—Use a cascading style sheet to control the position of objects in the output. The engine compiles text styles into an external style sheet named `Styles.css` in the same directory as the output file. Creating a separate

file (an external style sheet) is useful to enforce consistent formatting across several webpages.

**Note:** When you are producing HTML5 output, this option is selected by default and you cannot change it.

- **Table in table positioning**—Use tables to control the position of objects in the output. Positioning with tables is not as precise as positioning with style sheets. This feature is typically used with frames on HTML pages.
5. From the Menu bar, select **Edit > Save**.

## 2.9.6 Adding Hyperlinks at the Top of Each HTML Page

You can add hyperlinks (also called bookmarks) at the top of each HTML page that direct to sections on the same page. An end user will use a single click to follow hyperlink addresses in HTML output.

**Note:** This option is available only when you are producing HTML 4.01 Transitional or MIME HTML output. You cannot include hyperlinks at the top of each page if you are producing HTML5 output.

For more information about creating different versions of HTML output, see “[Creating Basic HTML Output](#)” on page 93.

To add hyperlinks at the top of each page:

1. In Design Manager, from the Library, drag an HTML output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Include page links at the top of each HTML output page** check box.
4. From the Menu bar, select **Edit > Save**.

## 2.9.7 Adding Metadata (<meta> Elements) to HTML and HTML (Email) Output

You can add <meta> elements to HTML and HTML (email) output files by adding metadata to the application object in Design Manager. The name specified for the metadata object is assigned to the name attribute of the <meta> element, and the value of the metadata object is assigned to the content attribute. The value of a metadata object can be static or variable text.

For example, suppose that you have a metadata object named "author" and the value of that metadata object is a name. The resulting HTML might look like the following:

```
<meta name="author" content="Jane Doe">
```

When designing for mobile use, you might use these meta elements to change the appearance of certain elements on the screen, such as disabling the default behavior that telephone numbers appear as links in an email. To do that, you can create a metadata object named "format-detection" with a value of "telephone=no". The resulting HTML would look like the following:

```
<meta name="format-detection" content="telephone=no">
```

For HTML and HTML (email) output, only metadata tags on application objects appear as <meta> elements in the output. However, you can tag objects in a container design with metadata objects to specify class names for elements in the HTML.

For more information about using metadata objects and adding metadata to objects in Design Manager and Designer, or about using metadata to specify a class name for elements in HTML, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To add meta elements to HTML or HTML (email) output:

1. Create a metadata object.
  - a. In Design Manager, in the Library, navigate to **Environment > System > Metadata**.
  - b. Right-click the **Metadata** heading or subheading and select **New Metadata**.
  - c. On the **New Metadata** dialog box, in the **Name** box, enter a name. In the **Description** box, enter a description (optional).

**Tip:** The value that you enter in the **Name** box will be the value of the name attribute for the <meta> tag. For example, if you enter keywords in the **Name** box, the resulting HTML would be: <meta name="keywords">.

- d. Click **Finish**.
2. Apply a value to the metadata object to use at the application level.
  - a. In Design Manager, from the Library, right-click the application object and select **Metadata**.
  - b. In the **Metadata dialog** box, from the **Available Metadata** area, select the metadata object that you want to use to define the attribute <meta> element value.
  - c. Click .

- d. In the **Applied Metadata** area, select the metadata object.
- e. In the **Metadata value** area, specify the value in one of the following ways:

To	Do this
Assign a static text value to the metadata object	<ol style="list-style-type: none"><li>a. In the <b>Default value</b> box, enter the static text value that you want to assign the metadata</li><li>b. Click <b>OK</b>.</li></ol>
Assign a default variable value to the metadata object	<ol style="list-style-type: none"><li>a. Click .</li><li>b. On the <b>Select Variable</b> dialog box, select the default variable that you want to assign to the metadata object.</li><li>c. Click <b>OK</b>.</li></ol>

## 2.9.8 Setting Up to Transform HTML Output

If you are delivering container designs in HTML output, you can transform that output during production by enabling the embedded XSLT engine in the properties of an HTML output driver. If enabled, the XSLT engine automatically runs, using the HTML output file created by the Exstream engine and an XSL file that you provide, and transforms the HTML output created in Exstream for final presentation. You can use an XSL transformation to customize or add content to HTML and HTML (email) output, such as JavaScript or other custom content that you would normally include as a standard part of a webpage. For example, suppose that you want to include functionality from the jQuery JavaScript library in your final HTML, and you want to use jQuery functions to provide dynamic navigation features in a summary table. You can use an XSL transformation to add a reference to the jQuery library and a jQuery function that adds navigation to the table.

For more information about creating container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

**Note:** When you are creating MIME HTML output, you cannot run the XSLT engine to transform the output.

For more information about creating MIME HTML output, see “[Creating MIME HTML Output To Include All Resources in a Single File](#)” on page 95

Before you can successfully transform HTML output during production, you must download the XHTML DTD and character entity files, which are available from the W3C website. You must place these files in the engine installation directory. The files that you must download are listed in the following table:

Default file name	Contents of file
xhtml1-transitional.dtd	Extensible HTML version 1.0 Transitional DTD
xhtml1-lat1.ent	Latin-1 character entity set (referenced by the DTD)
xhtml-special.ent	Special character entity set (referenced by the DTD)
xhtml-symbol.ent	Symbol character entity set (referenced by the DTD)

For more information about the XHTML DTD, see the W3C website.

You can specify an alternative location for these files using the `HTML_DTD_LOCATION` engine switch.

For more information about the `HTML_DTD_LOCATION` engine switch, see *Switch Reference* in the Exstream Design and Production documentation.

To help you target certain elements for transformation in your XSL file, you can use metadata to specify the `class` attribute for objects in HTML output. Additionally, you can create a metadata report to validate that the metadata values or the variables that provide metadata values in the design are consistent with what you expect for classes in HTML output.

For more information about using metadata to specify the `class` attribute in HTML output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about creating metadata reports, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To set up to transform HTML output:

1. In Design Manager, from the Library, drag an HTML output object to the Property Panel.  
The output properties open in the Property Panel for you to define.
2. Click the **Basic** tab.
3. Select the **Use container design** check box.
4. In the **Container design label** box, click  and select the name of the container design that you want to deliver in the output.
5. Select the **Run XSLT engine on output** check box.

6. Specify an XSL file by doing one of the following:

To	Do this
Use a variable to dynamically specify an XSL file for both testing and production	Click the <b>Variable for file naming</b> box and select a variable.
Specify static locations for the test and production XSL files	<ol style="list-style-type: none"><li>In the <b>Test XSL file</b> box, enter the path to the XSL file that you want to use for testing. Alternatively, you can click  and browse to the XSL file that you want to use.<p><b>Tip:</b> If you want to view the XSL file, click . The XSL file opens in Notepad.</p></li><li>In the <b>Production XSL file</b> box, enter the path to the XSL file that you want to use for production.</li></ol>

7. From the Menu bar, select **Edit > Save**.

## Creating Transformed Output on UNIX and Linux

If you are using the embedded XSLT engine with the HTML output driver on UNIX or Linux, you must set up the environment for the platform you are using. The library path environment variable must contain the path to the directory that contains the ExEngTrans library. The actual directory path is dependent on where you have installed Exstream Design and Production.

The following table lists the environment variables that you must set up to use the XSLT engine, depending on your platform:

Environment variables to set up on UNIX/Linux

Platform	Environment variable
HP-UX	SHLIB_PATH
IBM AIX	LIBPATH
SuSE	LD_LIBRARY_PATH
Red Hat	LD_LIBRARY_PATH
Sun Solaris	LD_LIBRARY_PATH

## 2.9.9 Avoiding File Name Conflicts with CSS Files When Using Multiple HTML Output Objects

When you include multiple output queues in your application, and you assign different HTML output objects to those queues, the external CSS files will have the same name. (For example, you might have such a setup if you are producing both HTML 4.01 Transitional output and HTML5 output from the same application.) If each output is created in the same location, the CSS file from each subsequent output will overwrite the CSS files from any previously produced outputs.

You can avoid such file name conflicts either by creating outputs in different locations, or by adding a prefix to the names of CSS files in the settings of each HTML output object.

To add a prefix to the name of the CSS file that is produced with HTML output:

1. In Design Manager, from the Library, drag an HTML output object to the Property Panel.  
For more information about creating an HTML output object, see [“Creating Basic HTML Output” on page 93](#).
2. Click the **Resource Management** tab.
3. In the **CSS file name prefix** box, enter the prefix that you want to add to the beginning of the name of the CSS file that is produced with the HTML output. You can use only alphabetic characters (a-z or A-Z) for the prefix.
4. From the Menu bar, select **Edit > Save**.

**Note:** To avoid file name conflicts with the HTML output files themselves, you must also specify a unique test file name and production file name for each output queue to which an HTML output object is assigned.

## 2.10 Setting Up HTML (Email) Output

When you add an HTML (email) output object or output queue to your application, Exstream produces a set of files in the output directory that you specify in the output object properties. This includes the following files:

- **HTML (email) output file (.html)**—The source HTML file
- **Plain text file (.txt)**—The plain text version of the email output (optional)
- **GIF files (.gif)**—These files contain the graphic design objects (such as charts, lines, and shapes) that you added to your design using the drawing tools in Designer. The engine converts these design objects into separate image files in GIF format, and then references

those image files in the HTML (email) output, along with any images you imported at run time in their original format (such as JPEG or PNG).

**Note:** You can specify how images are referenced and stored in HTML (email) output using the **Image Management** options on the **Resource Management** tab of the output object. For more information about managing image resources in HTML (email) output, see “[Managing Image Resources](#)” on page 235.

In HTML (email) output, all of the text and paragraph styles that you applied in Designer are embedded in the .html output files, either inline with the HTML elements or within the <head> element at the top of the files. Because all of the styles are embedded with the content in the .html files, you do not receive a separate CSS file in the output.

If you apply custom CSS styles to an email container design by linking a cascading style sheet object to the container design label, you can specify whether you want those custom CSS styles to be placed inline at the HTML element level or within the <head> element at the top of the .html output files. You do not have the option to place the custom CSS in a separate, external file because email clients do not support external style sheets.

The following table identifies the ways in which you can customize your HTML output:

For information about	See
Selecting a container design label to use when producing HTML (email) output	<a href="#">“Creating Basic HTML (Email) Output” on the next page</a>
<b>Note:</b> Container designs are required for producing HTML (email) output.	
Selecting an external email template	<a href="#">“Creating HTML (Email) Output With an External Email Template” on page 110</a>
Formatting the HTML (email) output using cascading style sheets	<ul style="list-style-type: none"><li>• <b>Using cascading style sheets in Exstream</b>—<i>Designing Customer Communications</i> in the Exstream Design and Production documentation</li><li>• <b>Specifying whether styles are included inline or at the top of the file</b>—<a href="#">“Specifying Where Styles Are Included in HTML (Email) Output” on page 108</a></li><li>• <b>Adding custom id attributes to objects</b>—<i>Designing Customer Communications</i> in the Exstream Design and Production documentation</li></ul>
Importing an external HTML email template at run time	<a href="#">“Creating HTML (Email) Output With an External Email Template” on page 110</a>
Selecting which fonts to display if the fonts used in the design are not supported by the browser or email client	<a href="#">“Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output” on page 114</a>
Applying a background color to the HTML (email) output	<a href="#">Creating Output</a> in the Exstream Design and Production documentation

For information about	See
Adding metadata (a <code>meta</code> element) to the <code>&lt;head&gt;</code> section of the HTML (email) file	<a href="#">"Adding Metadata (<code>&lt;meta&gt;</code> Elements) to HTML and HTML (Email) Output" on page 115</a>
Using an XSL file to perform post-processing tasks to transform the HTML (email) output for final presentation	<a href="#">"Setting Up to Transform HTML Output" on page 117</a>
Using a DDA output file to perform post-processing tasks to manipulate the output before it reaches the print stream	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>

#### Related information:

*Designing Customer Communications* in the Exstream Design and Production documentation

### 2.10.1 Creating Basic HTML (Email) Output

When you create basic HTML (email) output, you must specify the container designs that you want to deliver in the output. Optionally, you can also create a plain text version of the output.

To create basic HTML (email) output:

1. If you do not know the application mode of the database in which you are working, review the **Application mode** drop-down list on the **Workflow** tab of the **System Settings** object.  
For information about the application mode of a database, see *System Administration* in the Exstream Design and Production documentation.
2. Based on the application mode of the database in which you are working, do one of the following:

For this application mode	Do this
SBCS	<ol style="list-style-type: none"><li>In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs</b>.</li><li>Right-click the <b>Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li></ol>
DBCS	<ol style="list-style-type: none"><li>In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs</b>.</li><li>Right-click the <b>Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li></ol>

For this application mode	Do this
SBCS/DBCS	<ul style="list-style-type: none"> <li>• To create SBCS output, complete the following steps:           <ul style="list-style-type: none"> <li>i. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs &gt; SBCS Outputs</b>.</li> <li>ii. Right-click the <b>SBCS Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li> </ul> </li> <li>• To create DBCS output, complete the following steps:           <ul style="list-style-type: none"> <li>i. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs &gt; DBCS Outputs</b>.</li> <li>ii. Right-click the <b>DBCS Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li> </ul> </li> </ul>

3. In the **Name** box, enter a name for the output device.

4. In the **Description** box, enter a description (optional).

5. Click **Finish**.

The output object opens in the Property Panel for you to define.

6. From the **Driver** drop-down list, select **HTML (email)**.

7. To include a plain text version of the email output, select the **Include plain text** check box.

8. In the **Container design label** box, click  and select the name of the container design that you want to deliver in the output.

For more information about container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

9. From the **HTML email alignment** drop-down list, select the horizontal alignment that you want to apply to the primary container of the HTML email output. Keep in mind that this setting does not affect the alignment of the content that is inside the primary container.

10. From the Menu bar, select **Edit > Save**.

## 2.10.2 Specifying Where Styles Are Included in HTML (Email) Output

When you use a cascading style sheet (CSS) object to control the final appearance of HTML (email) output, you can instruct the engine to place your custom CSS code in one of the following two locations:

- Inline at the HTML element level within each .html output file
- Within the <head> element at the top of each .html output file

When you are deciding which CSS placement option to choose, you should consider several factors. Storing all styles at the top of each .html file makes it easier to modify the appearance of an email message outside of Exstream than if you embed the styles inline in the HTML code, because all of the styles are contained in one location in each file instead of spread throughout the HTML code. However, you must also keep in mind that some email clients support inline CSS only, and will ignore any CSS code placed in the <head> section of an .html file. If you are not sure which email clients your customers are using and you want to ensure compatibility with the largest number of email clients, you should select the inline CSS placement option. Additionally, if your design uses a user-defined CSS to specify font information for HTML (email) output, you must use the inline option for CSS placement to ensure that your user-defined styles appear correctly in the output.

You can also choose to exclude from your HTML (email) output any custom CSS styles that are linked to your application. Note that this option is primarily for backwards compatibility.

Keep in mind that the CSS placement options for HTML (email) output affect the placement of only the custom CSS styles that you apply to an email design by linking a CSS object to the container design label. They do not affect the placement of the CSS styles that the Exstream engine automatically generates to define how your content should be displayed based on the layout, styles, and formatting properties that you applied while working in Designer.

To specify where custom CSS styles are placed in HTML (email) output, you must first create a custom CSS object and link it to the container design label, which in turn is linked to the output object for your design. For more information about using cascading style sheets to control the appearance and layout of HTML and HTML (email) container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To specify where custom CSS styles are placed relative to the HTML code in HTML (email) output:

1. In Design Manager, from the Library, drag a cascading style sheet object to the Property Panel.
2. Click the **Basic** tab.
3. From the **CSS placement in HTML email output** drop-down list, select one of the following options:

To	Do this
Place CSS styles inline in HTML (email) output	Select <b>Inline</b> .
Place CSS styles in the <head> element at the top of the HTML (email) output files	Select <b>Internal (Top of file)</b> .

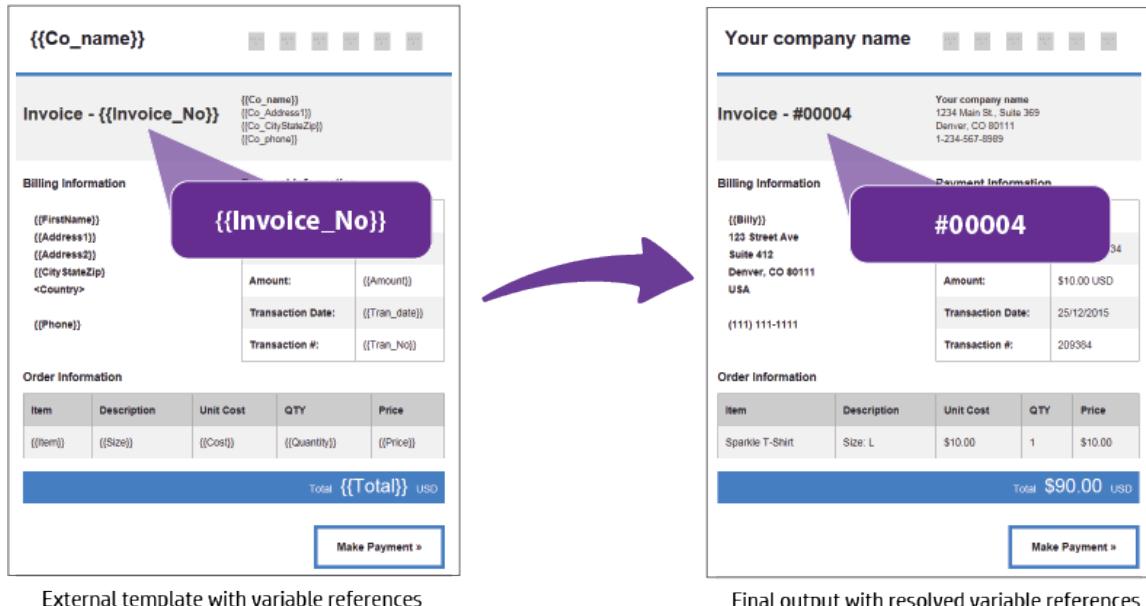
To	Do this
Exclude CSS styles from HTML (email) output	Select <b>None</b> .

4. From the Menu bar, select **Edit > Save**.

### 2.10.3 Creating HTML (Email) Output With an External Email Template

In addition to creating HTML (email) output from an Exstream container design, you also have the option to create HTML (email) output using an external HTML email template that you import into Exstream at run time. You might want to use an external email template if you want to use responsive email features that are not available in Exstream. With this method, you can use a third-party HTML editor to create the email content, and then leverage the power of Exstream to personalize and deliver the communication to your target customers.

To use Exstream to personalize an externally created email design, you must embed variable references in your design where you want the Exstream engine to insert variable data. When you import the design at run time, the engine will replace the variable references with the appropriate mapped variable data, as shown in the following illustration:



Note that Exstream does not change the static content or the HTML markup in your external HTML email template file in any way when it generates the output; it only replaces the variable references in the email design with the appropriate customer-specific data.

To create HTML (email) output with an external email template, you must complete the following tasks:

- “[Embedding Exstream Variables in an External HTML Email Template](#)” below
- “[Setting Up an Exstream Application to Personalize and Deliver an Externally Created Email Design](#)” below

## Embedding Exstream Variables in an External HTML Email Template

To enable the Exstream engine to personalize an externally created HTML email design, you must insert variables that you have defined in Exstream into your email design wherever you want the engine to insert customer-specific data. When you are embedding Exstream variables in your external HTML email template, keep in mind the following guidelines to ensure that the Exstream engine can properly resolve the variables when it creates your final output:

- Insert defined variable names enclosed in angle brackets inside the email content where you want variable data to appear. Since HTML uses angle brackets to identify tags, you must use the HTML entities &lt; and &gt; to represent the angle brackets. You can also use double braces. For example, an embedded variable for the customer name could look like &lt;CustomerName&gt; or {{CustomerName}}.

**Note:** Do not use the character codes &#60; and &#62; to represent angle brackets; these codes are not supported.

- Any variable used in the external file must also be used somewhere in the application or package file in order for the variable to be recognized by the Exstream engine.
- Variable names are case-sensitive and must match the variable name spelling exactly as it appears in the Exstream Library. If the engine identifies an entry as a variable, but is unable to match the variable name to a variable used in the application or package file, the entry in the external content will be ignored and left unchanged in the final output. In the engine message file, the engine issues an error message that states the name of the imported variable that is not referenced in the application or package file.

For example, suppose the variable name used in the application is 'Customer\_Name', but the imported HTML contains Dear {{CustomerName}} and the variable 'CustomerName' does not exist in the application. Instead of the output including something like Dear Susan, it will contain Dear{{CustomerName}}.

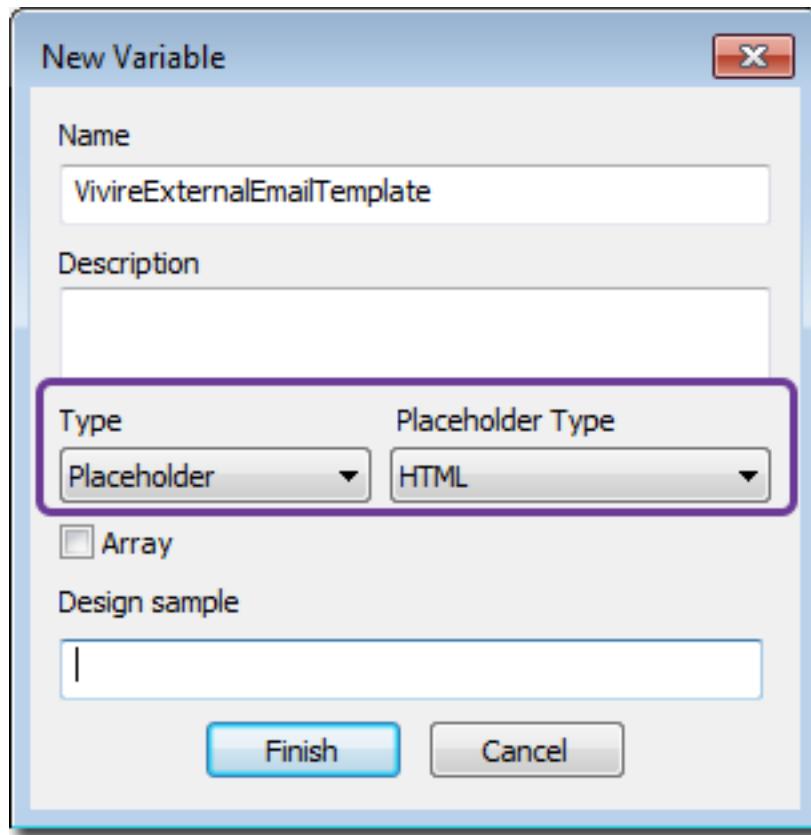
- You can embed any type of variable in your external email template *except* Formatted Text, Placeholder, or Tagged Text variables.

## Setting Up an Exstream Application to Personalize and Deliver an Externally Created Email Design

After you have created your external email template, you must complete the following tasks in Exstream before you can generate output using the external template:

1. Create a placeholder variable that specifies the external HTML email template file that you want to import at run time.

In the **New Variable** dialog box, be sure to select **Placeholder** from the **Type** drop-down list and **HTML** from the **Placeholder Type** drop-down list.



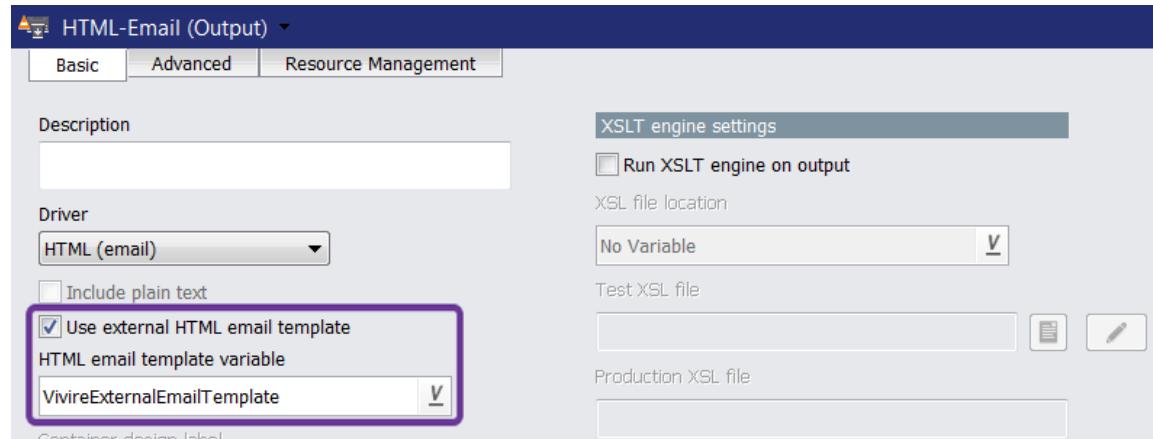
Set the value of the variable to the external email template file name and path (for example, C:\Templates\VivireEmailTemplate1.html). You can use any of the available methods for specifying the variable value (for example, a user specified value, a mapped data file, or a formula). For information on setting up variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

**Note:** Alternately, you can set the value of the variable to the template file name (for example, VivireEmailTemplate1.html) and use the IMPORTDIRECTORY switch in a control file to specify the file location. For information on the IMPORTDIRECTORY switch, see *Switch Reference* in the Exstream Design and Production documentation.

2. Create an HTML (email) output object that instructs the engine to use the external email template when creating the output.

On the **Basic** tab of the HTML (email) object properties, select the **Use external**

**HTML email template** check box and, in the **HTML email template variable** box, specify the name of the placeholder variable that you created in step 1.



3. Set up the application object with the required structure and data to build the package file and generate the output.

The application object must contain references to the following:

- An output queue object that references the HTML (email) output object you created in step 2
- The data files containing the data that is mapped to the variables embedded in your external email template
- A document that contains a page object

**Note:** Although the content for your email message will be contained in your external template, not in an Exstream page design, you must include a page object in your application in order to run the engine and generate the output. The page can be blank, or it can contain content. If the page that you use contains content, the engine ignores the content in the Exstream page design and includes only the content in the external email template in the HTML (email) output. (Conversely, if you include additional output queues for different types of output in the same application, the content in the page design, and not the content in the external email template, will be included in non-email output).

After you have completed these tasks, you are ready to package the application and run the engine to create the output.

## 2.10.4 Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output

If you are creating designs for HTML, HTML (email), or Multi-Channel XML outputs, it is important to remember that browsers, email clients, and XML display systems support a limited list of fonts. If your design uses fonts that are not supported by a particular display system, the display system will automatically replace these fonts with supported fonts.

For more precise control over the appearance of your output, it is a best practice to specify HTML fallback fonts that will be substituted for any unsupported design fonts in your output.

Exstream Design and Production allows you to specify a secondary font family for each design font using the **HTML fallback font options** section of the font properties. You can also specify a generic font family to use if both the design font and the secondary font family are not supported.

### Enabling HTML Fallback Font Options

**Important:** You must enable HTML fallback font options individually for each font in your design.

To enable HTML fallback font options for a design font:

1. In Design Manager, in the Library, go to **Environment > Design > Fonts**.
2. Select the font for which you want to specify fallback font options and drag it to the Property Panel.
3. In the **HTML fallback font options** section of the font properties, select the **Use HTML fallback font family** check box.

**Note:** By default, when you import a new font, the **Use HTML fallback font family** check box *is not* selected. This means that Exstream does not specify a fallback font family for that font and the display system selects a substitute font for any unsupported fonts.

However, when you upgrade your design database from a previous version of Exstream, by default the check box *is* selected and the **HTML generic font family** list is set to **Preserve existing** for all existing design fonts to preserve previous functionality.

4. In the **HTML secondary font family** box, enter the name of the font family to use if the design font is not supported.

Note that this box is disabled if **Preserve existing** is selected in the **HTML generic font family** list.

5. In the **HTML generic font family** list, select one of the available options to specify the generic font family to use if both the design font and the secondary font family are not supported:
  - Select **Preserve existing** to preserve the appearance of fonts in your output as they appeared in previous versions of Exstream. Keep in mind that selecting this option disables the **HTML secondary family** box; however, if you previously entered a secondary font family name, this information is not discarded.
  - Select one of the following options to instruct the display system to substitute any unsupported design font with a suitable font from the selected generic font family.
    - **Sans-serif**
    - **Serif**
    - **Cursive**
    - **Fantasy**
    - **Monospace**

## 2.10.5 Adding Metadata (<meta> Elements) to HTML and HTML (Email) Output

You can add <meta> elements to HTML and HTML (email) output files by adding metadata to the application object in Design Manager. The name specified for the metadata object is assigned to the name attribute of the <meta> element, and the value of the metadata object is assigned to the content attribute. The value of a metadata object can be static or variable text.

For example, suppose that you have a metadata object named "author" and the value of that metadata object is a name. The resulting HTML might look like the following:

```
<meta name="author" content="Jane Doe">
```

When designing for mobile use, you might use these meta elements to change the appearance of certain elements on the screen, such as disabling the default behavior that telephone numbers appear as links in an email. To do that, you can create a metadata object named "format-detection" with a value of "telephone=no". The resulting HTML would look like the following:

```
<meta name="format-detection" content="telephone=no">
```

For HTML and HTML (email) output, only metadata tags on application objects appear as <meta> elements in the output. However, you can tag objects in a container design with metadata objects to specify class names for elements in the HTML.

For more information about using metadata objects and adding metadata to objects in Design Manager and Designer, or about using metadata to specify a class name for elements in HTML, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To add meta elements to HTML or HTML (email) output:

1. Create a metadata object.
  - a. In Design Manager, in the Library, navigate to **Environment > System > Metadata**.
  - b. Right-click the **Metadata** heading or subheading and select **New Metadata**.
  - c. On the **New Metadata** dialog box, in the **Name** box, enter a name. In the **Description** box, enter a description (optional).

**Tip:** The value that you enter in the **Name** box will be the value of the name attribute for the <meta> tag. For example, if you enter keywords in the **Name** box, the resulting HTML would be: <meta name="keywords">.

- d. Click **Finish**.
2. Apply a value to the metadata object to use at the application level.
  - a. In Design Manager, from the Library, right-click the application object and select **Metadata**.
  - b. In the **Metadata dialog** box, from the **Available Metadata** area, select the metadata object that you want to use to define the attribute <meta> element value.
  - c. Click .
  - d. In the **Applied Metadata** area, select the metadata object.
  - e. In the **Metadata value** area, specify the value in one of the following ways:

To	Do this
Assign a static text value to the metadata object	<ol style="list-style-type: none"><li>a. In the <b>Default value</b> box, enter the static text value that you want to assign the metadata</li><li>b. Click <b>OK</b>.</li></ol>
Assign a default variable value to the metadata object	<ol style="list-style-type: none"><li>a. Click .</li><li>b. On the <b>Select Variable</b> dialog box, select the default variable that you want to assign to the metadata object.</li><li>c. Click <b>OK</b>.</li></ol>

## 2.10.6 Setting Up to Transform HTML Output

If you are delivering container designs in HTML output, you can transform that output during production by enabling the embedded XSLT engine in the properties of an HTML output driver. If enabled, the XSLT engine automatically runs, using the HTML output file created by the Exstream engine and an XSL file that you provide, and transforms the HTML output created in Exstream for final presentation. You can use an XSL transformation to customize or add content to HTML and HTML (email) output, such as JavaScript or other custom content that you would normally include as a standard part of a webpage. For example, suppose that you want to include functionality from the jQuery JavaScript library in your final HTML, and you want to use jQuery functions to provide dynamic navigation features in a summary table. You can use an XSL transformation to add a reference to the jQuery library and a jQuery function that adds navigation to the table.

For more information about creating container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

**Note:** When you are creating MIME HTML output, you cannot run the XSLT engine to transform the output.

For more information about creating MIME HTML output, see “[Creating MIME HTML Output To Include All Resources in a Single File](#)” on page 95

Before you can successfully transform HTML output during production, you must download the XHTML DTD and character entity files, which are available from the W3C website. You must place these files in the engine installation directory. The files that you must download are listed in the following table:

Default file name	Contents of file
xhtml1-transitional.dtd	Extensible HTML version 1.0 Transitional DTD
xhtml1-lat1.ent	Latin-1 character entity set (referenced by the DTD)
xhtml-special.ent	Special character entity set (referenced by the DTD)
xhtml-symbol.ent	Symbol character entity set (referenced by the DTD)

For more information about the XHTML DTD, see the W3C website.

You can specify an alternative location for these files using the `HTML_DTD_LOCATION` engine switch.

For more information about the `HTML_DTD_LOCATION` engine switch, see *Switch Reference* in the Exstream Design and Production documentation.

To help you target certain elements for transformation in your XSL file, you can use metadata to specify the `class` attribute for objects in HTML output. Additionally, you can create a metadata

report to validate that the metadata values or the variables that provide metadata values in the design are consistent with what you expect for classes in HTML output.

For more information about using metadata to specify the `class` attribute in HTML output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about creating metadata reports, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To set up to transform HTML output:

1. In Design Manager, from the Library, drag an HTML output object to the Property Panel.  
The output properties open in the Property Panel for you to define.
2. Click the **Basic** tab.
3. Select the **Use container design** check box.
4. In the **Container design label** box, click  and select the name of the container design that you want to deliver in the output.
5. Select the **Run XSLT engine on output** check box.
6. Specify an XSL file by doing one of the following:

To	Do this
Use a variable to dynamically specify an XSL file for both testing and production	Click the <b>Variable for file naming</b> box and select a variable.
Specify static locations for the test and production XSL files	<ol style="list-style-type: none"><li>a. In the <b>Test XSL file</b> box, enter the path to the XSL file that you want to use for testing. Alternatively, you can click  and browse to the XSL file that you want to use.<div style="border: 1px solid #ccc; padding: 10px; margin-top: 10px;"><p><b>Tip:</b> If you want to view the XSL file, click . The XSL file opens in Notepad.</p></div></li><li>b. In the <b>Production XSL file</b> box, enter the path to the XSL file that you want to use for production.</li></ol>

7. From the Menu bar, select **Edit > Save**.

## Creating Transformed Output on UNIX and Linux

If you are using the embedded XSLT engine with the HTML output driver on UNIX or Linux, you must set up the environment for the platform you are using. The library path environment variable must contain the path to the directory that contains the ExEngTrans library. The actual directory path is dependent on where you have installed Exstream Design and Production.

The following table lists the environment variables that you must set up to use the XSLT engine, depending on your platform:

Environment variables to set up on UNIX/Linux

Platform	Environment variable
HP-UX	SHLIB_PATH
IBM AIX	LIBPATH
SuSE	LD_LIBRARY_PATH
Red Hat	LD_LIBRARY_PATH
Sun Solaris	LD_LIBRARY_PATH

## 2.11 Setting Up IJPDS Output

This section discusses tasks you can complete for IJPDS output in Exstream. Since each IJPDS printer can vary, you should specify configurations for each printer in a separate IJPDS object in the Library.

This section discusses the following topics:

- [“Creating an IJPDS Index” on the next page](#)
- [“Specifying a Linearization Table for Tone Correction” on the next page](#)
- [“Setting Up an IJPDS Output Configuration” on page 121](#)

In addition to the topics discussed in this section, you can also use the following basic setup options for IJPDS output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>

Setup option	See this section for more information
Highlighting added text or images in your output with a black underline	" <a href="#">Highlighting New Text and Images in Output</a> " on page 49

## 2.11.1 Creating an IJPDS Index

An IJPDS index is a separate binary file that describes the content in the print file. Some printers require an IJPDS index for large print jobs. You might want to create an IJPDS index if your print file is larger than 2GB.

To create an IJPDS index:

1. In Design Manager, from the Library, drag an IJPDS output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Create IJPDS index** check box.
4. Save the output object.

## 2.11.2 Specifying a Linearization Table for Tone Correction

For IJPDS output, you can provide linearization tables (also known as lookup tables or LUTs) for tone correction. Although linearization tables enhance color output, they also slow processing time since the engine must process all the images, graphics, and text to correct tone settings.

You can specify linearization tables for color planes and image types, based on the paper types and inks in use. You can also specify different linearization tables in all four color planes (C,M,Y,K) for all three bitmap types (fonts, graphics, and images).

**Tip:** To increase print efficiency, use font colors that do not require bitmaps in each color plane. Fonts are drawn once for each print head.

During a production run, the intensity value of each pixel in the .lut file acts as a key into the linearization table to find a corresponding adjusted pixel-intensity value.

To specify a linearization table:

1. In Design Manager, from the Library, drag an IJPDS output object to the Property Panel.
2. Click the **Configuration** tab.
3. In the **Linearization Tables** area, from the **Color** drop-down list, select a color plane to use with a bitmap type:

- **Cyan**
  - **Magenta**
  - **Yellow**
  - **Black**
4. From the **Type** drop-down list, specify the bitmap type to use with the specified color.  
Options are:
- **Text**
  - **Graphics**
  - **Images**
5. Next to the **File** box, click .
- The **Open** dialog box opens.
6. Browse to the .lut file you want to use and click **OK**.
7. Save the output object properties.

### 2.11.3 Setting Up an IJPDS Output Configuration

You specify an IJPDS output configuration based on the number of raster image processors (RIPs) and print heads in use. Since each IJPDS printer can vary, you should specify configurations for each printer in a separate IJPDS object in the Library.

For more information on RIPs and print head components, consult the HP Scitex Corporation documentation.

To configure an IJPDS output object, you must complete the following tasks:

1. [“Configuring Logical RIPs” below](#)
2. [“Configuring Print Heads” on the next page](#)

You can also complete the following optional task as needed:

- [“Specifying the Multiple-Up Settings for IJPDS Output” on page 123](#)

#### Configuring Logical RIPs

A logical RIP translates computer-based instructions into a page description language that the IJPDS system can understand. You cannot use more than 16 logical RIPs with an output object.

To configure logical RIPs:

1. In Design Manager, from the Library, drag an IJPDS object to the Property Panel.
2. Click the **Advanced** tab.
3. In the **Print heads** area, under **Logical RIPs**, do the following to add or remove logical RIPs:

To	Do this
To add a logical RIP	Click  A logical RIP is added to the <b>Print heads</b> box.
To remove a logical RIP	a. Select the logical RIP you want to remove. b. Click .

4. Save the output object.

## Configuring Print Heads

You can configure up to four print heads for each logical RIP, up to 16 total. You cannot use more than 16 total print heads with one output object.

To configure print heads:

1. In Design Manager, from the Library, drag an IJPDS output object to the Property Panel.
2. Click the **Configurations** tab.
3. In the **Print heads** area, do the following to add or remove a print head:

To	Do this
Add a print head	<ol style="list-style-type: none"><li>a. Select the logical RIP to which you want to add a print head.</li><li>b. Under <b>Print</b>, click .</li></ol> <p>A print head is added to the <b>Print heads</b> box below the logical RIP you selected.</p>
Remove a logical RIP	<ol style="list-style-type: none"><li>a. Select the print head you want to remove.</li><li>b. Under <b>Print</b>, click .</li></ol> <p>If a logical RIP has only one print head, removing the logical RIP removes the print head. If a logical RIP has only one print head, you cannot remove the print head.</p>

4. In the **Print heads** area, select a print head.
5. Click .

The **Print Head Properties** dialog box opens.

6. To identify the print head number and the logical RIP you are defining, see the inactive **Print head** and **Logical RIP** boxes.
7. In the **X offset** box, specify an offset for the print head. When you specify an offset, all IJPDS output for that print head is repositioned using the offset. For example, if you have a 4-inch print head with a 4-inch offset, anything located at the 4-inch mark on the page is repositioned to zero. Keep in mind that output with offsets appear differently in the HP Scitex proofer.

**Caution:** If objects are positioned outside printable area boundaries or if your settings do not match a recognized HP Scitex printer, make sure the print heads are set at the correct position on the page.

8. From the **Width** drop-down list, select the approximate width of the print head in inches.
9. From the **Duplex side** drop-down list, identify the side for the duplex printing.
  - **Default**—If you select **Default** as the **Duplex side** option, Exstream analyzes your setup and determines how the print head is configured. This option is less efficient than the other options.
  - **Front**
  - **Back**
  - **M-up front**
  - **M-up back**If you select either **M-up front** or **M-up back**, the **M-up method** drop-down list appears on the **Configuration** tab.
10. From the **Ink color** drop-down list, identify the color of the ink used with the print head.
11. To identify the number of drops or dots per pixel, select either of the following radio buttons:
  - **1 Drops/Dot**
  - **2 Drops/Dot**
12. Click **OK**.
13. Save the output object.

## Specifying the Multiple-Up Settings for IJPDS Output

For some IJPDS printers, you must use specific multiple-up settings to create multiple-up output. These printers require multiple print heads. You must select one of the multiple-up

options from the **Duplex side** drop-down list on the **Print Head Properties** dialog box to specify multiple-up settings.

For more information the **Duplex side** drop-down list, see “[Configuring Print Heads](#)” on [page 122](#).

For more information about multiple-up settings on an output queue, see “[Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects](#)” on [page 300](#).

To specify the multiple-up settings for IJPDS output:

1. In Design Manager, from the Library, drag an IJPDS output object to the Property Panel.
2. Click the **Configurations** tab.
3. From the **M-up method** drop-down list, select one of the following options:
  - **Each customer document starts sheet-east/west**—Each new customer forces a new sheet of paper to start. Pages alternate between the left and right frames.
  - **Combine customers—east/west**—This option positions customers in the output so that they alternate between the left and right frames.
  - **Combine customers—north/south**—This option positions customers in the output so that they are printed from top to bottom in the first column of frames, then top to bottom in the second column of frames.
4. Save the output object.

## 2.12 Setting Up Metacode Output

This section discusses the following topics:

- “[Setting Up for Metacode Output in the System Settings](#)” on the next page
- “[Printing Metacode Output Offline \(to a Tape Device\)](#)” on [page 126](#)
- “[Using Image Blocking for Metacode Output When Not Printing Offline](#)” on [page 127](#)
- “[Setting Print Stream Options for Metacode Output](#)” on [page 127](#)

In addition to the topics discussed in this section, you can also use the following basic setup options for Metacode output:

Setup option	See this section for more information
Creating an output object	“ <a href="#">Creating an Output Object</a> ” on <a href="#">page 34</a>
Specifying the resolution for your output	“ <a href="#">Managing output resolution in Exstream</a> ” on <a href="#">page 37</a>

Setup option	See this section for more information
Setting up simplex and duplex options	<a href="#">"Using Simplex and Duplex in Output" on page 39</a>
Specifying how to use color	<a href="#">"Specifying the Way Color is Produced in Output" on page 40</a>
Creating each page as an image	<a href="#">"Creating Pages as Images" on page 44</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">"Using Search Keys in Output" on page 46</a>
Specifying how to set up unprintable areas in the page margins	<a href="#">"Specifying Unprintable Areas for Page Margins" on page 48</a>
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>
Setting up paper bin options	<a href="#">"Setting Up Paper Bins" on page 50</a>

## 2.12.1 Setting Up for Metacode Output in the System Settings

Before you create Metacode output, the system administrator must specify the following options in the System Settings on the **Production Print** tab:

1. In Design Manager, in the Library, navigate to **Environment > System Settings**.
2. Drag the **System Settings** to the Property Panel.
3. Click the **Production Print** tab.

4. Set up the following options:

To	Do this
Specify a character prefix for images in Metacode output	In the <b>Resource naming convention</b> area, in the <b>Image</b> box, enter the character prefix you want to use. You can use either one or two characters.
Specify a character prefix for fonts in Metacode output	In the <b>Resource naming convention</b> area, in the <b>Font</b> box, enter the character prefix you want to use. You can use either one or two characters.
Specify the IDEN command for Metacode. The IDEN command is a character sequence that identifies and invokes DJDE records for the print stream.	<ol style="list-style-type: none"><li>In the <b>Metacode IDEN</b> area, in the <b>String</b> box, identify the IDEN string (for example, \$\$DJDE) to include in all Metacode outputs involving the current database.</li><li>For online (to printer) production runs, do the following:<ul style="list-style-type: none"><li>In the <b>Skip</b> box, specify the number of characters to skip before reading actual data.</li><li>In the <b>Offset</b> box, specify the starting position of the IDEN string, based on zero in the first position.</li></ul></li><li>Select the <b>ASCII</b> check box to set the character set for the <b>String</b>, <b>Skip</b>, and <b>Offset</b> options to ASCII. To set the character set to EBCDIC, clear the check box. Match this to your JSL settings.</li></ol>

**Note:** As you save or exit the **System Settings**, if the system issues a warning message that the IDEN information is incorrect, verify that there are no spaces in the string.

5. Save the **System Settings**.

For more information about specifying resources for Metacode output, see “[Managing Resources for the Print Stream](#)” on page 229.

## 2.12.2 Printing Metacode Output Offline (to a Tape Device)

By default, Exstream produces Metacode output for inline printers. You can compose Metacode output to a tape device using the offline feature. When you use the offline feature, Exstream automatically increases the skip and offset in the JSL by one.

To compose Metacode offline:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.

3. Select the **Offline** check box.
4. Save the output object.

### 2.12.3 Using Image Blocking for Metacode Output When Not Printing Offline

If you are not printing Metacode offline, you can specify that the Exstream engine write resource records without the CC or endrec characters, as it does for Shared Disk Interface (SDI) or offline output.

To structure Metacode output as if it were offline:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Force image blocking** check box. The option appears only if the **Offline** check box is cleared.
4. Save the output object.

### 2.12.4 Setting Print Stream Options for Metacode Output

To customize the print stream for Metacode output, complete the following tasks as needed:

- [“Specifying the Record Structure for the Print Stream” below](#)
- [“Specifying the JDL and JDE to Use for Output” on page 129](#)
- [“Specifying the Position Coordinates to Begin Printing” on page 129](#)
- [“Adding a Form Feed Carriage Control After the DJDE Header” on page 130](#)
- [“Listing Fonts and Resources at the Top of Each Page File” on page 130](#)
- [“Specifying the Type of Carriage Controls” on page 130](#)

#### Specifying the Record Structure for the Print Stream

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Print stream blocking method** drop-down list, identify the record structure. To support the processing of records, change the default (**None**) to one of the following

settings:

Setting	Description
CRLF	Records are separated by 0D0A (carriage return/line feed).
<b>2 byte, big endian, inclusive</b>	The record length indicator is 2 bytes long, with the most significant bytes first. The record length includes the record length itself.
<b>2 byte, big endian, exclusive</b>	The record length indicator is 2 bytes long, with the most significant bytes first. The record length does not include the record length itself.
<b>2 byte, little endian, inclusive</b>	The record length indicator is 2 bytes long, with the least significant bytes first. The record length includes the record length itself.
<b>2 byte, little endian, exclusive</b>	The record length indicator is 2 bytes long, with the least significant bytes first. The record length does not include the record length itself.
<b>4 byte, big endian, inclusive</b>	The record length indicator is 4 bytes long, with the most significant bytes first. The record length includes the record length itself.
<b>4 byte, big endian, exclusive</b>	The record length indicator is 4 bytes long, with the most significant bytes first. The record length does not include the record length itself.
<b>4 byte, little endian, inclusive</b>	The record length indicator is 4 bytes long, with the least significant bytes first. The record length includes the record length itself.
<b>4 byte, little endian, exclusive</b>	The record length indicator is 4 bytes long, with the least significant bytes first. The record length does not include the record length itself.
<b>MVS FTP</b>	The record length indicator starts with 80, followed by a 2-byte length count (exclusive). The output can be successfully uploaded to mainframe without unblocking, or downloaded to a PC with no blocking required.
<b>RDW</b>	Each record is preceded by a 4-byte Record Descriptor Word (RDW), with the most significant bytes first. The record length includes the record length itself.
<b>Fixed length records</b>	Records are padded as necessary to force them to an equal size. In the adjacent box, enter the record size. The maximum size for a record is 1024.
<b>Variable block record (VBR)</b>	A two-byte block length is followed by a 2-byte pad, with the most significant bytes first. The record length includes the record length itself.
<b>Variable length record (VLR)</b>	The record length indicator is 4 bytes long, with the most significant bytes first. The record length includes the record length itself, and it also includes 2-byte trailing pad.
<b>Barr short record format (255)</b>	The spooling format is a fixed record length (255 bytes) with a 4-byte header, the least significant bytes first, a 1-byte start length that does not include the byte, and a 1-byte end length that does not include the byte.
<b>Barr long record format (64k)</b>	The spooling format is a fixed record length (64K bytes) with a 2-byte start length, the least significant bytes first, and a 2-byte end length that does not include the bytes.

Setting	Description
<b>Xerox shared disk (SDI)</b>	The offline Shared Disk Interface (SDI) contains a block length of 512 bytes represented by 4 bytes, with the most significant bytes first. The record length includes the block length, with a 2-byte record length, and does not include the record length itself.
<b>Offline variable block</b>	Use this option in offline mode. This option is similar to the Xerox SDI format, but with longer block lengths (SDI uses a block length of 512 bytes, whereas this option uses any multiple of 512 from 512B to 32768B). The remaining space in a block is padded with null characters (hex 00).
<b>Offline fixed block</b>	Use this option in offline mode. This option is identical to <b>Offline variable block</b> option, except that it does not have the 4 bytes for the block size indicator or null bytes in front of the block. Therefore, the block length must be known by the JSL reading the Metacode. Enter the record size in the adjacent box.

## Specifying the JDL and JDE to Use for Output

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **JDL** box, enter the Job Description Language (JDL) you want to use. The default is EXSJDL.
4. In the **JDE** box, enter the Job Descriptor Entry (JDE) you want to use. The default is EXSJDE.
5. Save the output object.

## Specifying the Position Coordinates to Begin Printing

The x and y coordinates you specify set the position on the page to begin printing. Refer to the procedures for your organization for the settings for these options.

To specify the printing position coordinates:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **BEGIN Y (scan lines)** box, specify the beginning vertical position determining entity (PDE) line.
4. In the **BEGIN X (dots)**, specify the beginning horizontal PDE.
5. Save the output object.

## Adding a Form Feed Carriage Control After the DJDE Header

Depending on your output device, you might need to add a form feed carriage control after the DJDE header. Refer to your output device documentation for more information.

To add a form feed carriage control:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Add form feed after DJDE header** check box.
4. Save the output object.

## Listing Fonts and Resources at the Top of Each Page File

If you want to create self-contained Metacode pages, you can specify that Exstream add a list of fonts and resources at the top of each page file.

To list fonts and resources at the top of each page file:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Add font list to each page** check box.
4. Save the output object.

## Specifying the Type of Carriage Controls

By default, Exstream uses machine carriage controls in the print stream. You can change the carriage character control mode to either ANSI (ASCII) or ANSI (EBCDIC) for all production runs. The important difference between ANSI and machine carriage control is that an ANSI carriage control is executed before the line of data is written. A machine carriage control is executed after the data is written.

To specify the carriage control mode:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Carriage controls** drop-down list, select one of the following options:
  - **ANSI (ascii)**
  - **ANSI (ebcdic)**

- **Machine**
  4. Save the output object.

## 2.13 Setting Up MIBF Output

This section discusses the following topics:

- “[Specifying the Output Format Mode for MIBF Output](#)” below
- “[Specifying ICC Profiles](#)” on the next page
- “[Setting Up an MIBF Printer Configuration](#)” on page 133

In addition to the topics discussed in this section, you can also use the following basic setup options for MIBF output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>

### 2.13.1 Specifying the Output Format Mode for MIBF Output

MIBF output requires multiple output files based on the number of bitmap image processors (BIPs) specified in the printer setting. You can choose to combine the font and page data into one file or to include the files separately.

If you choose to include font and page data in one file, the file includes the following:

- One index file. The name is based on a specified file name in the output queue.
- One data file per BIP. The file name is based on the specified base file name and the extension .mbd## where ## is the BIP number.

If you choose to include font and page data in separate files, the following files are added to the print stream:

- One index file. The name is based on the specified file name in the output queue.
- One font file and one page data file per BIP. The font file name is based on the specified base file name and extension .mbf## where ## is the BIP number. The name of the file containing page data is based on the specified base file name and extension .mbp## where ## is the BIP number.

To specify the output format mode for MIBF output:

1. In Design Manager, from the Library, drag a Metacode output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Output** drop-down list, select the output format mode:
  - **MB2 combined**—Includes font and page data in one file
  - **MB2 separated**—Includes font and page data in separate files
4. Save the output object.

## 2.13.2 Specifying ICC Profiles

Exstream uses International Color Consortium (ICC) profile information during color conversions, so that the output contains the best possible color images. Color conversion takes place during packaging for static images and during the engine run for referenced images. For MIBF output, you can specify the ICC profile information to use.

To set up ICC profiles:

1. In Design Manager, from the Library, drag an MIBF output object to the Property Panel.
2. Click the **Configuration** tab.
3. In the **ICC Profiles** area, in the **Input Profile** box, click  and browse to the ICC profile path for input.
4. Click **OK**.
5. In the **Output Profile** box, click  and browse to the MIBF color information.

6. Click **OK**.
7. Save the output object.

**Note:** During a production run, Exstream applies the profiles to all images during packaging.

### 2.13.3 Setting Up an MIBF Printer Configuration

You specify an MIBF output configuration based on the number of print heads and bitmap image processors (BIPs) that are in use. Exstream provides options that let you define the dithering properties of the output object.

Since each MIBF printer can vary, you should specify configurations for each printer in a separate MIBF output object in the Library.

To configure an MIBF output object, you must complete the following tasks:

1. [“Configuring Print Heads” below](#)
2. [“Adding BIPs” on the next page](#)

#### Configuring Print Heads

You can configure one print head for each BIP. The total number of print heads cannot exceed eight in any single MIBF output object.

To configure a print head:

1. In Design Manager, from the Library, drag an MIBF output object to the Property Panel.
2. Click the **Configuration** tab.
3. In the **Print heads** area, do the following to add or remove a print head:

To	Do this
Add a print head	<ol style="list-style-type: none"><li>a. Select the logical RIP to which you want to add a print head.</li><li>b. Under <b>Print</b>, click  . A print head is added to the <b>Print heads</b> box below the logical RIP you selected.</li></ol>
Remove a print head	<ol style="list-style-type: none"><li>a. Select the print head you want to remove.</li><li>b. Under <b>Print</b>, click  . If a logical RIP has only one print head, removing the logical RIP removes the print head. If a logical RIP has only one print head, you cannot remove the print head.</li></ol>

4. Select a print head in the list and click .

The **Print Head Properties** dialog box opens.

5. To identify the print head number you are defining, view the inactive **Print head** box.
6. From the **Width** drop-down list, select one of the following options for the width of the print head:
  - **19" full**—Print in full-width mode.
  - **19" half**—Print in half-width mode.

**Note:** If you are using two BIPs per print head, the only option available is **19" half**.

7. From the **Ink color** drop-down list, identify the color of the ink used with the print head.
8. From the **Duplex side** drop-down list, select the side for duplex printing from the following options:
  - **Front**
  - **Back**
9. Save the output object.

## Adding BIPs

You can have two BIPs per print head.

To configure BIPs:

1. In Design Manager, from the Library, drag an MIBF output object to the Property Panel.
2. Click the **Configuration** tab.
3. To add a BIP to a print head, select **2 per print head** from the **BIPs** drop-down list. Design Manager adds BIPs sequentially.
4. To remove a BIP, select the item to remove and click .
5. Save the output object.

**Note:** If a print head has only one BIP, removing the BIP removes the print head.

## 2.14 Setting Up PCL Output

This section discusses the following topic:

- “[Using the PCL Output Driver on z/OS](#)” below

In addition to the topic discussed in this section, you can also use the following basic setup options for PCL output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>
Specifying how to set up unprintable areas in the page margins	<a href="#">“Specifying Unprintable Areas for Page Margins” on page 48</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Setting up paper bin options	<a href="#">“Setting Up Paper Bins” on page 50</a>

### 2.14.1 Using the PCL Output Driver on z/OS

If you are using the PCL output driver with a z/OS mainframe engine, you must set up the directory to receive the output. Use the following parameters for the PCL output to work correctly:

```
Record format . . . : VB
Record length . . . : 1048
Block size . . . . : 23476 Current Utilization Record format . . . : VB
Record length . . . . : 1048
Block size . . . . : 23476 Current Utilization
```

## 2.15 Setting Up PDF Output

This section discusses the following topics:

- [“About PDF Output” below](#)
- [“Applying a PDF Preset” on the next page](#)
- [“Customizing the PDF Output Setup” on page 138](#)
- [“Adding Hyperlinks to PDF Output” on page 141](#)
- [“Adding Metadata to PDF Output” on page 142](#)
- [“Creating PDF AcroForms output” on page 146](#)

In addition to the topics discussed in this section, you can also use the following basic setup options for PDF output:

Setup option	See this section for more information
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Setting up paper bin options	<a href="#">“Setting Up Paper Bins” on page 50</a>

PDF output objects have other unique properties that allow you to control how bookmarks are included in the output.

For information about controlling how bookmarks are included in PDF output, see [“Controlling how bookmark entries are included in output” on page 215](#).

### 2.15.1 About PDF Output

PDF output from Exstream is very versatile. You can customize how Exstream creates PDF output to meet specific delivery needs. Different combinations of settings on the PDF output object can affect processing, file size, resource management, and general file quality.

For example, you can set up your PDF output object using a [PDF preset](#), which is a pre-determined collection of settings that affect the way in which PDF output is created. The settings applied with each preset are intended to ensure that you receive PDF output with the right balance of file size, resource inclusion, and output quality to accommodate the way in which the PDF is delivered to customers. This method can be useful if you want to quickly set up a PDF output object with the optimal settings needed for a specific delivery method.

You can also [customize each setting on the PDF output object](#) as needed, based on your organization's output needs. This method is useful if your organization has specific PDF output needs that are not met by one of the common delivery methods used to define a PDF preset. For example, suppose you are delivering high-quality PDFs through the Internet. Using a **Web Delivery** preset might not provide the resolution controls you want, and a **Print High Quality** preset might not provide the quick online viewing controls you want. Instead, you can customize the individual PDF output settings to get a mixture of high resolution settings to increase the visual quality of the PDF and compression or linearization settings to provide better online viewing.

Exstream also supports creating editable PDF AcroForms output. You can design forms that include interactive controls such as editable text fields, pre-populated lists, check boxes, and radio buttons in your PDF or PDF/A output. You must add these controls to your design in Designer for them to appear in the output. For more information about adding PDF AcroForms controls to your design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Additionally, Exstream allows you to [add metadata to PDF output](#) by applying metadata objects to the PDF output object. The metadata that you apply in Design Manager is applied at the PDF document level, not on any lower-level components. Although you can apply metadata to individual objects in a database, only metadata that is applied to the output object appears in the PDF output.

## 2.15.2 Applying a PDF Preset

A PDF preset automatically sets up a PDF output object to create PDF output with the optimal PDF size, resource inclusion, and output quality for common delivery methods. For example, if you are delivering content through email, applying an **Email Delivery** preset can ensure that you get the smallest possible file size so that you do not overcrowd customer inboxes.

Before making a final decision to apply a preset, you can review preset settings to make sure that the settings meet your output needs. This can be useful if you are updating an existing PDF output object and you want to review how your current PDF output settings compare to using a specific preset.

If needed, you can change any preset setting after a preset has been applied. When you change a setting that is associated with a preset, Exstream issues a warning to confirm that you want to change the setting and the **PDF preset** value is changed to **Custom** to indicate that full preset settings are no longer in use. Keep in mind that custom PDF output settings cannot be stored as a preset for future use. If you have custom output settings and later apply a preset, you will not be able to restore the previous custom settings.

For more information about customizing the PDF output setup, see “[Customizing the PDF Output Setup](#)” below.

To apply a PDF preset:

1. In Design Manager, from the Library, drag a PDF output object to the Property Panel.
  2. Click the **Basic** tab.
  3. In the **PDF Preset** box, click  .
- The **PDF Preset Options** dialog box opens.
4. From the **PDF Presets** drop-down list, select one of the following preset options:
    - **Web Delivery**—This preset optimizes PDF output for viewing over the web. Some key features of this preset include a reduced file size and linearization for faster viewing over the web.
    - **Email Delivery**—This preset optimizes PDF output for delivery to customers through email. Some key features of this preset include a reduced file size and referenced resources for easier delivery.
    - **Print Standard**—This preset optimizes PDF output for standard printing. Some key features of this preset include an equal balance between files size and print quality.
    - **Print High Quality**—This preset optimizes PDF output for high quality printing. Some key features of this preset include a high output resolution and higher quality rendering of fonts and images included in the output.
    - **Archive**—This preset optimizes PDF output for archival storage. Some key features of this preset include embedded fonts and image resources for ease of access.

5. In the **Changes** area, review the settings that will be applied with the preset.
6. To approve the settings, click **OK**.

The PDF preset settings are applied to the PDF output object.

### 2.15.3 Customizing the PDF Output Setup

Customizing the PDF output setup lets you control each option in the PDF output individually to determine how PDF output is produced in Exstream. You can review each of the settings on the PDF output object to determine which settings best meet your organization's output needs.

**Tip:** It can be helpful to apply a preset option that closely resembles your organization's output needs and then make changes as needed. This technique can reduce the number of options you must modify to achieve the desired PDF output settings.

This section discusses the following topics:

- “[Specifying a PDF Version for Compatibility with Viewers, Process Tools, and Utilities](#)” below
- “[Specifying the Encoding Option for Compressed Data](#)” on the next page

In addition to the topic discussed in this section, you can also use the following basic setup options for PDF output:

#### Custom PDF output setup tasks

To	See this content for more information
Specify the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Specify how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a> <a href="#">“Setting Up Color Options for Output” on page 59</a>
Compress the PDF file	<a href="#">“Compressing PDF files” on page 206</a>
Linearize the PDF for faster online viewing	<a href="#">“Creating linearized PDF output” on page 207</a>
Creating editable PDF AcroForms as output	<a href="#">“Creating PDF AcroForms output” on page 146</a>
Make sure fonts appear correctly in the output	<a href="#">“Making sure fonts appear correctly in output” on page 189</a>
Calculate character widths for text positioning	<a href="#">“Calculating character widths for text positioning” on page 191</a>
Manage font resources	<a href="#">“Managing Font Resources” on page 229</a>
Manage image resources	<a href="#">“Managing Image Resources” on page 235</a>
Manage overlays and overlay processing	<a href="#">“Using Overlays in Output” on page 249</a>
Manage print resources	<a href="#">“Managing Print Resources” on page 249</a>
Add bookmarks to PDF output	<i>Designing Customer Communications</i> in the Exstream Design and Production documentation
Enable the use of electronic signatures	<i>Designing Customer Communications</i> in the Exstream Design and Production documentation

## Specifying a PDF Version for Compatibility with Viewers, Process Tools, and Utilities

The PDF version of PDF output is used to identify the output's compatibility with different PDF viewers, process tools, and utilities. For example, for maximum accessibility and reuse, PDF output that you intend to archive generally uses PDF version 1.3 so that the PDF is compatible with any version of Acrobat 4.0 or later. To determine the PDF version required for your

production and delivery needs, consult the documentation for any PDF viewer, processing tool, or utility your organization uses.

From Exstream, you can create PDF output using any of the following PDF versions:

- Version 1.3 (compatible with Acrobat version 4.0 and later)
- Version 1.4 (compatible with Acrobat version 5.0 and later)
- Version 1.5 (compatible with Acrobat version 6.0 and later)
- Version 1.6 (compatible with Acrobat version 7.0 and later)
- Version 1.7 (compatible with Acrobat version 8.0 and later)

Each PDF version uses a specific set of PDF capabilities. Depending on the PDF version you use, different PDF setup options are available. Individual version requirements are listed with specific setup tasks, where applicable.

To specify a PDF version:

1. In Design Manager, from the Library, drag a PDF output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Version** drop-down list, select the appropriate version.
4. From the Menu bar, select **File > Save**.

## Specifying the Encoding Option for Compressed Data

For PDF output, you can specify how compressed data should be encoded in the PDF print stream. Exstream allows either binary or ASCII85 formats. In general, binary data creates smaller files.

To specify the encoding option for compressed data:

1. In Design Manager, from the Library, drag a PDF output object to the Property Panel.
2. Click the **Basic** tab.
3. To include binary data (such as images) in the PDF file, select the **Binary allowed** check box. If you clear the **Binary allowed** check box, Exstream uses ASCII85 format to encode binary data.
4. Save the output object.

## 2.15.4 Creating PDF AcroForms output

If you have licensed the Interactive Forms module, you can create editable PDF AcroForms as output. You can add interactive controls such as text fields, combo boxes, check boxes, radio

buttons, and date pickers to your PDF or PDF/A output. You must add these controls to your design in Designer for them to appear in the output.

For more information about adding PDF AcroForms controls to your design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Exstream supports creating PDF AcroForms for PDF versions 1.4 and later, and for all supported versions of PDF/A. PDF AcroForms output is not supported for PDF/VT, linearized PDFs, PDFs that use subset fonts, or PDFs where each page is an image. When you create PDF AcroForms, the engine forces TrueType fonts because AcroForms support only embedded TrueType fonts.

To create PDF AcroForms output:

1. Open a PDF or PDF/A output object in the Property Panel.
2. For PDF output, in the **Version** list, select **1.4** or later.
3. Clear the **Linearized** check box, if selected.
4. Select the **AcroForm** check box.

PDF AcroForms support only one customer per file, so breaks on the output queue must be set to break on each customer. To verify that the breaks are set on the output queue:

1. Open the output queue object in the Property Panel.
2. Click the **Output** button .
3. In the **Select Output** dialog box, choose the PDF or PDF/A output object that has PDF AcroForms enabled and click **OK**.
4. Click **OK** on the message that you receive stating that the breaks settings were changed.
5. Save the output queue object.

## 2.15.5 Adding Hyperlinks to PDF Output

You can add hyperlinks to a design to enable customers to easily access additional sources of information, such as websites, other documents, or other locations within the same document. You can assign hyperlinks to specific text in a design as well as to the following objects: text boxes, static or dynamic images, polygons, predefined shapes, charts, paragraph objects, and message objects. For external hyperlinks, which direct customers to external resources such as websites, the destination address can be static or dynamic (set by variable values). A customer will use a single click to follow hyperlink addresses in PDF output.

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

## 2.15.6 Adding Metadata to PDF Output

**Note:** The concepts surrounding PDF metadata discussed in this section also apply to PDF/A, PDF/VT, and VDX output.

You can include metadata as part of a PDF by applying metadata objects to the PDF output object. The metadata you apply in Design Manager is applied at the PDF level—that is, although you can apply metadata to individual objects in the Library, only metadata that is applied to the output object appears in the PDF output.

Metadata lets you locate specific files more efficiently, especially if you are dealing with a large number of PDFs. You can apply static or variable values for each metadata object that you include in your PDF output. For example, suppose you have several thousand customer PDF files but need to locate only those for customers in a certain city. In this scenario, you can create a metadata object that uses an address variable as its value. After you apply the metadata object and create PDF output, the customer address is available as metadata on each customer's PDF. You can then use a search engine to look for only that term, instead of conducting a time-intensive full-text search on the entire set of PDF files.

Exstream provides a set of system metadata objects that you can apply to your PDF output. These objects (SYS\_Author, SYS\_Keywords, SYS\_Subject, and SYS\_Title) each map to a corresponding metadata term (Author, Keywords, Subject, and Title, respectively) in the PDF Document Information Dictionary and appear as part of the "Description" in the document properties of the PDF file.

You can also create your own metadata objects to apply to your PDF output. The metadata specified by these objects appears as part of the "Custom Properties" in the document properties of the PDF file. You use the same method to create these metadata objects as you do for other metadata objects used elsewhere in your database.

For more information about creating metadata objects, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## Special Considerations for Naming PDF Metadata

If you use Adobe Acrobat to read your PDF files, the metadata that you apply to your PDF output object appears in the PDF file properties. When you apply metadata terms for use in Adobe Acrobat, keep in mind that you cannot use the "SYS\_" prefix when you name metadata objects in Exstream. Moreover, Adobe reserves the following metadata names, which you cannot use to create custom metadata names in your PDF output:

- Author (use the Exstream SYS\_Author system metadata object instead)
- Keywords (use the Exstream SYS\_Keywords system metadata object instead)
- Subject (use the Exstream SYS\_Subject system metadata object instead)
- Title (use the Exstream SYS\_Title system metadata object instead)

- Creator
- Producer
- CreationDate
- ModDate
- Trapped

For more information about using metadata in Adobe Acrobat, see the Acrobat documentation.

## Using Metadata Objects in PDF Output

This topic assumes that you are either using system metadata objects, or that you have already created your own metadata objects.

For more information about creating metadata objects, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To use metadata objects in PDF output:

1. In Design Manager, in the Library, right-click the PDF output object and select **Metadata**.
2. In the **Available Metadata** list, select a metadata object and click  to apply it to the object.
3. In the **Applied Metadata** list, select the metadata object that you just applied to the output object.
4. In the **Metadata value** area, do one of the following:

To	Do this
Use a static value for the metadata object	<ol style="list-style-type: none"><li>a. Enter a text value in the <b>Value</b> field.</li><li>b. Click <b>OK</b>.</li></ol>
Use a variable value for the metadata object	<ol style="list-style-type: none"><li>a. Click .</li><li>b. Select a variable from the list and click <b>OK</b>.</li></ol> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"><p><b>Note:</b> You should use a string variable when assigning a variable value to a metadata object. Other variable types can produce unexpected results in the output.</p></div> <ol style="list-style-type: none"><li>c. Click <b>OK</b>.</li></ol>

**Note:** You must assign a value to metadata objects that you apply to PDF, PDF/A, PDF/VT, or VDX output objects.

5. Drag the PDF output object to the Property Panel.
6. On the **Basic** tab, select the **Include metadata objects in the output** check box.  
Selecting this check box removes the PDF metadata that Exstream includes by default and replaces it with the metadata that you have specified.

**Note:** If you do not add metadata objects to your PDF output, Exstream automatically applies values for the "Title" (the name of the application) and "Author" (the name of the user on the license) metadata terms.

7. From the Menu bar, select **File > Save**.

## 2.16 Setting Up PDF/A Output

This section discusses the following topics:

- “[Specifying the Color Profile for PDF/A Output](#)” on the next page
- “[Creating PDF AcroForms output](#)” on page 146
- “[Adding Static Subject and Keyword Metadata to the Output File](#)” on page 147

In addition to the topics discussed in this section, you can also use the following basic setup options for PDF/A output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>

Setup option	See this section for more information
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>
Setting up color formatting options	<a href="#">"Setting Up Color Options for Output" on page 59</a>
Allowing binary data	<a href="#">"Specifying the Encoding Option for Compressed Data" on page 140</a>
Adding metadata to the output object	<a href="#">"Adding Metadata to PDF Output" on page 142</a>

PDF/A output supports the inclusion of hyperlinks in the output file. You can design hyperlinks on pages for specific text or a paragraph within a text box. You can also assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The link addresses can be static or dynamic (set by variable values). Keep in mind that an end user cannot follow hyperlinks in PDF/A output. However, the hyperlink address is visible when an end user hovers the pointer over the text, image, or shape that includes the hyperlink.

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

PDF/A output objects have other unique properties that allow you to control how bookmarks are included in the output.

For information about controlling how bookmarks are included in PDF/A output, see ["Controlling how bookmark entries are included in output" on page 215](#).

As you set up PDF/A output, keep in mind that the following Exstream options that are available for PDF output are not available for PDF/A output:

- **Multiple-ups**—Accessibility tags are not compatible with multiple-up output. Since accessibility tags are automatically enabled for PDF/A-1a output, if you specify multiple-up options for an output queue with a PDF/A-1a driver, the output file will be compliant only with PDF/A-1b specifications.
- **Highlight colors**—You cannot use highlight colors with PDF/A output.
- **Encryption**—Encryption and password-protection are not supported for PDF/A output. You can, however, add a digital signature to PDF/A output.

## 2.16.1 Specifying the Color Profile for PDF/A Output

If the color in your PDF/A document must be reproduced accurately, you can specify a color profile for PDF/A output. Color profiles are commonly maintained in an industry-standard registry, such as the International Color Consortium (ICC) Characterization Data Registry.

To specify the color profile for PDF/A output:

1. In Design Manager, from the Library, drag a PDF/A output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Output Condition Identifier** box, enter the reference name of the ICC profile to use for your PDF/A output. The default is sRGB.
4. If you do not use the sRGB default setting in the **Output Condition Identifier** box, you must enter the location of the ICC profile in the **Destination Profile** box.
5. Save the output object.

## 2.16.2 Creating PDF AcroForms output

If you have licensed the Interactive Forms module, you can create editable PDF AcroForms as output. You can add interactive controls such as text fields, combo boxes, check boxes, radio buttons, and date pickers to your PDF or PDF/A output. You must add these controls to your design in Designer for them to appear in the output.

For more information about adding PDF AcroForms controls to your design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Exstream supports creating PDF AcroForms for PDF versions 1.4 and later, and for all supported versions of PDF/A. PDF AcroForms output is not supported for PDF/VT, linearized PDFs, PDFs that use subset fonts, or PDFs where each page is an image. When you create PDF AcroForms, the engine forces TrueType fonts because AcroForms support only embedded TrueType fonts.

To create PDF AcroForms output:

1. Open a PDF or PDF/A output object in the Property Panel.
2. For PDF output, in the **Version** list, select **1.4** or later.
3. Clear the **Linearized** check box, if selected.
4. Select the **AcroForm** check box.

PDF AcroForms support only one customer per file, so breaks on the output queue must be set to break on each customer. To verify that the breaks are set on the output queue:

1. Open the output queue object in the Property Panel.
2. Click the **Output** button .
3. In the **Select Output** dialog box, choose the PDF or PDF/A output object that has PDF AcroForms enabled and click **OK**.

4. Click **OK** on the message that you receive stating that the breaks settings were changed.
5. Save the output queue object.

### 2.16.3 Adding Static Subject and Keyword Metadata to the Output File

In Exstream, as part of PDF/A output, you can add static subject and keyword information about the content of the documents you are creating. The option to add this information is required for PDF/A compliance, but the information itself is optional. The information shows up in PDF/A output as Subject and Keywords metadata, and it can be used by search utilities to help you find information. For example, you can enter multiple keywords that can later help you find a document in a focused search.

Exstream also allows you to add a more robust set of metadata objects to your PDF/A output, including system metadata for author, keywords, subject, and title, in addition to an unlimited number of metadata objects that you can create yourself. If you add additional metadata objects to your PDF/A output, keep in mind that the **Subject** and **Keywords** fields will be unavailable on the PDF/A output object. As an alternative, you can apply the SYS\_Subject and SYS\_Keywords system metadata objects to your PDF/A output. Additionally, you can assign variables for the values of those two system metadata objects, which you cannot do when using the **Subject** and **Keywords** fields on the PDF output object.

To add static subject and keyword metadata to the PDF/A output file:

1. In Design Manager, from the Library, drag a PDF/A output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Subject** box, you can enter information about the content of the file.
4. In the **Keywords** box, you can enter keyword information about the content of the documents.

**Note:** If you want to add additional metadata to the output object, you must select the **Include metadata objects in the output** check box and then apply the metadata separately. If you select the check box, then the **Subject** and **Keywords** fields will be unavailable.

For more information about applying metadata objects to PDF output, see “[Adding Metadata to PDF Output](#)” on page 142.

5. From the Menu bar, select **File > Save**.

## 2.17 Setting Up PDF/VT Output

This section discusses the following topics:

- [“Specifying the Color Profile for PDF/VT Output” on the next page](#)
- [“Using Document Part Metadata for PDF/VT Output” on page 150](#)

In addition to the topics discussed in this section, you can also use the following setup options for PDF/VT output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<ul style="list-style-type: none"><li>• <a href="#">“Specifying the Way Color is Produced in Output” on page 40</a></li><li>• <a href="#">“Setting Up Color Options for Output” on page 59</a></li></ul>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Controlling how bookmarks are included	<a href="#">“Controlling how bookmark entries are included in output” on page 215</a>
Compressing the PDF file	<a href="#">“Compressing PDF files” on page 206</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Using search keys in PDF/VT output	<a href="#">“Using Application Search Keys” on page 326</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Allowing binary data	<a href="#">“Specifying the Encoding Option for Compressed Data” on page 140</a>
Adding hyperlinks to the output	<a href="#">“Adding Hyperlinks to PDF Output” on page 141</a>
Adding metadata to the output object	<a href="#">“Adding Metadata to PDF Output” on page 142</a>
Setting up paper bin options	<a href="#">“Setting Up Paper Bins” on page 50</a>

As you set up Exstream PDF/VT output, keep in mind that the following options that are available for PDF output are not available for PDF/VT output:

- Accessibility
- Permission options
- Encryption and password-protection
- Digital signatures

Additionally, the following output object options (found on the **Resource Management** tab) cannot be adjusted for PDF/VT output:

Option	PDF/VT static setting
Font usage	Embed all fonts
Image processing	Place at top of print stream
Overlay processing	Create and include in print file
Resource inclusion	Used resources only

## 2.17.1 Specifying the Color Profile for PDF/VT Output

If the colors in your document must be reproduced accurately, you can specify a color profile for PDF/VT output. Color profiles are commonly maintained in an industry-standard registry, such as the International Color Consortium (ICC) Characterization Data Registry.

In order to achieve the correct colors in printed output and pass the printer preflight process, the following design and output object properties must have the same color settings (**CMYK** or **RGB**):

- The ICC color profile of the output object
- The **Color model** selection of the output object
- The color selection of the design objects in the application (including images imported during run time or design time)

**U.S. Web Coated (SWOP)**, which has the reference name CGATS TR 001, is the only color profile that is installed by Exstream for PDF/VT output. However, because this profile is likely to be unsuitable for PDF/VT output in many cases, OpenText recommends that you change the color profile to best match the devices and media on which your PDF/VT output will be printed.

**Note:** The file for this color profile, USWebCoatedSWOP.icc, is installed in the same directory as Exstream Design and Production.

To specify the color profile for PDF/VT output:

1. In Design Manager, from the Library, drag a PDF/VT output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Output Condition Identifier** box, enter the reference name of the ICC profile to use for your PDF/VT output.
4. If you do not use the default **Output Condition Identifier** setting of CGATS TR 001, you must enter the location of the ICC color profile in the **Destination Profile** box.

You can click  to browse for the profile that you want to use.

5. Save the output object.

## 2.17.2 Using Document Part Metadata for PDF/VT Output

The PDF/VT file format utilizes a data structure known as document part hierarchy to specify the sequence and relationship of documents or document parts within the file. The components of this hierarchy are known as document part (DPart) nodes. In addition to conveying the structure of the document parts, DPart nodes can also have document part metadata (DPM) associated with each node in the hierarchy. The DPart structure and any associated DPM contribute to the job definition format (JDF) job ticket for the PDF/VT output.

DPM is a standardized metadata format that can include customer processing information and production workflow commands. For example, you can use DPM to sort customer records by postal code, or include instructions for a digital press to change paper stock depending on the parts of a document being printed so that cover pages use heavier paper than inside pages.

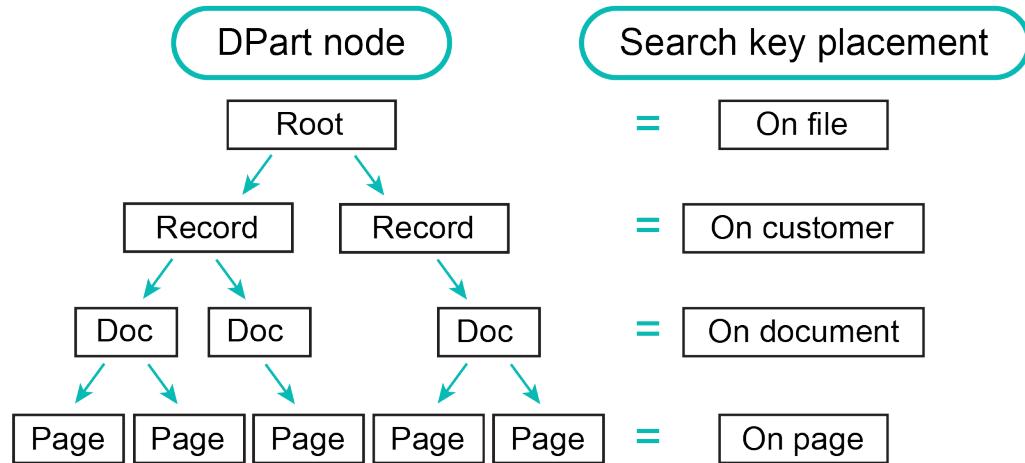
Exstream uses [application search keys](#) to convey DPM in PDF/VT output.

For additional information about DPM search key implementation, see the following specifications:

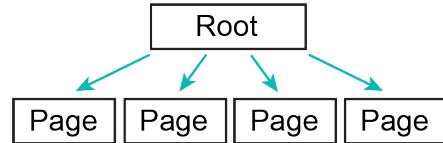
- ISO 32000 PDF standard
- ISO 16612 PDF/VT standard
- CIP4 Interoperability Conformance Specifications (ICS) that relate to PDF and PDF/VT

## DPart Tree Structure in Exstream

Exstream applies DPM search keys to its DPart nodes using the following tree structure:



However, if your application is set to use multiple-up pages, Exstream uses the following structure:



Each **Page** in the tree structure represents a design page. PDF output pages created from a design page become children of that particular page node.

## DPM Search Key Format

Valid PDF/VT DPM search key strings are written in the form /Keyname (Value).

### Example 1: DPM search key in simple format

This style of search key is entered without the tree structure.

/Exstream\_CustomerZip (24601)

### Example 2: DPM search keys in CIP4 format

This style of search key is entered with the tree structure.

```
/CIP4_Root <<
    /CIP4_Summary <<
        /CIP4_PageCount 7
    >>
    /CIP4_Production <<
        /CIP4_CopyCount 1
        /CIP4_Part <<
            /CIP4_ProductType /Pamphlet
        >>
    >>
    /CIP4_Recipient <<
        /CIP4_UniqueID (ID_317)
        /CIP4_Contact <<
            /CIP4_Person <<
                /CIP4_FirstName (Margaret)
                /CIP4_LastName (Simpson)
            >>
            /CIP4_Address <<
                /CIP4_StreetName (Evergreen Terrace)
                /CIP4_CivicNumber (742)
                /CIP4_City (Springfield)
                /CIP4_PostalCode (40069)
                /CIP4_Region (KY)
                /CIP4_Country (USA)
            >>
        >>
    >>
>>
```

## Using DPM in Conjunction With a PDF/VT Media Name

When you specify the **PDF/VT media name** for the [paper type object](#), for your PDF/VT output, by default, the media name is inserted in CIP4 format, with the following tree structure:

```
/CIP4_Root/.../CIP4_MediaQuality (<media name>)
```

If you include DPM search keys for PDF/VT output and you have also entered a PDF/VT media name for the corresponding paper type object, OpenText recommends that you insert the media name in simple (non-CIP4) format.

If you are using CIP4-style DPM search keys, the default setting for the paper type media name results in output that does not pass PDF validation. This is because both the paper type PDF/VT media name and the search keys add a /CIP4\_Root node to the document part hierarchy tree structure. Having two root nodes creates a conflict in the output file.

Even if you are using DPM search keys that are not in CIP4 format, OpenText recommends that you insert the media name in simple format. Doing so lets you use your paper type object without having to set up a CIP4 tree structure for it.

To insert the PDF/VT media name in simple format:

1. In Design Manager, in the Library, go to **Environment > Delivery > Outputs**.
2. Open the PDF/VT output object in the Property Panel.
3. On the **Basic** tab, in the **PDF/VT controls** area, select the **Insert media name as is** check box.

Selecting the **Insert media name as is** check box enters the PDF/VT media name for the object as <media name>, without creating the tree structure.

The following table provides information about the validity of your PDF/VT output based on the format of your search keys when also including a PDF/VT media name in the paper type object:

Format considerations for DPM search keys

If you use	And	Then your output is
CIP4 format DPM search keys	The <b>Insert media name as is</b> check box is selected	Valid
	The <b>Insert media name as is</b> check box is not selected	Not valid
Simple format DPM search keys	The <b>Insert media name as is</b> check box is selected	Valid
	The <b>Insert media name as is</b> check box is not selected	Valid, but not recommended

## 2.18 Setting Up PostScript Output

This section discusses the following topic:

- “[Controlling the Type of Comments Sent to the Output Device](#)” on the next page

In addition to the topics discussed in this section, you can also use the following basic setup options for PostScript output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>

Setup option	See this section for more information
Setting up simplex and duplex options	<a href="#">"Using Simplex and Duplex in Output" on page 39</a>
Specifying how to use color	<a href="#">"Specifying the Way Color is Produced in Output" on page 40</a>
Creating each page as an image	<a href="#">"Creating Pages as Images" on page 44</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">"Using Search Keys in Output" on page 46</a>
Specifying how to set up unprintable areas in the page margins	<a href="#">"Specifying Unprintable Areas for Page Margins" on page 48</a>
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>
Setting up paper bin options	<a href="#">"Setting Up Paper Bins" on page 50</a>
Setting up color formatting options	<a href="#">"Setting Up Color Options for Output" on page 59</a>

## 2.18.1 Controlling the Type of Comments Sent to the Output Device

Document Structuring Convention (DSC) commands are comments on PostScript-based output files that printers and viewers use to determine information about the file. DSC commands divide a PostScript file into three main portions: the header (or prologue), the page area, and the trailer.

DSC commands provide page independence, which allows the pages to be rendered in any sequence. As a result, the header should contain all setup information, and the trailer code restores the interpreter state to that which existed before the file was processed. Each page should contain the information required to render that page, including text, graphics, or font data required by that page.

To control which DSC commands to send to a PostScript device:

1. In Design Manager, from the Library, drag a PostScript or PostScript-based output object to the Property Panel.
2. From the **DSC commands** drop-down list, select one of the following options:
  - **Default**—To allow only basic comments, if you are concerned with backward compatibility
  - **Enhanced**—To allow additional comments that printers and viewers use to determine

information about the file. The engine places the most up-to-date DSC commands in the print stream.

3. Save the output object.

## 2.19 Setting Up PowerPoint Output

This section discusses the following topic:

- “[Creating PowerPoint Output as a Single File or with One File Per Page](#)” below

In addition to the topic discussed in this section, you can also use the following basic setup options for PostScript output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>

When you create PowerPoint output, all pages in the output are either portrait or landscape but not mixed.

### 2.19.1 Creating PowerPoint Output as a Single File or with One File Per Page

The PowerPoint output driver produces modified HTML output. The files imported to PowerPoint are HTML with PowerPoint commands for design objects such as lines, shapes, and text boxes.

1. In Design Manager, from the Library, drag a PowerPoint output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Output type** drop-down list, select one of the following options:
  - **Single file**—Exstream creates a single file that contains all the PowerPoint pages.

- **One file per page**—Exstream creates a separate file for each PowerPoint page.
4. Save the output object.

## 2.20 Setting Up PPML Output

You can use the following basic setup options for PPML output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>
Specifying how to set up unprintable areas in the page margins	<a href="#">“Specifying Unprintable Areas for Page Margins” on page 48</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Setting up paper bin options	<a href="#">“Setting Up Paper Bins” on page 50</a>
Setting up color formatting options	<a href="#">“Setting Up Color Options for Output” on page 59</a>
Controlling which DSC commands to send to the output device	<a href="#">“Controlling the Type of Comments Sent to the Output Device” on page 154</a>

## 2.21 Setting Up RTF Output

This section discusses the following topics:

- [“Using Absolute Object Positioning” on the next page](#)
- [“Setting Up Editable RTF Output” on page 158](#)

In addition to the topics discussed in this section, you can also use the following basic setup options for RTF output:

Setup option	See this section for more information
Creating an output object	<a href="#">"Creating an Output Object" on page 34</a>
Specifying how to use color	<a href="#">"Specifying the Way Color is Produced in Output" on page 40</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">"Using Search Keys in Output" on page 46</a>
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>

## 2.21.1 Using Absolute Object Positioning

Using absolute positioning is intended for RTF output that you do not plan to change in Microsoft Word or RTF output with advanced formatting (such as borders around text boxes). When you use the absolute object positioning, keep in mind the following:

- RTF output is produced with the text box properties you designed.
- Objects appear in fixed positions without relativity. For example, a text box you change in Microsoft Word can overlap other objects or leave gaps, and if you remove objects from the page, the remaining objects retain their absolute position.
- Headers and footers are static.

For more information about relativity, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To specify absolute positioning:

1. In Design Manager, from the Library, drag an RTF output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Object Positioning** drop-down list, select the **Absolute positioning with word objects** option.
4. Save the output queue object.

## 2.21.2 Setting Up Editable RTF Output

Editable RTF output can easily be revised using a separate software tool (for example, Microsoft Word). Editable RTF output produces an RTF output file that you can change in Microsoft Word without disrupting the layout. Editable RTF output includes the following features:

- Each customer is a continuous stream of RTF data that can be repaginated or edited in Microsoft Word while keeping the document properly formatted.
- Text box properties, such as borders, are ignored.
- Objects move in predictable ways as you change the content in Microsoft Word. For example, if you remove a line of text, objects below the line move up.
- You can define headers and footers.

Keep in mind that objects cannot overlap in RTF output when you use Editable RTF. Overlapped objects on your design page are placed apart in the actual output file.

To set up Editable RTF:

1. In Design Manager, from the Library, drag an RTF output object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Object Positioning** drop-down list, select **Editable top-down RTF with Graphic Objects**.
4. In the **X Margin** box, set the default left and right margins for all documents.
5. In the **Y Margin** box, set the default top and bottom margins for all documents.
6. Save the output object.

**Tip:** You can make changes to the **X Margin** and **Y Margin** defaults on individual pages on the **Basic** tab of a page in the Property Panel.

## Designing Page Headers and Footers for Editable RTF Output

In Designer, the placement of objects on a page can determine headers and footers in the RTF output. If you have a document in Designer with one ordered page and repeating flow pages, objects above the top design margin become header content, and objects below the bottom design margin become footer content for each page. Since the headers and footers are processed inline in the file between the on-page data, the objects at the top and bottom of the page remain if the file is edited or repaginated. New pages in an edited RTF file do not have headers or footers.

For more information about creating flowing documents, see *Designing Customer Communications* in the Exstream Design and Production documentation.

In a flow section of an RTF file, objects in the header and footer are static. The engine uses the header and footer created on the last page in the flow section on every flowing page. For example, if objects flow to pages 4–8 of your document, the page number will be page 8 on each page within that range. Page numbering resumes as usual after the flow section.

**Caution:** For RTF output, you cannot have a duplex flow page with static content on the back. This results in one page of flowing data, followed by the contents of the back page, and so on. The document does not repaginate properly.

## Setting the Header Margins

You can specify a margin for the top of a page so that objects you design above the specified margin become the content of a running header in an RTF output file.

To set the header margin for RTF output:

1. In Design Manager, from the Library, drag a page object to the Property Panel.
2. Click the **Basic** tab.
3. In the **RTF header size** box, specify the margin for the header of the page. The top of the page, above the margin you set, is used as the header. Make sure the number you enter is large enough on the first page to capture any non-relative objects that follow a flowing table or text box.
4. Save the page object.

## Setting the Footer Margins

You can specify a margin for the top of a page so that objects you design below the specified margin become the content of a running footer in an RTF output file.

1. In Design Manager, from the Library, drag a page object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Page flow margin** box, specify the margin for the footer of the page. The bottom of the page, below the margin you set, is used as the footer. Make sure the number you enter is large enough on the flow page to capture non-relative objects before and after the flow frame. Be careful not to overlap the flow frame.
4. Save the page object.

## 2.22 Setting Up TIFF Output

You can use the following basic setup options for TIFF output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>

## 2.23 Setting Up TOP Output

You can use the following basic setup options for TOP output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>
Specifying how to set up unprintable areas in the page margins	<a href="#">“Specifying Unprintable Areas for Page Margins” on page 48</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>

Setup option	See this section for more information
Setting up paper bin options	<a href="#">"Setting Up Paper Bins" on page 50</a>
Setting up color formatting options	<a href="#">"Setting Up Color Options for Output" on page 59</a>
Controlling which DSC commands to send to the output device	<a href="#">"Controlling the Type of Comments Sent to the Output Device" on page 154</a>

## 2.24 Setting Up VDX Output

You can use the following basic setup options for VDX output:

Setup option	See this section for more information
Creating an output object	<a href="#">"Creating an Output Object" on page 34</a>
Specifying the resolution for your output	<a href="#">"Managing output resolution in Exstream" on page 37</a>
Setting up simplex and duplex options	<a href="#">"Using Simplex and Duplex in Output" on page 39</a>
Specifying how to use color	<a href="#">"Specifying the Way Color is Produced in Output" on page 40</a>
Creating each page as an image	<a href="#">"Creating Pages as Images" on page 44</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">"Using Search Keys in Output" on page 46</a>
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>
Setting up color formatting options	<a href="#">"Setting Up Color Options for Output" on page 59</a>
Adding metadata to the output object	<a href="#">"Adding Metadata to PDF Output" on page 142</a>

VDX output supports the inclusion of hyperlinks in the output file. You can design hyperlinks on pages for specific text or a paragraph within a text box. You can also assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The link addresses can be static or dynamic (set by variable values). An end user will use a single click to follow hyperlink addresses in VDX output.

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

VDX output objects have other unique properties that allow you to control how bookmarks are included in the output.

For information about controlling how bookmarks are included in VDX output, see “[Controlling how bookmark entries are included in output](#)” on page 215.

## 2.25 Setting Up VIPP Output

You can use the following basic setup options for VIPP output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Specifying the resolution for your output	<a href="#">“Managing output resolution in Exstream” on page 37</a>
Setting up simplex and duplex options	<a href="#">“Using Simplex and Duplex in Output” on page 39</a>
Specifying how to use color	<a href="#">“Specifying the Way Color is Produced in Output” on page 40</a>
Creating each page as an image	<a href="#">“Creating Pages as Images” on page 44</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>
Highlighting added text or images in your output with a black underline	<a href="#">“Highlighting New Text and Images in Output” on page 49</a>
Setting up paper bin options	<a href="#">“Setting Up Paper Bins” on page 50</a>
Setting up color formatting options	<a href="#">“Setting Up Color Options for Output” on page 59</a>
Controlling which DSC commands to send to the output device	<a href="#">“Controlling the Type of Comments Sent to the Output Device” on page 154</a>

## 2.26 Setting Up VPS Output

You can use the following setup options for VPS output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>

Setup option	See this section for more information
Specifying the resolution for your output	<a href="#">"Managing output resolution in Exstream" on page 37</a>
Setting up simplex and duplex options	<a href="#">"Using Simplex and Duplex in Output" on page 39</a>
Specifying how to use color	<a href="#">"Specifying the Way Color is Produced in Output" on page 40</a>
Creating each page as an image	<a href="#">"Creating Pages as Images" on page 44</a>
Controlling the comments you can send to the output device	<a href="#">"Controlling the Type of Comments Sent to the Output Device" on page 154</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">"Using Search Keys in Output" on page 46</a>
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>
Setting up paper bin options	<a href="#">"Setting Up Paper Bins" on page 50</a>
Setting up color formatting options	<a href="#">"Setting Up Color Options for Output" on page 59</a>
Controlling which DSC commands to send to the output device	<a href="#">"Controlling the Type of Comments Sent to the Output Device" on page 154</a>

## 2.27 Setting Up XML (Composed) Output

This section discusses information about setting up XML (composed) output in Exstream.

In addition to the topic discussed in this section, you can also use the following basic setup options for XML (composed) output:

Setup option	See this section for more information
Creating an output object	<a href="#">"Creating an Output Object" on page 34</a>
Specifying the resolution for your output	<a href="#">"Managing output resolution in Exstream" on page 37</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">"Using Search Keys in Output" on page 46</a>
Highlighting added text or images in your output with a black underline	<a href="#">"Highlighting New Text and Images in Output" on page 49</a>

XML (composed) output supports the inclusion of hyperlinks in the output file. You can design hyperlinks on pages for specific text or a paragraph within a text box. You can also assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The link addresses can be static or dynamic (set by variable values). Keep in mind that an end user cannot follow hyperlinks in XML (composed) output. However, the hyperlink address is visible when an end user hovers the pointer over the text, image, or shape that includes the hyperlink.

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

XML (composed) supports the inclusion of metadata tags in the output file. You can add metadata to XML (composed) output following the same steps as you would for other objects in Design Manager and Designer. With XML (composed), any metadata tag applied to an object in Design Manager or Designer appears in the output file.

For more information on creating and adding metadata in Design Manager and Designer, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## 2.27.1 Applying an Encoding to XML (Composed) Output (DBCS)

For DBCS applications, you must apply an XML encoding (or the default code page) to create valid XML output. You must set the properties of the encoding to allow it to be used for output files.

For more information on encoding, see *System Administration* in the Exstream Design and Production documentation.

The encoding you specify is displayed in the XML header.

For example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

To apply an encoding to XML output:

1. In Design Manager, from the Library, drag an XML (composed) output object in the Property Panel.
2. Click the **Basic** tab.
3. Specify an encoding in one of the following ways:
  - Click the **Encoding** box and select an encoding.

- From the Library, drag an encoding to the **Encoding** box.
4. Save the output object.

## 2.28 Setting Up XML (Content) Output

This section discusses the following topics:

- “[Defining a Basic XML \(Content\) Output Object](#)” below
- “[Producing UTF-8 XML \(Content\) Output](#)” on page 169

In addition to the topics discussed in this section, you can also use the following basic setup options for XML (content) output:

Setup option	See this section for more information
Creating an output object	<a href="#">“Creating an Output Object” on page 34</a>
Assigning a connector for testing and production	<a href="#">“Assigning a Connector to an Output Object for Testing and Production” on page 45</a>
Specifying the type of search keys that can be used with the output	<a href="#">“Using Search Keys in Output” on page 46</a>

XML (content) supports the inclusion of metadata tags in the output file. You can add metadata to XML (content) output following the same steps as you would for other objects in Design Manager and Designer. With XML (content), any metadata tag applied to an object in Design Manager or Designer appears in the output file.

For more information on creating and adding metadata in Design Manager and Designer, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 2.28.1 Defining a Basic XML (Content) Output Object

1. In Design Manager, from the Library, drag an XML (content) output object to the Property Panel.
2. Click the **Basic** tab.

3. From the **Variable output** drop-down list, select the amount of detail you want to print for variables:

- **Inline text**—In the output, the variable values appear as regular text within the paragraph where they occur.

#### Inline text example

```
<dlg:Message name="Test_Para">
  <dlg:Metadata name="Met4" value="Metadata 4" />
  <dlg:Metadata name="Met2" value="Metadata 2" />
  <dlg:Paragraph>This is paragraph text</dlg:Paragraph>
  <dlg:Paragraph />
  <dlg:Paragraph>This is paragraph text</dlg:Paragraph>
  <dlg:Paragraph>This is paragraph text</dlg:Paragraph>
</dlg:Message>
```

- **Complete variable content**—In the output, the individual variables and their attributes (name, data type, and value) appear within in the paragraph where they occur.

#### Complete variable content example

```
<dlg:TextBox name="Text Box">
  <dlg:Paragraph>
    Account Number:
    <dlg:Variable name="CustomerNumber"
      data-type="integer">4875113</dlg:Variable>
    <dlg:Variable name="StatementBegin"
      data-type="date">Dec 18 2005</dlg:Variable>
    through
    <dlg:Variable name="StatementEnd"
      data-type="date">Jan 22 2006</dlg:Variable>
    Page
    <dlg:Variable name="SYS_PageInDocument"
      data-type="integer">1</dlg:Variable>
    of
    <dlg:Variable name="SYS_PageTotalInDocument"
      data-type="integer">3</dlg:Variable>
  </dlg:Paragraph>
</dlg:TextBox>
```

4. To apply an encoding to an XML output file, do one of the following:

To	Do this
Specify an encoding for your XML data files if you are working with an SBCS application	<ol style="list-style-type: none"><li>Click <b>Encoding string</b> box.</li><li>Enter the encoding string for a known, standard encoding you want to use.</li></ol>
Specify the XML encoding (the default code page) if you are working with a DBCS application	<p>Do either one of the following:</p> <ul style="list-style-type: none"><li>Click the <b>Encoding</b> box and select an encoding.</li><li>Drag an encoding into the <b>Encoding</b> box.</li></ul>

The encoding you specify is displayed in the XML header.

For example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

For more information about encodings supported with XML (content) output, see [“Supported Encodings for XML \(Content\) Output” below](#).

5. Save the output object.

## Supported Encodings for XML (Content) Output

The following table lists encodings supported for SBCS applications:

Encodings supported by Exstream

Encoding	ID String
ASCII	ISO-8859
EBCDIC	IBM-37
NATIVE	ISO-8859-1 (Native ASCII)
NATIVE	IBM-37 (Native EBCDIC)
BIG5 with HKSCS	IBM-1375_P100-2007
Chinese EUC	IBM-1383
Chinese2	ISO_2022,locale=zh,version=0
Chinese Standard Interchange Code	CNS11643-1992
Korean1	IBM-970

Encodings supported by Exstream, continued

Encoding	ID String
Korean2	ISO_2022,locale=ko,version=0
Japan EUC	IBM-33722
Japanese1	IBM-33722
Japanese2	ISO_2022,locale=ja,version=0
SHIFT_JIS	IBM-943_P14A-2000
Simplified Chinese	GB18030
Traditional Chinese (Taiwan)	EUC-TW
UTF8	UTF-8
UTF16BE	UTF-16BE (BigEndian)
UTF16LE	UTF-16LE (LittleEndian)
UTF16	UTF-16
UTF-32	UTF-32
ISO88592	IBM-912
WinLatin1	IBM-5348
WinLatin2	IBM-5346
EBCDICJapKata	IBM-290
EBCDIKJapKanji	IBM-930
IBM943	IBM-943_P130-2000
EBCDICJapKanji	IBM-939
EBCDIKJapKanji83	IBM-930-83
EBCDIKJapKanji78	IBM-930-78
EBCDICJapKanji83	IBM-939-83
EBCDICJapKanji78	IBM-939-78
BASESHIFT_JIS	IBM-943_exSJIS_base

#### Encodings supported by Exstream, continued

Encoding	ID String
GBK_CP936	IBM-1386
JEF930	JEF-930
JEF939	JEF-939
ISO88591	ISO-8859-1

## 2.28.2 Producing UTF-8 XML (Content) Output

When working with SBCS applications, you can specify UTF-8 as the encoding for your XML data files if you have the XML (content) Output module. You must follow specific guidelines in your design, however.

For more information on designing for UTF-8 XML output, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

## 2.29 Setting Up Multi-Channel XML Output

This section discusses the following topics:

- “About Multi-Channel XML Output” on the next page
- “Creating Basic Multi-Channel XML Output” on page 172
- “Specifying How Table Cells are Separated in Plain Text Multi-Channel XML Output” on page 176
- “Tagging Objects for Multi-Channel XML Output with Metadata” on page 177
- “Setting Up to Create Transformed Output” on page 182
- “Applying an Encoding to Multi-Channel XML Output (DBCS)” on page 184

In addition to the topics discussed in this section, you can also use the following basic setup option for Multi-Channel XML output:

Setup option	See this section for more information
Specifying the resolution for your output	“Managing output resolution in Exstream” on page 37
Assigning a connector for testing and production	“Assigning a Connector to an Output Object for Testing and Production” on page 45

Furthermore, you can also use the following setup option that is shared with HTML output:

Setup option	See this section or guide for more information
Specifying fallback font options for fonts that may not be supported in some browsers	<a href="#">"Specifying Fallback Fonts for HTML, HTML (Email), or Multi-Channel XML Output" on page 114</a>

## 2.29.1 About Multi-Channel XML Output

Multi-Channel XML output is XML that can be transformed for delivery to different media channels (for example, Short Message Service (SMS) messages, social media posts, and VoiceXML-based speech applications). To transform the Multi-Channel XML output from Exstream into output that is ready for these media channels, you provide an Extensible Stylesheet Language (XSL) file. After Exstream creates Multi-Channel XML output, the output file is sent along with the XSL file to the embedded XSL Transformation (XSLT) engine during the production run. The transformed output produced from the XSLT engine can be in many different media formats, as determined by the XSL file you provide.

By default, Multi-Channel XML output includes all design objects that are in your application, including text boxes, images, tables, and so on. If you choose to use the embedded XSLT engine to create transformed output, you can use unique object attributes (for example, the object name) in your XSL file to choose the objects to include in the transformed output. This is the fastest way to create Multi-Channel XML output and transformed output from an existing application.

You can also use the metadata feature in Exstream to label (or tag) design elements in the application. This method lets you easily manage objects in an application for Multi-Channel XML output and create a shorter, more streamlined final output file.

For more information about using metadata for Multi-Channel XML output, see ["Tagging Objects for Multi-Channel XML Output with Metadata" on page 177](#).

Multi-Channel XML output supports the inclusion of hyperlinks in the output file. You can design hyperlinks on pages for specific text or a paragraph within a text box. You can also assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The link addresses can be static or dynamic (set by variable values). Keep in mind that an end user cannot follow hyperlinks in Multi-Channel XML output. However, the hyperlink address is visible when an end user hovers the pointer over the text, image, or shape that includes the hyperlink.

For more information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

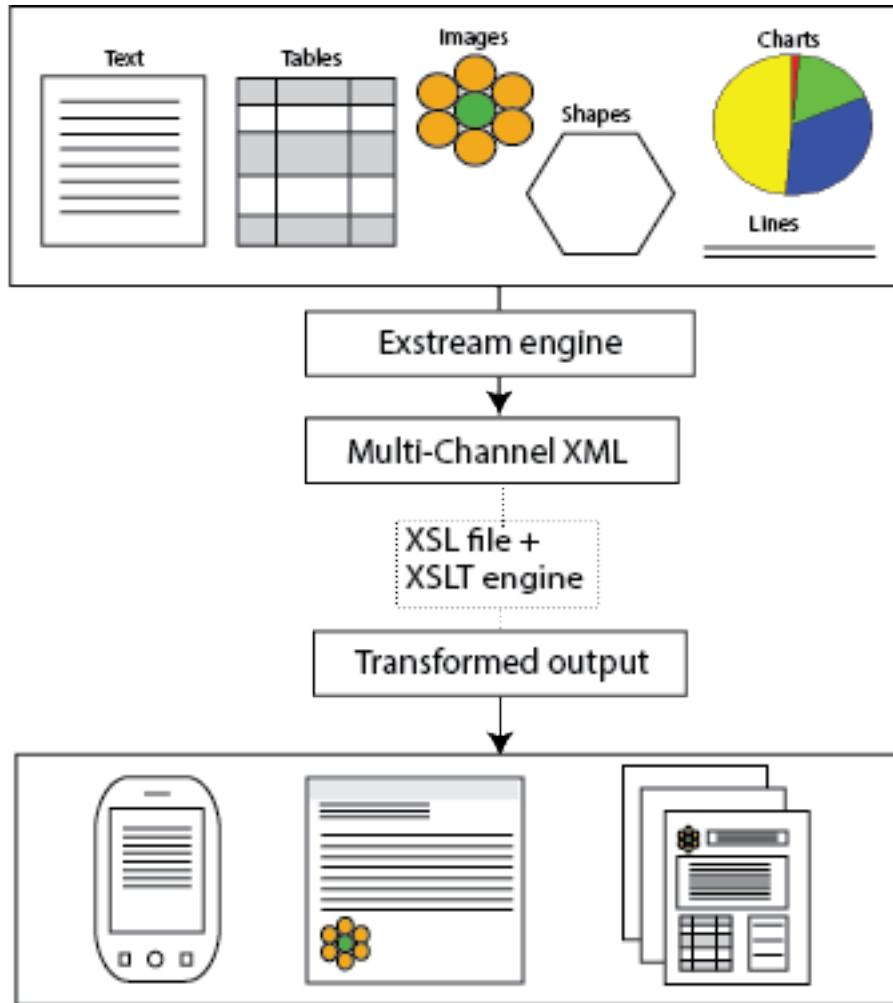
Objects that can be exported to Multi-Channel XML output include the following:

List of objects that can be exported to Multi-Channel XML output

Program	Objects included in Multi-Channel XML output
Design Manager	You can include the following objects in Multi-Channel XML output: <ul style="list-style-type: none"><li>• Documents</li><li>• Pages</li><li>• Paragraphs</li><li>• Messages</li><li>• Barcodes</li><li>• Library components</li></ul>
Designer	You can include all design objects, except for frame, TOC, and index objects in Multi-Channel XML output.  Keep in mind the following design consideration:  For table objects, repeating footers behave differently in Multi-Channel XML output. In Exstream, repeating footers are intended to repeat at the bottom of the table each time the table breaks across a page or frame. However, in Multi-Channel XML output, tables are intended to have different breaks and flows depending on how the content is delivered to customers. For example, the same content will have different breaks when delivered to the small screen of a phone or personal device than when delivered to the web. To allow tables that are in Multi-Channel XML output to accommodate for different breaks and flows, repeating footers are always moved to the end of the table in the final output.

Keep in mind that Multi-Channel XML output does not take flow into consideration. This means that flow pages are not applicable and do not occur in the output. Everything that falls on flow pages is processed and placed with the original design page.

The following graphic illustrates the basic process for producing Multi-Channel XML output:



## 2.29.2 Creating Basic Multi-Channel XML Output

When you create basic Multi-Channel XML output, you must set your preference for the object format (HTML or plain text) and specify whether you want to include CDATA tags in the final output.

Objects in Multi-Channel XML output can be produced in either plain text or HTML format. If you create Multi-Channel XML output in plain text format, style sheet information is not included in the output file. However, if you specify HTML format, Exstream includes HTML tags in the Multi-Channel XML output. HTML tags let you preserve formatting and style information, including style sheet information. Only the style sheet definitions that are referenced by the objects that are exported to output are included in the output. Style definitions are listed at the top of the Multi-Channel XML.

Depending on the text used in your output, you can also control whether CDATA tags are included in the output. By default, CDATA tags are included to preserve special characters in

the text (such as &, \$, or \*). If CDATA tags are not included in the output, all special characters are automatically converted to the XML code equivalent (for example "&" is automatically converted to "&"). Additionally, if your XSLT has difficulty accessing content (such as text in an individual table cell or individual pieces of content with large amounts of formatting), you might gain better access to this content by not including CDATA tags in the output.

If you are using the optional built-in embedded transform processor to create Multi-Channel XML output on the z/OS platform, you must first download and install the "XML Toolkit for z/OS" from the IBM website. Then, you must point your engine to run Job Control Language (JCL) to the installed Partitioned Data Set Extended (PDSE). If you are not using the built-in embedded transform processor, then you do not need the toolkit.

When you produce Multi-Channel XML output, keep in mind the following considerations:

- The embedded XSLT engine in Exstream supports XSLT version 1.0.
- If you run the 64-bit version of the Exstream engine on Windows, you cannot use the embedded XALAN XSL transform processor.
- To transform Multi-Channel XML output produced by the 64-bit version of the Exstream engine on Windows, use an alternative transform processor during post-processing.

**Note:** When you produce Multi-Channel XML output, the z/OS engine does not support creating external files. Because all non-text-based objects (such as barcodes or charts) become external image files in the output, Multi-Channel XML output on z/OS is limited to text-based objects only. However, you can configure images on a design page so that they are not generated in the output but instead point to an external file path or URL location.

To create a basic Multi-Channel XML output:

1. If you do not know the application mode of the database in which you are working, review the **Application mode** drop-down list on the **Workflow** tab of the **System Settings** object.

For information about the application mode of a database, see *System Administration* in the Exstream Design and Production documentation.

2. Based on the application mode of the database in which you are working, do one of the following:

For this application mode	Do this
SBCS	<ol style="list-style-type: none"><li>In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs</b>.</li><li>Right-click the <b>Outputs</b> heading and select <b>New Output</b>. The <b>New Output</b> dialog box opens.</li></ol>

For this application mode	Do this
DBCS	<p>a. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs</b>.</p> <p>b. Right-click the <b>Outputs</b> heading and select <b>New Output</b>.</p> <p>The <b>New Output</b> dialog box opens.</p>
SBCS/DBCS	<ul style="list-style-type: none"> <li>• To create SBCS output, complete the following steps:           <ul style="list-style-type: none"> <li>i. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs &gt; SBCS Outputs</b>.</li> <li>ii. Right-click the <b>SBCS Outputs</b> heading and select <b>New Output</b>.</li> </ul> <p>The <b>New Output</b> dialog box opens.</p> </li> <li>• To create DBCS output, complete the following steps:           <ul style="list-style-type: none"> <li>i. In Design Manager, in the Library, go to <b>Environment &gt; Delivery &gt; Outputs &gt; DBCS Outputs</b>.</li> <li>ii. Right-click the <b>DBCS Outputs</b> heading and select <b>New Output</b>.</li> </ul> <p>The <b>New Output</b> dialog box opens.</p> </li> </ul>

3. In the **Name** box, enter a name for the output device.

4. In the **Description** box, enter a description (optional).

5. Click **Finish**.

The output object opens in the Property Panel for you to define.

6. From the **Driver** drop-down list, select **XML (multi-channel)**.

7. From the **Object output format** drop-down list, select one of the following options:

To	Do this	Behaviors to keep in mind
Create output that you can transform into a channel that is optimized for plain text, such as SMS messages or plain text email communication	Select Plain text.	<p>When you use the plain text option for Multi-Channel XML output, keep in mind the following:</p> <ul style="list-style-type: none"> <li>For text objects, all style and formatting options are removed.</li> <li>Images in the output will be represented by the image name and, depending on the output settings, the path to the image.</li> </ul> <p>For more information about setting up images for Multi-Channel XML output, see “<a href="#">Referencing and Storing Images for HTML, HTML (Email), and Multi-Channel XML Output</a>” on page 244.</p> <ul style="list-style-type: none"> <li>For table objects, the content of each cell is followed by a table cell separator, and each row breaks to a new line.</li> </ul> <p>For more information about table cell separators used in plain text Multi-Channel XML output, see “<a href="#">Specifying How Table Cells are Separated in Plain Text Multi-Channel XML Output</a>” on the next page.</p>
Create output that you can transform into a channel optimized for the use of images and hyperlinks, such as MMS messages	Select HTML.	When you apply the HTML object format, the Multi-Channel XML output is similar to the output that the HTML output driver produces in Exstream.

8. To control the inclusion of CDATA tags in the output, do one of the following:

To	Do this
Include CDATA tags in the output (default)	Select the <b>Include CDATA tags</b> check box.
Not include CDATA tags in the output	Clear the <b>Include CDATA tags</b> check box.

9. Select the type of designs that you want to deliver in the output:

To	Do this
Deliver standard designs in the output (default)	Clear the <b>Use container design</b> check box.
Deliver container designs in the output	<ol style="list-style-type: none"> <li>Select the <b>Use container design</b> check box.</li> <li>In the <b>Container design</b> label box, click  and select the name of the container design that you want to deliver in the output.</li> </ol>

For more information about working with container designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

10. From the Menu bar, select **Edit > Save**.

## 2.29.3 Specifying How Table Cells are Separated in Plain Text Multi-Channel XML Output

When tables are delivered as plain text, the table rows appear as lines of text, rather than appearing as a fully drawn table. For each row, the cell contents are appended together in a single line, and the contents of each cell are separated by a specific table cell separator. By default, the table cell separator is a tab (a series of spaces used to align content). Optionally, you can define a custom set of characters to act as a table cell separator. A custom table cell separator can be useful if you are transforming Multi-Channel XML content using an XSL file. For example, you can use the table cell separator as a key term to identify where the content for each table cell begins, in order to get better transform access to cell content. If you use a custom separator to give the XSL file better access to content, the character(s) you use as a separator must not appear within any of the table content.

To specify how table cells are separated in plain text Multi-Channel XML output:

1. In Design Manager, from the Library, drag an XML (multi-channel) output object to the Property Panel.

The output properties open in the Property Panel for you to define.

2. Click the **Basic** tab.
3. From the **Table cell separator** drop-down list, select one of the following options:

To	Do this
Use a tab as a table separator (default)  For example:  Lorem Epsom Delorum 1 2 3	Select <b>Tab</b> .
Use a custom table cell separator  For example, if you use an & as a custom cell separator, the table would appear as follows:  Lorem&Epsom&Delorum 1&2&3	<ol style="list-style-type: none"><li>Select <b>Custom</b></li><li>In the adjacent box, enter the character(s) you want to use as a separator. You can enter a maximum of 32 SBCS or DBCS characters.</li></ol>

4. From the Menu bar, select **Edit > Save**.

## 2.29.4 Tagging Objects for Multi-Channel XML Output with Metadata

You can use the metadata feature in Exstream to label (or tag) design elements in the application. When you use metadata to tag objects for Multi-Channel XML output, metadata functions as a descriptive attribute that you can use in an XSL file as a means of labeling and identifying the objects that you want to include in your final output. As a simple example, you can apply the same metadata (for example, "InfoMessages" for a messaging campaign) to multiple objects in an application that you want to include in your final output. When you set up your XSL file, you can include that metadata in your XSL file to pull all the objects with the "InfoMessages" metadata into the final transformed file. This method lets you easily manage objects in an application for Multi-Channel XML output and create a shorter, more streamlined final output file.

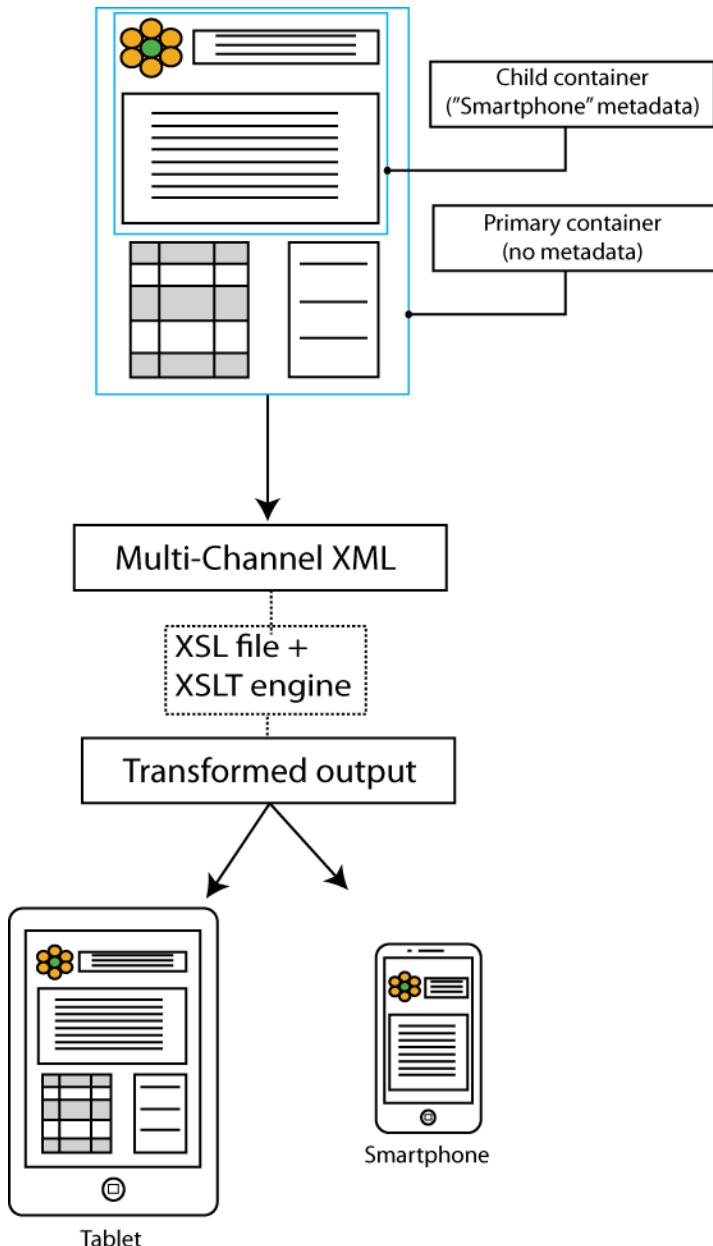
This section discusses the following topics:

- ["About Metadata Tags for Multi-Channel XML Output" below](#)
- ["Assigning Metadata to Individual Objects" on page 179](#)
- ["Assigning Metadata to Groups of Objects Designed for the Same Output" on page 180](#)
- ["Including Only Objects with Metadata in Multi-Channel XML Output" on page 181](#)

### About Metadata Tags for Multi-Channel XML Output

In a standard design, you must tag individual objects in order to group them using metadata. However, in a container design, you can use the parent/child relationship between containers and the objects inside them to visually group objects and more easily assign metadata. For example, suppose you are using Multi-Channel XML output to create two sets of HTML output—one for viewing on a tablet, and one for viewing on a smartphone. For larger tablet screens, you include all of the objects in your design. For smaller smartphone screens, however, you want to include only a subset of the objects in your design. You can use containers to group objects that will be included in the smartphone output. You can then create a metadata object (such as "Smartphone") and assign it to those containers. When you create your XSL file, you can use the "Smartphone" tag to include the smartphone containers and specifically exclude the other containers from your smartphone output.

The following graphic gives an overview of how you can use metadata within the same container design to create output for both a tablet and a smartphone.

**Example of using metadata to create different types of transformed output**

In the previous example, you use a single container design to create outputs for multiple screen sizes. If you want to use containers as a means of previewing how different designs will look on different screen sizes, you must use separate pages to create each container design. For example, if you want to see how a design looks on a particular smartphone as you are designing, you must create a page with a container design that approximates the size of that smartphone screen. If you are also designing for a particular tablet, you must then create a separate page with a container design that approximates the size of a tablet screen.

For more information about using containers in a design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Whether you tag multiple objects with a single metadata object, or a single object with multiple metadata objects, you should use tags that clearly identify the object. For example, suppose you want to create transformed output to send messages to customers with overdrawn accounts. You can use the metadata object named "InfoMessages" and apply the metadata to a text box, which contains static text and variable text. Additionally, you can use a metadata object named "TransactionTable" and apply this metadata to a table showing a customer's monthly transactions. Both metadata objects can be applied to all customer applications with overdrawn accounts and sent to a downstream process for delivery.

When the designer in charge of distributing the application to customers in various deliverable types receives the application, he or she can then use the "InfoMessages" or "TransactionTable" metadata objects to create another metadata object for customers according to their preferred deliverable type. For example, the developer can create a new metadata object named 'SMS\_Overdrawn' and apply the metadata object to the same text box to which you applied the "InfoMessages" metadata object. In addition, the developer can create a new metadata object named 'MMS\_Overdrawn' and apply the metadata object to the same table to which you applied the "TransactionTable" metadata object.

## Assigning Metadata to Individual Objects

You can apply metadata to any object in Design Manager except System Settings, and to any object in Designer except frames. Before you can assign metadata to an object, you must first create the metadata objects. Keep in mind, however, that not all objects can be exported to Multi-Channel XML output, even if metadata is assigned to those objects.

For more information about creating metadata and using the metadata feature, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about objects that can be exported to Multi-Channel XML output, see ["About Multi-Channel XML Output" on page 170](#).

This section discusses the following topics:

- ["Assigning Metadata to a Library Object in Design Manager" below](#)
- ["Assigning Metadata to an Object in Designer" on the next page](#)

### Assigning Metadata to a Library Object in Design Manager

If you use metadata to tag objects in your standard design, you must tag each object separately. Although you can use a single metadata tag to group like objects, you must assign metadata to each object in the group. If you want to group objects visually for electronic output, you should use a container design.

For more information about using containers in a design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To assign metadata to a library object in Design Manager:

1. In Design Manager, in the Library, right-click the object to which you want to add metadata and select **Metadata**.

The **Metadata** dialog box opens.

2. In the **Available Metadata** list, select a metadata object and click  to apply it to the object. Repeat this step as necessary.

3. Click **OK**.

The **Metadata** dialog box closes and the metadata is applied to the object.

4. If you applied metadata to a folder, the **Apply Metadata** dialog box opens. Select an option for how to apply metadata to the folder and its objects, and click **OK**.

## Assigning Metadata to an Object in Designer

1. In Designer, select the object to which you want to assign metadata.

2. Click .

The object properties dialog box opens.

3. Click the **Metadata** tab.

4. In the **Available Metadata** list, select a metadata object and click  to apply it to the object. Repeat this step as necessary.

5. Click **OK**.

The object properties dialog box closes.

6. From the Menu bar, select **File > Save**.

## Assigning Metadata to Groups of Objects Designed for the Same Output

You can use containers to more efficiently assign metadata to groups of objects. In contrast to a standard design, which requires that you group objects by assigning the same metadata tag to each object in the group, a container design lets you assign metadata to a container and then use the hierarchy in the resulting Multi-Channel XML to group the objects that are inside that container. For example, suppose you create a design and want to include only certain objects in the HTML output that you produce for a smartphone. You can create a metadata object named "Smartphone" in either Designer or Design Manager. You can then place all objects that are intended for output to smartphones in a single container and assign the "Smartphone" metadata object to the container. You can then include the "Smartphone" tag in your XSL file so that only that container (and its contents) are included in the smartphone output. If you use a standard design, however, you must assign the "Smartphone" metadata object to each object that you

want to include in the smartphone output, and then specify the proper hierarchy in your XSL file. In either case, you can code your XSL file so that it includes all objects for the tablet output.

For more information about including only objects with metadata in your output, see “[Including Only Objects with Metadata in Multi-Channel XML Output](#)” below.

When producing output from a container design, it is not necessary to use metadata in your XSL file to include or exclude objects. You can also use the hierarchy of the XML itself in your XSL file to differentiate between the containers and objects that you want to include in your output.

For more information about using containers in a design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## Assigning Metadata to a Container in Designer

1. In Designer, open a container design and select the container to which you want to assign metadata.

2. Click .

The **Container Properties** dialog box opens.

3. Click the **Metadata** tab.

4. In the **Available Metadata** list, select a metadata object and click  to apply it to the object.

5. Click **OK**.

The **Container Properties** dialog box closes.

6. From the Menu bar, select **File > Save**.

## Including Only Objects with Metadata in Multi-Channel XML Output

A typical application contains many design objects, only some of which you might want to include for Multi-Channel XML output. If all the design objects you want to include in the transformed output can be identified by means of metadata, you can choose to include in the Multi-Channel XML output only objects that have metadata applied to them. Including only metadata objects will reduce the size of your Multi-Channel XML output file.

All objects contained within a metadata object are included in Multi-Channel XML output, even if metadata is not applied to those objects. For example, if you add metadata to a page object, all applicable objects on that page are included in a hierarchy in the same content block in the output. In addition, if you add metadata to a table, anything embedded within that table (such as a text box) will be also be included in Multi-Channel XML output.

Keep in mind, however, that objects that are not explicitly assigned a metadata object will not have a metadata tag in Multi-Channel XML output. If you want to use metadata as the only means in your XSL file to specify objects to include in transformed output, you must apply

metadata to each object you want to include. For example, suppose you add "Met2" to a page object, but you do not add "Met2" to a text box (named "Text1") on that page. Although the text box object will be exported to the Multi-Channel XML file, it will not have the "Met2" metadata object applied to it. To include the text box in the transformed output using only "Met2," you must apply "Met2" to the text box.

The following example shows Multi-Channel XML output that includes a page tagged with "Met2" and the objects on that page. Notice that the objects on the page, including a text box and a graphic, are not tagged with "Met2" in the output:

#### Example of Multi-Channel XML output with metadata

```
<dlg:Page name="Page">
<dlg:Metadata name="Met2" value="A0167665" />
<dlg:Metadata name="Met3" value="FreeForm" />
<dlg:TextBox name=Text1">
    <dlg:ComponentContent>
        <![CDATA[This text is in the text box.]]>
    </dlg:ComponentContent>
</dlg:TextBox>
<dlg:Graphic name="Shape">
    <dlg:ComponentContent>
        <![CDATA[graphic1.gif]]>
    </dlg:ComponentContent>
</dlg:Graphic>
</dlg:Page>
```

To include only objects with metadata in Multi-Channel XML output:

1. In Design Manager, from the Library, drag an XML (multi-channel) output object to the Property Panel.
2. Click the **Basic** tab.
3. To include only objects that have metadata applied to them, select the **Include only objects with metadata in the output** check box. If you clear the **Include only objects with metadata in the output** check box, then all design objects will be included in your output.
4. From the Menu bar, select **Edit > Save**.

## 2.29.5 Setting Up to Create Transformed Output

The XSLT engine is embedded in the Exstream engine. If you want to create transformed output, you can enable the embedded XSLT engine on the properties of an XML (multi-channel) output driver. If enabled, the XSLT engine automatically runs using the Multi-Channel XML output file created by the Exstream engine. You provide an XSL file that is used with the embedded XSLT engine to transform the Multi-Channel XML output that is produced by Exstream into a transformed file that can be converted into multiple media formats.

When you attach an XSL file to an XML (multi-channel) output driver, you can easily open and view the XSL file from Design Manager. If you are using metadata to create Multi-Channel XML output, you can create a metadata report to help you make sure that the metadata names that are used in the design are consistent with the metadata names that are used in the XSL file.

For more information about metadata reports, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To set up to create transformed output:

1. In Design Manager, from the Library, drag an XML (multi-channel) output object to the Property Panel.

The output properties open in the Property Panel for you to define.
  2. Click the **Basic** tab.
  3. Select the **Run XSLT engine on output** check box.
  4. To use a variable to specify a dynamic XSL file, select a variable in the **XSL file location variable** box.
  5. To test output before production, enter the path to the XSL file you want to use for testing in the **Test XSL file location** box. Alternatively, you can click  and browse to the XSL file that you want to use.
  6. If you want to view the XSL file, click .
- The XSL file opens in Notepad.
7. In the **Production XSL file location** box, enter the path to the XSL file that you want to use.
  8. From the Menu bar, select **Edit > Save**.

## Creating Transformed Output on z/OS

If you are using the XML (multi-channel) output driver with a z/OS mainframe engine, Exstream writes the Multi-Channel XML output to an interim file before the XSLT engine runs. Before you run the Exstream engine, therefore, you must allocate a temporary file ( DD: TMPMULTI). The space allocated for your temporary file must be at least as large as your main file.

If you are using the optional built-in embedded transform processor to create Multi-Channel XML output on the z/OS platform, you must first download and install the "XML Toolkit for z/OS" from the IBM website. Then, you must point your engine to run Job Control Language (JCL) to the installed Partitioned Data Set Extended (PDSE). If you are not using the built-in embedded transform processor, then you do not need the toolkit.

**Note:** When you produce Multi-Channel XML output, the z/OS engine does not support creating external files. Because all non-text-based objects (such as barcodes and charts) become external image files in the output, Multi-Channel XML output on z/OS is limited to text-based objects only. However, you can configure images on a design page so that they are not generated in the output but instead point to an external file path or URL location.

## Creating Transformed Output on UNIX and Linux

If you are using the embedded XSLT engine with the XML (multi-channel) output driver on UNIX or Linux, you must set up the environment for the platform you are using. The library path environment variable must contain the path to the directory that contains the ExEngTrans library. The actual directory path is dependent on where you have installed Exstream Design and Production.

The following table lists the environment variables you must set up to use the XSLT engine, depending on your platform:

Environment variables to set up on UNIX/Linux

Platform	Environment variable
HP-UX	SHLIB_PATH
IBMAIX	LIBPATH
SuSE	LD_LIBRARY_PATH
Red Hat	LD_LIBRARY_PATH
Sun Solaris	LD_LIBRARY_PATH

## 2.29.6 Applying an Encoding to Multi-Channel XML Output (DBCS)

For DBCS applications, you can apply an XML encoding (or the default code page). You must set the properties of the encoding to allow it to be used for output files. By default, Multi-Channel XML output is created with UTF-8 encoding. When you apply an encoding other than the default, this encoding is applied only to Multi-Channel XML output. If you want to use any encoding other than UTF-8 in the transformed output file, you must specify that encoding in the XSL file.

For more information on encoding, see *System Administration* in the Exstream Design and Production documentation.

The encoding you specify is displayed in the XML header.

For example:

```
<?xml version="1.0" encoding="utf-8" ?>
```

To apply an encoding to XML output:

1. In Design Manager, from the Library, drag an XML (multi-channel) output object to the Property Panel.
2. Click the **Basic** tab.
3. Specify an encoding in one of the following ways:
  - Click the **Encoding** box and select an encoding.
  - From the Library, drag an encoding to the **Encoding** box.
4. From the Menu bar, select **Edit > Save**.

## 2.30 Setting Up ZPL Output

You can use the following basic setup options for ZPL output:

Setup option	See this section for more information
Creating an output object	<a href="#">"Creating an Output Object" on page 34</a>
Specifying the resolution for your output	<a href="#">"Managing output resolution in Exstream" on page 37</a>
Assigning a connector for testing and production	<a href="#">"Assigning a Connector to an Output Object for Testing and Production" on page 45</a>

# Chapter 3: Optimizing output production and enhancing output files

This chapter discusses the following topics related to optimizing and enhancing your output:

- **Optimizing the size and structure of the print stream**—This section discusses settings for many output drivers that can reduce the size of the output file and reduce run time.
- **Optimizing text quality**—This section discusses multiple ways you can improve how text appears in your output.
- **Optimizing and enhancing output for each output driver**—You must keep in mind certain considerations for each output driver as you design your application and set up for output. These sections discuss information you can use to produce the highest quality output for your business needs. These sections also discuss various features in Exstream that you can use to enhance your output. Not all output drivers are discussed individually in this section.

For more information about setting up your output driver, see “[Setting Up Output Properties](#)” on [page 19](#).

For information about managing resources, see “[Managing Resources for the Print Stream](#)” on [page 229](#).

## 3.1 Optimizing the size and structure of the print stream

For some output drivers, Exstream gives you some control over how objects are grouped, which can potentially reduce the print file size and the time it takes the engine to run. You can specify options that can affect the memory usage, processing speed, and resource demands.

To set print stream optimization options:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Basic** tab.

3. From the **Print stream optimization** drop-down list, select one of the following options, depending on the output driver:

Output Driver	Options available
AFP	<ul style="list-style-type: none"> <li>• <b>None: recommended if overlapping objects</b>—Use this option if you have overlapping text and graphics or if you are using layers. This is the slowest option.</li> <li>• <b>Medium: group fonts only</b>—This option processes text in a group. This is not the best option if you are using layers.</li> <li>• <b>High: May cause problems with overlapping objects</b>—This option groups all objects. This is the fastest option, but you should not select this option if you are using design layers or have designed objects so that they overlap.</li> </ul> <div data-bbox="616 656 1334 747" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> The Print stream optimization drop-down list is inactive when you use the knockout feature.</p> </div> <p>For more information about the knockout feature, see “<a href="#">Using Knockout to Prevent the Mixing of Colors in Overlapping Objects</a>” on page 67.</p> <p>For more information about design layers, see <i>Designing Customer Communications</i> in the Exstream Design and Production documentation.</p>
IJPDs	<ul style="list-style-type: none"> <li>• <b>None: leave all data uncompressed (minimize Engine run time)</b>— Exstream does not compress any data.</li> <li>• <b>Medium: compress text characters (higher processing speed)</b>— Exstream compresses text characters during packaging.</li> <li>• <b>High: compress all bitmaps (maximum Output processing speed)</b>— Exstream compresses bitmaps during packaging and when you run the engine. This option can slow engine processing speed, depending on the amount of images and linearized objects in the output.</li> </ul>
Metacode	<ul style="list-style-type: none"> <li>• <b>None: recommended for most Metacode printers</b>—For most Metacode printers, this is the best option. The other options require that you use fast online printers.</li> <li>• <b>Medium: consolidate text records (fast online printers ONLY)</b>—If you are using fast online printers, select this option to consolidate text records but not graphics.</li> <li>• <b>High: consolidate all records (fast online printers ONLY)</b>—If you are using fast online printers, select this option to consolidate both text and graphics.</li> </ul> <p>When you select <b>Medium</b> or <b>High</b>, the engine sends information into consolidated records of up to 30,000 bytes each (or the limit you specify in the <b>Print stream blocking method</b> drop-down list, also on the <b>Basic</b> tab). This process enables most fast online printers to operate at rated speeds.</p> <div data-bbox="616 1607 1334 1712" style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Note:</b> If the cover pages print, but the document pages do not, and the Metacode printer console issues an error of Too much data on the page, then your print stream optimization setting is too low. Change the print stream optimization to a higher setting.</p> </div>

Output Driver	Options available
MIBF	<ul style="list-style-type: none"> <li><b>None: Leave all data uncompressed</b>—Exstream does not compress any data.</li> <li><b>High: compress all graphics and on page images</b>—Exstream compresses graphics and images during packaging and when you run the engine. This option can affect engine processing speed, depending on the number of images in the output. Graphics and on-page images are compressed with run length compression to decrease the file size and enhance printing performance.</li> </ul>
PPML	<ul style="list-style-type: none"> <li><b>Use disposable resources</b>—This option creates resources that are deleted when the print job is completed. Use this option when each print job going to the printer is unique.</li> <li><b>Use global resources (not deleted after each job)</b>—This option creates resources that are then saved when the print job is completed. Use this option when similar print jobs are repeatedly run on the same printer.</li> </ul>
VIPP	<ul style="list-style-type: none"> <li><b>Make "thin" VIPP (using external resource files)</b>—This option uses resources (images and forms) as separate files the printer can RIP and store for reuse with many print files. This saves both engine processing time and print production time. Resources are placed in the same directory as the composed file.</li> <li><b>Make "thick" VIPP</b>—This option embeds all resources at the top of the print stream and makes each print file a complete job. This version of VIPP output cannot reference external resources. If you are producing output on z/OS platforms, you must use this option.</li> </ul>
VPS	<ul style="list-style-type: none"> <li><b>Make "thin" VPS (using external resource files)</b>—This option uses resources (images and forms) as separate files the printer can RIP and store for reuse with many print files. This saves both engine processing time and print production time. Resources are placed in the same directory as the composed file.</li> <li><b>Make "thick" VPS</b>—This option embeds all resources at the top of the print stream and makes each print file a complete job. This version of VPS output cannot reference external resources. If you are producing output on z/OS platforms, you must use this option.</li> </ul>

## 3.2 Optimizing text quality for output

This section discusses the following topics:

- “[Making sure fonts appear correctly in output](#)” on the next page
- “[Enhancing character print quality](#)” on page 190
- “[Text quality considerations in multiple queue production](#)” on page 192

### 3.2.1 Making sure fonts appear correctly in output

Exstream pulls metrics, such as height and width, from the font. By default, the engine uses integer measurements (which are used most often by bitmap fonts); however, rounding errors can cause the cursor position in your output to be different from what you expected (especially if you use outline fonts). Usually, this issue can be corrected by increasing the output resolution. If this solution does not work, you can adjust the character fidelity.

For best results, use the default settings for character positioning. If you are experiencing issues with overlapping text or incorrect spacing between words or characters, try adjusting your resolution settings first. For PostScript printers, try using Type1 fonts. If you are using multiple output queues for production, some character fidelity settings might be affected.

For more information about resolution settings, see [“Managing output resolution in Exstream” on page 37](#).

For more information about Type1 fonts, see *System Administration* in the Exstream Design and Production documentation.

For more information about text considerations for multiple queue production, see [“Text quality considerations in multiple queue production” on page 192](#).

The engine matches font metrics, so it is rare that you must select anything other than the default setting. Select one of the other options only when necessary, since they can adversely affect processing times, memory requirements, and the size of the output file.

You can adjust the character fidelity settings for the following output drivers:

- IJPDS
- Metacode
- MIBF
- PCL
- PDF
- PDF/VT
- PostScript
- PPML
- TOP
- VDX
- VIPP
- VPS

To adjust character fidelity settings:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Advanced** tab.
3. In the **Character fidelity** list, select one of the following options to specify how text is placed:
  - **Auto, character**—This option is the default. Use this setting unless you are having issues with overlapping text or incorrect spacing in your output.
  - **Auto, word**—Adds positioning commands for each word in the output only if required; otherwise, positioning commands are added for each line of text. This option should allow you to set a lower print resolution, which reduces the size of images in output.
  - **Auto, line**—Adds positioning commands for each line of text and automatically positions text within that line. This option produces the smallest output size.
  - **Position every word**—Applies placement logic to every word in the output, which increases print sizes. This setting forces Exstream to apply a new x/y coordinate for every word, rather than one x/y coordinate per line of text.
  - **Position every character**—Applies placement logic to every character in the output, which significantly increases print sizes. This setting forces Exstream to apply a new x/y coordinate for every character, rather than one x/y coordinate per line of text. If you choose this option for any output driver in a multiple queue production run, the setting is applied to the text on all charts (for example, data labels).
4. If you want Exstream to add placement logic to characters in the output that are smaller than a specific point size, enter a value (in tenths of a point) in the **Position every character below** box. If you enter a value other than the default value of 0, you receive a message when you package the application. This option is not available if you select **Position every character** in the **Character fidelity** list.

### 3.2.2 Enhancing character print quality

#### Overview

Exstream pulls metrics, such as height and width, from the font. By default, the engine uses integer measurements (which are used most often by bitmap fonts); however, rounding errors can cause the cursor position in your output to be different from what you expected, especially if you use outline fonts. Normally, the difference can be ignored or corrected by adjusting the character fidelity.

Two features, text position calculation and text position adjustment, work together to control how the engine determines the size of each character and any adjustments that must be made based on the size. These features help improve consistency when working with multiple output queues that have different requirements.

You can change how the engine calculates output cursor positions in the following ways:

- **Calculating character widths for text positioning**—Depending on the type of font your output device uses and how tightly you must control character placement, you can change the way the engine calculates character width. You can change this setting with the text position calculation feature. Keep in mind that if you are producing high-volume output with multiple output queues, the text position calculation setting on the first output queue in the application applies to all subsequent output queues.
- **Accommodating font differences between output drivers**—If you are using more than one output queue, you can adjust the text position with the text position adjustment feature to accommodate for font differences between the output drivers that are associated with those output queues. You can use the text position adjustment feature to make output that uses an outline font look like output that uses bitmap fonts.

You use these options for the following output drivers:

- AFP
- IJPDS
- Metacode
- MIFB
- PCL
- PDF
- PDF/VT
- PostScript
- PPML
- TOP
- VDX
- VIPP
- VPS

## Calculating character widths for text positioning

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Advanced** tab.
3. In the **Text position calculation** list, select one of the following options:
  - **Use native PDL font metrics**—Exstream calculates character width using the resolution you set on the **Basic** tab.

- **Use outline font metrics**—Exstream calculates character width using a high resolution that enhances the look of outline fonts when produced as output. This option ensures that the appearance of the text in the final output is consistent, even if one output driver uses outline fonts and one output driver uses bitmap fonts.

## Accommodating font differences between output drivers

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Advanced** tab.
3. In the **Text position adjustment** list, select one of the following options:
  - **No adjustment**—Text output is not changed.
  - **Adjust at word boundaries**—Exstream inserts position commands as necessary at word boundaries.
  - **Adjust characters**—Exstream inserts position commands as necessary on a per-character basis. Text is not searchable when you select **Adjust characters**.

## Solving problems with output queues

Occasionally, when there are lines of text with varying fonts and styles, rounding differences in the output can cause character spacing issues. If you experience any character fidelity issues in your output, try the following troubleshooting tips in order:

1. Increase the [output resolution](#).
2. If you are using [outline font metrics](#), switch to [native font metrics](#) or vice versa.
3. Change the [text position adjustment](#).
4. Change the [character fidelity settings](#).

### 3.2.3 Text quality considerations in multiple queue production

When producing high-volume documents, you often use multiple output queues, some of which are associated with different types of output drivers. When you set up for multiple queue production, keep in mind the following:

- If you select the **Use first queue output character fidelity settings for all queues** check box in the application properties, and the character fidelity settings do not match for all output

drivers in one application, then the settings for the first output queue in the application are applied to the subsequent output queues, and packaging continues.

For more information about output queues, see “[Controlling and Setting Up High Volume Production Options with Output Queues](#)” on page 315.

- If you design an application that contains static text, keep in mind that character fidelity settings are controlled on a per-queue basis. For example, if the character fidelity settings in the output driver for a secondary queue differ from those on the output driver for the primary queue, the engine honors the settings for the secondary queue.

For more information about character fidelity, see “[Making sure fonts appear correctly in output](#)” on page 189.

- The **Text position calculation** setting on the first output queue in the application applies to all subsequent output queues.

For more information about text position calculation, see “[Enhancing character print quality](#)” on page 190.

- When you assign the same type of output driver to multiple queues in an application (for example, if you are using two PDF-driven output queues), you must use the same font type (for example, all bitmap fonts or all outline fonts) for all output drivers.

For more information about font types, see *System Administration* in the Exstream Design and Production documentation.

### 3.3 Optimizing 3211 line data

The 3211 line data driver is used primarily to send data to legacy archival systems. The 3211 line data output is also used as input with the Exstream Print Miner module. It is a relatively fast output format; however, it has several limitations. These are not production environment limitations, but rather technology limitations within the 3211 line data format.

For more information about the Print Miner module, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

Limitations include the following:

- **Output**—Line data output can include only text and table output. The engine ignores all other design objects (for example, graphics, images, lines, and charts).
- **Fonts**—Line data output is designed to use only one font per application.
- **Overlapping**—You can use only one font for 3211 line data output. Headers and cells in tables cannot overlap.
- **Colors**—You can print only in black and white.
- **Text Positioning**—Several settings affect the exact positioning of text on the page,

including the following:

- The **Characters per inch** setting, on the text box properties
- The **Lines per inch** setting, on the text box properties
- The position of the text box
- Placement of tabs
- Paragraph properties, such as text alignment

For more information about designing for output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## 3.4 Optimizing AFP output

This section discusses the following topics:

- “[How AFP resource files are created in Exstream](#)” below
- “[Engine switches unique to AFP output](#)” on the next page

### 3.4.1 How AFP resource files are created in Exstream

Exstream automatically creates AFP resources based on the options you select when you set up for AFP output. These files are saved as part of the output file and include the following:

- **Form definition files** (formdefs)—When you select the **Create and include Formdef in print file** check box on the **Resource Management** tab of the AFP output properties, Exstream creates a formdef and places it inline automatically to control such aspects as overlays, bin selection, copygroups, jogging, and so on. By default, Design Manager gives the formdef file the name of FORMDEFR. You can customize this name in the AFP output object.
- **Page definition files** (pagedefs)—Pagedefs are used to control variable elements of page formats. This file is optional, since Exstream produces fully composed output.
- **Fonts**—These resources appear in three elements: Coded Font (Xn), Character Sets (Cn), and Code Page (Tn).
- **Page segments** (.psg files)—These resources typically hold images, logos, and signatures.
- **Overlays** (.oly or .ovl files)—These are page formats stored on the mainframe in test/production libraries and referenced during an engine run.

**Note:** Unicode is not supported for AFP resource names. When naming the objects, you must use 8-bit English or European characters.

For more information about managing resources for output, see “[Managing Resources for the Print Stream](#)” on page 229.

### 3.4.2 Engine switches unique to AFP output

The following table lists some of the key engine switches unique to the AFP print stream:

Switch	Value	Default
AFPFONT=mcf num	Use the AFPFONT switch or the AFPFONTSTART switch to generate Mapped Coded Font (MCF) records. You can use these switches to overlay data using StreamWeaver. The argument for these switches is a number (between 1 and 110) to use as the starting number in all MCF records.	Default is 1
AFPMCF1	Use the AFPMCF1 switch to use the old MCF type 1 font records in AFP instead of the newer MCF2 records. Type 1 has been retired according to AFP specifications, but many composition tools continue to support it.	No default
AFP_SEPARATE_INSERTER_BARCODES	Use the AFP_SEPARATE_INSERTER_BARCODES switch to place each inserter barcode into a separate presentation text (PTX) record along with a no-operation (NOP) locator description of the barcode.	No default
AFP_WORKFLOW_BUNDLE	Use the AFP_WORKFLOW_BUNDLE switch to create Begin Named Group (BNG)/End Named Group (ENG) pairs with the same value for each customer in a bundle for consumption by InfoPrint Workflow. This switch requires the starting bundle number as its argument.	No default
COPYGROUPEACHPAGE	Use the COPYGROUPEACHPAGE switch to force pages to be placed on the back of other pages, even with different paper types. This switch suppresses Invoke Media Map (IMM) records before each new paper type, if the IMM appears on the back of a page. This is useful for printers using advanced multiple-up (MUP) form definition files (formdefs).	No default
NO_AFP_DRAWRULE	Because the engine uses Graphics Object Content Architecture (GOCA) fill patterns, you can use the NO_AFP_DRAWRULE switch to override draw rules for rectangular shading.	No default

For more information about engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

## 3.5 Optimizing and enhancing HTML and HTML (Email) output

This section discusses the following topics:

- “[Designing for HTML and HTML \(Email\) output](#)” below
- “[Text Restrictions in HTML and HTML \(Email\) Output](#)” on the next page
- “[Using tables in HTML and HTML \(Email\) output](#)” on page 198

### 3.5.1 Designing for HTML and HTML (Email) output

The appearance of your HTML or HTML (email) output can sometimes differ from your original design. These differences might be due to the relative inflexibility of HTML, different resolutions, substituted fonts, variations in browsers, variations in email clients, or variations in operating systems. For best results, keep in mind the following considerations as you design content for HTML or HTML (email) output:

- **Absolute positioning**—Most email clients do not support absolute positioning. If you are designing for HTML (email) output, avoid using absolute position containers in your container design. If you use an application that includes an absolute position container, then the engine issues a warning and you can receive unexpected results.
- **Color format**—HTML supports only the RGB color format.
- **Embedding linked objects**—Email clients do not support embedding linked objects in the content. If you are designing for HTML (email) output, avoid embedding linked objects in your design.
- **Fill patterns**—HTML does not support pattern fills. If you are designing for HTML or HTML (email) output, do not use patterns to fill objects.
- **External hyperlinks**—You can design external hyperlinks on pages for specific text or a paragraph within a text box or assign them to text boxes, messages, polygons, predefined shapes, static or dynamic images, or charts. The URL can be either static or dynamic (set by variable values). Images that contain an external hyperlink must have an HREF attribute in order to avoid causing security warnings in browsers with common settings.

For information about creating external hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For information about adding hyperlinks to a Live document, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

- **Internal hyperlinks**—You can add internal hyperlinks to text and objects in your designs to

enable customers to easily navigate to additional content within the same document, webpage, or email message. To create an internal hyperlink, you must also create and apply a hyperlink anchor to mark the text that will be the internal hyperlink destination. You cannot apply internal hyperlinks to objects such as images, charts, or shapes. However, if you want to direct a customer to a design object, you can achieve this by applying a hyperlink anchor object to text placed near the design object.

For information about creating internal hyperlinks and hyperlink anchors, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Page orientation**—The HTML output driver does not support mixed page orientations. If you are delivering standard designs to HTML output, you must use either portrait or landscape orientation for all pages in one output file.
- **Resolution**—HTML wraps text differently at different resolutions. For the most accurate display for HTML and HTML (email) output, set the design resolution to 96 dots per inch (dpi). At resolutions other than 96 dpi, some text might not appear at the bottom of text boxes. If you must design at a resolution other than 96 dpi, add extra line feeds at the end of text boxes.

For more information about setting the resolution, see “[Managing output resolution in Exstream](#)” on page 37.

### 3.5.2 Text Restrictions in HTML and HTML (Email) Output

HTML and HTML (email) output have a few unique text restrictions, including the following:

- **Fonts**—Browsers and email clients support a limited list of fonts. If you produce output with unsupported fonts, the browser or email client replaces these fonts with supported fonts. For more precise control over the appearance of your output, Exstream Design and Production allows you to specify fallback fonts for each design font using the **HTML fallback font options** section of the font properties. For more information about substituting unsupported fonts in your output, see “[Specifying Fallback Fonts for HTML, HTML \(Email\), or Multi-Channel XML Output](#)” on page 114.
- **Font sizes**—Exstream does not have direct control over the rendering of the text in browsers or email clients. Exstream provides instructions for font size and formatting, but it is up to the browser or email client to decide how to best render the text. Thus, some variance in font sizes from the original design can be expected.
- **Rotated text**—HTML does not support rotated text. The engine moves all rotated text in your design to a horizontal position during the production run. The only exception is in HTML output, if you use the SVG format for graphics, the engine rotates the text in charts.
- **Tabs**—HTML supports tabs only from the left margin. Exstream approximates left tab use by inserting blank white space with `<span>` tags into the output. All other tab types (including right, centering, and decimal tabs) are not supported.

### 3.5.3 Using tables in HTML and HTML (Email) output

Tables expand in HTML and HTML (email) output to include all text inside each cell, which can affect designs that use the following features:

- **Borders**—In HTML output, you must use CSS styles in order to include border lines on table cells.
- **Cell margins**—If you remove the top and bottom margins inside cells, the engine automatically inserts some padding above and below the text. Although each space is minimal, the total space can cause the original table to grow.
- **Font size**—Table rows can contract to fit small-sized text, but only for some fonts and sizes. For best results, set the font size no smaller than 8 points.
- **Overlapping objects**—HTML does not support overlapping objects, such as overlapping headers, in tables. If the page contains static or dynamic objects that overlap, the first object appears in the table and all overlapping objects are ignored. You also receive an error message from the engine.
- **Text wrapping**—Wrapping text causes an entire table row to grow as text wraps to one or more new lines. Make sure you design variable strings that are longer than the longest possible values for variables or substituted fonts.

For information about tables, see *Designing Customer Communications* in the Exstream Design and Production documentation.

**Tip:** Since expandable tables define object placement, positioning is not as precise as with CSS files. When you design pages, be sure to build in allowances for some variances in object positioning.

For more information about using a cascading style sheet (CSS) to control the final appearance of container design output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 3.5.4 Customizing the Application Title for HTML, EDGAR HTML, and HTML (Email) Output

If you are generating an application for HTML, EDGAR HTML, or HTML (email) output, you can use an array variable to ensure that each run of the same application will have a unique title. For example, you can use a variable that is based on a time stamp or the customer name. Then, when the applications are cataloged, they will have different names, making each version of the application easier to identify.

1. In Design Manager, drag an application object from the Library to the Property Panel.
2. Click the **Basic** tab.
3. In the **Application title** box, click  .
4. In the **Select Variable** dialog box, select the variable and click **OK**.

## 3.6 Optimizing IJPDS output

This section discusses the following topics:

- “Optimizing engine processing time for IJPDS output” below
- “Optimizing print quality for IJPDS output” on the next page
- “Using registration and cue marks” on page 201

### 3.6.1 Optimizing engine processing time for IJPDS output

Engine processing time for IJPDS output can be affected by the following:

- **Color mode**—When you create full-color print jobs, the engine creates a page for each color plane. Drawing the image, graphic, and text character for each plane decreases engine performance. If you create output in grayscale, you can have only print heads defined with ink color set to black. If you create output with highlight colors, you can use only one color plane in addition to the black color plane. For full-color production, all four color planes should be defined.

For more information about color mode, see “[Specifying the Way Color is Produced in Output](#)” on page 40.

- **Font format**—Each change to font color, size, and angle requires a new IJPDS font composed of 256 bitmaps. Therefore, fewer font variations result in faster output. For DBCS applications, changes to font color and size require a new IJPDS font composed of 65,535 bitmaps.
- **Print stream optimization setting**—If you set the print stream to compress all bitmaps, the engine compresses every bitmap in all color planes, which slows processing. This setting can increase engine run time, but it generally enhances printer performance.

For information about optimizing the print stream, see “[Optimizing the size and structure of the print stream](#)” on page 186.

- **Top-of-file images and overlays**—When you use top-of-file images and overlays, the

engine draws static items one time and then references them throughout the job. Using top-of-file objects lets you both improve processing speed, especially when creating output for many customers, and reduce IJPDS file size.

## 3.6.2 Optimizing print quality for IJPDS output

Because IJPDS is an inkjet print stream, print quality can be difficult to control. The dots that compose the text, graphics, and images must be organized and drawn with precision to ensure optimal print quality. External factors such as paper type and quality, ink quality, and printer calibration all affect print quality. Within Exstream, the following factors can affect print quality:

- **Image processing**—To ensure that the integrity of the image is maintained during processing, do not change the original resolution and color settings of imported images. Images that have been imported into Designer that are not CMYK JPEG or CMYK TIFF images must undergo color conversion to RGB, which can degrade print quality. You cannot use scaling for CMYK, JPEG, or CMYK TIFF images.

For information about specifying image settings for imported images, see *Importing External Content* in the Exstream Design and Production documentation.

- **Linearization table usage**—You can use linearization tables (lookup tables) to reduce color intensity and ink usage. Linearization tables also decrease packaging and engine run times.

For information about providing linearization tables, see “[Specifying a Linearization Table for Tone Correction](#)” on page 120.

- **Barcode drawing method**—Ink run during printing can sometimes cause barcodes to not scan or to scan improperly. For more control about to help ensure proper scanning, you can specify that the barcode is generated by drawing lines.

For more information about barcodes, see “[Using Barcodes in Exstream](#)” on page 256.

- **Image dithering**—When working with black-and-white images, the dithering method (using patterns to represent different colors) can affect print quality. Select one of eight dithering methods based on image data and printing needs.

For more information about dithering, see *Importing External Content* in the Exstream Design and Production documentation.

- **Object overlap**—Because it is difficult for Exstream and inkjet printers to handle overlapping objects perfectly, you might encounter problems with ink saturation and distortion on complex designs with overlapping objects.

- **Resolution**—Based on the resolution you specify, Exstream uses a single output resolution per print job. This setting can affect image quality.

For information about setting the resolution, see “[Managing output resolution in Exstream](#)” on page 37.

### 3.6.3 Using registration and cue marks

Many IJPDS production systems require marks to register CMYK documents (usually a 0.25 inch cross-hair printed at the top of the page). Some systems also require cue marks (usually a 0.25 x 0.25 inch black box in the upper left corner) to identify page fronts and backs for duplex jobs. You can incorporate cue and registration marks when you design your page.

## 3.7 Optimizing Metacode output

This section discusses the following topics:

- “[Designing for Metacode output](#)” below
- “[Creating reverse video text with named fonts](#)” on the next page
- “[Using VSAM package files to enhance performance on z/OS](#)” on page 203
- “[Building print resource files](#)” on page 203
- “[Engine switches unique to Metacode](#)” on page 204

### 3.7.1 Designing for Metacode output

In Metacode output, the following factors can affect the quality and processing time of print jobs:

- **Rotated text**—When you rotate a text box, the Metacode output driver interprets the rotated text as different font sets and creates a raster image for each character. Because this process results in a larger file size, try inserting the text as an image instead of using rotated text.
- **Edge-to-edge printing**—Unlike some output formats, Metacode does not support edge-to-edge printing. You must leave a small margin on either side of the page. If text boxes extend too far to the edges, text might be lost.
- **Shading**—In the Metacode output driver, shading information is placed at the end of a Metacode record. If long records are truncated anywhere in the workflow, you will see gaps in the shading. The data appears correct, but shading can appear in blocks of space, with white gaps appearing throughout the record.
- **Fonts**—The quality for fonts from Windows might not be as good as other fonts at small sizes. Bitmap (raster) fonts start out as outline (vector) fonts. At packaging time, the engine takes the outline fonts used in the design and rasterizes them for Metacode output. Since there is no pixel-to-pixel conversion, rounding (also called hinting) can affect perceived quality, especially at smaller sizes. The smaller the point size and the lower the resolution, the less the bitmapped version of the character prints like the original.

- **Tables**—Since the printer fills table cells individually, cell patterns or shading cannot be forced to match and align. Metacode has only three fill patterns. To increase the number of pattern options and enhance fill consistency, you can place two patterns on top of each other. If you see lines in your output that should not be there, you can reduce the visibility of these lines by selecting a single pattern fill. This way you see only the slight irregularity between each cell. Also, you can try several colors to see which fill looks best.

### 3.7.2 Creating reverse video text with named fonts

Reverse video text formatting (printing white text on a black background) offers a visual contrast to emphasize certain text on a page or message.

For static objects, you can create white text on a black background with embedded graphics. If the data is variable, however, you can purchase Metacode fonts that have the reverse video presentation. These fonts have the black background and white character set as part of their character glyphs. You must call these fonts from the Metacode printer as a named font.

To use a named font with Metacode:

1. Secure a TrueType font similar to the named Metacode font.
2. In Design Manager, create a font object for this font and open it in the Property Panel.
3. To map the font to a name table, complete one of the following tasks:

To	Do this
Map a font to a name table in SBCS mode	Click <b>ASCII Name Table</b> . The <b>Font Name Mapping</b> dialog box opens.
Map a font to a name table in DBCS mode	Click <b>DBCS Name Table</b> . The <b>Font Name Mapping</b> dialog box opens.
Map a font to an ASCII name table in SBCS/DBCS mode	Click <b>ASCII Name Table</b> . The <b>Font Name Mapping</b> dialog box opens.
Map a font to a DBCS name table in SBCS/DBCS mode	Click <b>DBCS Name Table</b> . The <b>Font Name Mapping</b> dialog box opens.

4. Click the **Metacode** tab.
5. Enter the Metacode callout name for the font. Your print room administrator can supply the Metacode font name.
6. From the Menu bar, select **Edit > Save**.
7. From the Library, drag a Metacode output object to the Property Panel.
8. Click the **Resource Management** tab.

9. From the **Fonts** drop-down list, select **Embed only unnamed fonts**. Exstream embeds only the fonts that do not have a Metacode font name in the name table.
10. From the Menu bar, select **Edit > Save**.

For more information about creating named fonts, see *System Administration* in the Exstream Design and Production documentation.

### 3.7.3 Using VSAM package files to enhance performance on z/OS

Using a Virtual Storage Access Method (VSAM) package file has the potential to reduce the production time. The production engine on the z/OS operating system reads application objects at the time they are required, instead of reading the entire package file into memory at the beginning of the run.

However, there are several factors that affect the amount of time you gain (or, potentially, lose) using VSAM package files. These include the following:

- **Size of the package file**—If you have small package files, production time is not affected, since the time to read the entire package file is short.
- **Percentage of documents accessed**—If you want to produce only a percentage of the documents in an application, you can see a large performance gain. The same package file in a larger run that uses all objects decreases performance.
- **Contents of the package file**—You gain performance benefits only if there is a higher ratio of application objects (such as pages and messages) to objects (such as fonts and output drivers).
- **Tuning status of VSAM datasets**—Because z/OS default settings are based on obsolete disk technology (such as sectors, cylinders, and tracks), you must tune the VSAM dataset to gain performance benefits on currently available hardware.

### 3.7.4 Building print resource files

Because inline resources are slow to load in Metacode, you can have Design Manager create separate print resource files for use with Metacode output. When you set up for packaging, you load these resource files separately to the printer or resource library.

For more information about creating print resource files, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

### 3.7.5 Engine switches unique to Metacode

The following table lists some of the key engine switches that are unique to the Metacode print stream:

Engine switches

Command	Value	Default
BLOCK_OUTPUT_MVS	Use the BLOCK_OUTPUT_MVS switch to enable byte-by-byte output, similar to that of a PC. This switch honors the blocking options on output object settings for the mainframe platform. This switch is used when output is made on a mainframe and then sent to a third party.	No default
METAPAGE=value	Use the METAPAGE switch to specify whether x8B or x89 indicates a new page for printing Metacode based on your consumer requirements.	No default
META_SIMPLEXDUPLEX_WITH_NUFRONT	Use the switch with the Metacode command SIDE=NUFRONT during Metacode simplex and duplex mode processing, if you want to produce simplex pages as output without changing the printer to simplex.	No default
NO_META_FORMFEED_AFTER_INTERLEAVED	Use the NO_META_FORMFEED_AFTER_INTERLEAVED switch to override \x8B\x20\x01 records (form feeds) after page interleaved graphics. \x8B\x20\x01 records (form feeds) cause blank pages on some Metacode printers.	No default

## 3.8 Optimizing MIBF output

This section discusses the following topics:

- “[Designing for MIBF output](#)” below
- “[Using tone correction curves to optimize ink usage](#)” on the next page
- “[Producing rotated output for multiple-up sheets in MIBF output](#)” on the next page

### 3.8.1 Designing for MIBF output

In MIBF output, the following factors can affect the quality and processing time of print jobs:

- **Image type**—For best output results, use CMYK JPEG and CMYK TIFF images. These image types allow the highest quality plane separation. All image types are processed, but only CMYK JPEGs and CMYK TIFFS receive optimized color separation. Other images

must be separated through a Leadtools toolkit, which can affect image quality.

For more information about images, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Barcode drawing method**—Ink run during printing can sometimes cause barcodes not to scan or to scan improperly. To help ensure proper scanning, you can specify that the barcode is generated by drawing lines.

For information about barcodes, see “[Using Barcodes in Exstream](#)” on page 256.

- **Print stream optimization setting**—If you set the print stream to compress all graphics and on-page images, the engine compresses every bitmap in all color planes, which slows processing. This setting can increase engine run time, but it generally enhances printer performance.

For information about optimizing the print stream, see “[Optimizing the size and structure of the print stream](#)” on page 186.

### 3.8.2 Using tone correction curves to optimize ink usage

You can use tone transfer curves (or correction curves) to reduce the amount of ink used on text, graphics, and images.

To set up tone correction curves:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Configuration** tab.
3. In the **Tone correction curves** area, from the **Color** drop-down list, select a color plane to use.
4. From the **Type** drop-down list, specify the bitmap type you want to use.
5. In the **File** box, identify the tone correction file to use.
6. From the Menu bar, select **Edit > Save**.

### 3.8.3 Producing rotated output for multiple-up sheets in MIBF output

To rotate pages for placement on multiple-up (MUP) sheets, you must use MUP objects and output queues. You must license the High-Volume Delivery module to use these features.

To produce rotated output, you must set the orientation of the design page to either portrait or landscape, and set the orientation of the MUP sheet to portrait. You can then use any orientation for the frames on the MUP sheet to produce the correctly rotated output.

For information about setting up MUP sheets and using MUP frames, see “[Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects](#)” on page 300.

## 3.9 Optimizing and enhancing PDF, PDF/A, or PDF/VT output

This section discusses the following topics:

- “[Compressing PDF files](#)” below
- “[Creating linearized PDF output](#)” on the next page
- “[Using editable PDF forms in Exstream](#)” on the next page
- “[Creating PDF AcroForms output](#)” on page 146
- “[Adding accessibility tags to PDF files for screen readers and text-to-speech converters](#)” on page 208
- “[Optimizing Adobe Acrobat tool compatibility](#)” on page 210
- “[Adding security to PDF files](#)” on page 210
- “[Using digital IDs to digitally sign and authorize documents](#)” on page 213
- “[Creating COLD PDF output](#)” on page 215
- “[Controlling how bookmark entries are included in output](#)” on page 215

### 3.9.1 Compressing PDF files

You can compress PDF files to reduce the file size of the final customer PDF. This can be useful if you are creating PDF files for electronic delivery (such as PDFs posted online or delivered through email). From Exstream, you can select a compression level **1** through **9** (where **1** is the lowest and **9** is the highest) or **Max**.

When selecting a compression level, keep in mind the following:

- Compression levels **1** through **9** compress the content of the PDF (such as text, tables, or images).
- A compression level of **Max** compresses the content of the PDF using a level **9** compression and applies additional object compression to the PDF. Object compression compresses the content in the object streams of the PDF (such as dictionaries of pages, annotations, or

fonts), which are not yet included in the print stream. A compression level of **Max** is available only if you have selected a PDF version 1.5 or later.

For more information about specifying a PDF version, see “[Specifying a PDF Version for Compatibility with Viewers, Process Tools, and Utilities](#)” on page 139.

- Standard PDF output typically uses a compression level of **6**.

To compress PDF files:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Compress** check box.
4. To specify how much the file is compressed, select a compression level from the **Compression level** box.
5. From the Menu bar, select **Edit > Save**.

### 3.9.2 Creating linearized PDF output

Unlike normal PDF files, the first pages of a Linearized PDF file appear on-screen before the entire PDF has downloaded. Linearized PDF lets you optimize the viewing speed for PDF files, which is helpful if you have large PDF files or if you intend to send PDF files over the Internet through email or for downloading. Linearized PDF output is not optimized for multiple-up (MUP) printing.

To create linearized PDF output:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Linearized** check box.
4. From the Menu bar, select **Edit > Save**.

### 3.9.3 Using editable PDF forms in Exstream

Editable PDF forms are often used for data collection purposes. If you are using the AcroForm Pass-through feature, you must enable this feature on the PDF output object. Acroform Pass-through is available only if you have selected a PDF version 1.4 or later.

For more information about specifying a PDF version, see “[Specifying a PDF Version for Compatibility with Viewers, Process Tools, and Utilities](#)” on page 139.

To set up to use editable PDF forms:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Basic** tab.
3. Select the **AcroForm Pass-through** check box.
4. From the Menu bar, select **Edit > Save**.

For more information about pre-filling and mining PDF forms, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

### 3.9.4 Adding accessibility tags to PDF files for screen readers and text-to-speech converters

**Note:** You can use accessibility features with PDF and PDF/A output, however, the PDF/VT output type does not support accessibility.

When you create PDF or PDF-based output, you can include accessibility tags to control and optimize the way a document will be read by screen readers and text-to-speech converters. For example, if you are creating student progress reports, you can use accessibility tags to enhance the readability of the reports in PDF output for visually impaired parents or guardians.

To set up a design to be read by accessibility tools, you must first turn on accessibility tagging by selecting an accessibility standard from the **Accessibility standard** drop-down list on the PDF output object. By default, when you enable accessibility tagging in Exstream, all text contained within textual design objects (such as table cells, text boxes, and paragraphs) will be read by an accessibility tool, but non-textual objects (such as images, charts, and shapes) will be skipped by accessibility tools. However, you can customize this behavior. For example, if your design includes an image that you do not want an accessibility tool to skip, you can specify alternate text for the image object, so that an accessibility tool will announce a description of that image.

Exstream also allows you to add and customize accessibility tags for additional design components such as titles, headings, and lists. However, keep in mind that you must manually add the accessibility tags to each of these components in your design.

For more information about optimizing a design for PDF accessibility tools, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Keep in mind the following considerations when adding accessibility tags to PDF files:

- Accessibility for PDF output is available only if you have selected a PDF version 1.4 or later.  
For more information about specifying a PDF version, see “[Specifying a PDF Version for Compatibility with Viewers, Process Tools, and Utilities](#)” on page 139.
- When you use accessibility tags, you cannot also use multiple-up (MUP) options. If you

specify MUP options for an output queue with a PDF or PDF/A driver, accessibility tags are automatically disabled during the engine run and you receive a message.

## Creating accessible PDF output

To create accessible PDF output:

1. In Design Manager, from the Library, drag a PDF or PDF/A output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Accessibility standard** list, select one of the following accessibility standards as required by your organization:

Accessibility standard	Description
WCAG 2.0	When you select <b>WCAG 2.0</b> , Exstream applies the default accessibility tagging for objects in your design to the output. However, you can also customize these settings and other accessibility objects in your design as needed.
PDF/UA  PDF/UA is not supported with PDF version 1.3.	<p>When you select <b>PDF/UA</b>, Exstream applies the default accessibility tagging for objects in your design to the output. However, you can also customize these settings and other accessibility objects in your design as needed.</p> <p><b>Important:</b> Selecting the <b>PDF/UA</b> option in the <b>Accessibility standard</b> list does not automatically make your document PDF/UA-compliant. At a minimum, you must design your document so that it is compliant with PDF/UA guidelines and standards.</p> <p>For information about PDF/UA documentation guidelines, see the PDF/UA documentation that your organization references.</p> <p>Keep in mind that Exstream does not validate your documentation that you design for PDF/UA compliance. If you create a document that is rejected by a PDF validation tool, then you must go back and fix your design using the design customization tools in Exstream. For example, suppose that your document is rejected because of a skipped heading level (such as a document with a heading level 1 followed by a heading level 3). To correct this issue, re-open your document in Designer and adjust the heading levels to occur in sequence (without skipping heading level 2 in this example) before re-generating the output.</p>

Keep in mind that if your design includes design time or run time import of a PDF with accessibility tagging, then the accessibility standard of the imported PDF should match the accessibility standard that is specified on the PDF output object. If these standards do not match, the imported PDF might lose some accessibility features in your final PDF output.

4. From the Menu bar, select **Edit > Save**.

For information about the default tagging structure that Exstream provides when you enable accessibility tagging in the output, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## 3.9.5 Optimizing Adobe Acrobat tool compatibility

When you design a document for PDF output, you can optimize the way PDF output in Exstream works with Acrobat tools in post-production.

This section discusses the following topics:

- “[Customizing the application title for PDF catalogs](#)” below
- “[About the touch-up text tool in Adobe Acrobat Reader](#)” below

### Customizing the application title for PDF catalogs

If you are generating an application for PDF output that will be cataloged, you can use an array variable to ensure that each run of the same application will have a unique title. For example, you can use a variable that is based on a time stamp or the customer name. Then, when the applications are cataloged, they will have different names, making each version of the application easier to identify. The application title information appears in the document information dictionary entry for Title (specified as /TITLE) in the PDF output file.

To customize the application title for PDF output:

1. In Design Manager, drag an application object from the Library to the Property Panel.
2. Click the **Basic** tab.

3. In the **Application title** box, click  .

The **Select Variable** dialog box opens.

4. Select the variable and click **OK**.

The **Select Variable** dialog box closes, and the variable you selected appears in the **Application title** box.

5. From the Menu bar, select **Edit > Save**.

### About the touch-up text tool in Adobe Acrobat Reader

The Touch-Up Text tool in Adobe Acrobat Reader works best with objects you create in Designer. The Touch-Up Text tool might not be able to parse the text you insert into the print stream at run time with placeholder variables.

## 3.9.6 Adding security to PDF files

If information included in a PDF is confidential, you can use security features in Exstream to limit access to the file and to limit who can change the security options. To secure a PDF file,

you can apply a password-protected encryption to the file so that only users with the correct password can open the PDF file. In addition, you can also restrict the permissions users have to print, highlight, or edit the content of the PDF after they have successfully accessed the PDF. For example, suppose you are distributing customer-sensitive information in a PDF, and to protect that information, you want to make sure the PDF content cannot be printed or redistributed in any way. You can set up the PDF to not only prevent anyone from being able to print the PDF, but also prevent anyone from being able to copy and paste content into a format that can be printed.

**Note:** Encryption and password-protection are not supported for PDF/A or PDF/VT output.

To add security to PDF files in Exstream, complete the following tasks as needed:

- “[Setting up encryption and passwords](#)” below
- “[Setting up permissions for password-protected documents](#)” on page 213

## Setting up encryption and passwords

Different encryption levels are available depending on the PDF version you are creating. The following table shows the PDF version required to use each level of encryption:

Encryption levels

Encryption level	Required PDF version
No encryption	1.3 or later
40-bit RC4 encryption	1.3 or later
128-bit RC4 encryption	1.4 or later
128-bit AES encryption	1.6 or later
256-bit AES encryption	1.7 or later

For more information about specifying a PDF version, see “[Specifying a PDF Version for Compatibility with Viewers, Process Tools, and Utilities](#)” on page 139.

Any passwords you set up must meet the following considerations:

- Passwords are case-sensitive and can be only alphanumeric characters.
- Characters such as \$,&, or ~ are not accepted.
- Passwords can be variable or static.

To set up encryption and passwords:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Security** tab.
3. In the **Encryption Options** area, from the **Encryption level** drop-down list, select one of the following options:
  - **No encryption**
  - **256-bit AES encryption**
  - **128-bit AES encryption**
  - **128-bit RC4 encryption**
  - **40-bit RC4 encryption**

**Note:** The option to select **256-bit AES encryption** in the **Encryption level** list appears only if you have previously specified **1.7** in the **Version** list on the **Basic** tab of the PDF output properties.

4. To set up the type of password to use to open a document, do one of the following:

To	Do this
Use one password	In the box below <b>Open documents</b> , enter a character string (up to 32 characters).
Use a variable-based password	In the <b>Open documents</b> box, click  and select a variable.  <b>Tip:</b> If you want to change the password each time there is a break in the output, use the same variable on the <b>Breaks</b> tab of an output queue.

5. To set up the type of password to use to change the security options, do one of the following:

To	Do this
Use one password	In the box below <b>Change security options</b> , enter a string (up to 32 characters).
Use a variable-based password	In the <b>Change security options</b> box, click  and select a variable.  <b>Tip:</b> If you want to change the password each time there is a break in the output, use the same variable on the <b>Breaks</b> tab of an output queue.

For more information about output queues, see “[Controlling and Setting Up High Volume Production Options with Output Queues](#)” on page 315.

## Setting up permissions for password-protected documents

After you have set up the encryption levels, you can prevent anyone—even users with valid passwords—from printing or changing a document.

To add limits to password-protected documents:

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Security** tab.
3. In the **Permission Options** area, specify the action or actions you want to limit:

To	Do this
Specify that no one can print any part of the document	Select the <b>Prevent printing</b> check box.
Specify that no one can highlight and copy any document content	Select the <b>Prevent text/graphic selection</b> check box.
Specify that no one can make any edits to either text or graphics	Select the <b>Prevent document changes</b> check box.
Specify that no one can change form fields or add new ones, but users can fill in existing fields on forms	Select the <b>Prevent annotation and form field alteration</b> check box.

### 3.9.7 Using digital IDs to digitally sign and authorize documents

Digital IDs are similar to handwritten signatures in that they act as an official verification or authorization. You might be required to validate that a PDF is coming from a specific company, organization, or person, and that no changes have been made before the recipient opens it. Recipients who receive digitally signed documents can be reasonably assured that the document is from the stated sender. Keep in mind that you can use digital signatures only with PDF or PDF/A output; the PDF/VT output type does not support digital signatures.

To include a digital ID in a package file, the ID must be in the Windows Personal Certificate folder on the workstation used for packaging. If the digital ID is password-protected, you are prompted to enter the password at packaging time. If you enter the wrong password, or no password, you receive an error and the packaging process stops.

**Note:** The password prompt uses the name entered when the digital ID was installed. If no name is available on the prompt, a hotfix is available from Microsoft.

During production, only one digital ID can be added to the output file. The digital ID is used as the author signature in the PDF output file, and it includes the following:

- Text entered in the **Reason** box, if any
- Text entered in the **Location** box, if any
- Time the output was generated

A document with a digital ID can still be changed in Adobe Acrobat. For example, additional digital IDs or notes could be added. To prevent changes and lock the document, you must set permission options in the **Permission Options** area.

For more information about setting permissions, see “[Setting up permissions for password-protected documents](#)” on the previous page.

Digital IDs are recognized only in Adobe Acrobat version 6.0 and later.

## Adding a digital ID

1. In Design Manager, drag an output object from the Library to the Property Panel.
2. Click the **Security** tab.
3. In the **Digital Signature Options** area, select the **Place digital ID on PDF** check box.
4. Under the **Name** box, click **Select**.

The **Digital ID Selection** dialog box opens, and it lists all of the digital IDs in your Windows Personal Certificate folder. Any digital IDs that are revoked, expired, not valid for signing, or not exportable are inactive and cannot be selected.

5. Select the digital ID you want to add and click **OK**.
6. If you want to add a reason for including the digital ID, enter the reason in the **Reason** box (optional).
7. If you want to enter the location of the digital ID owner, enter the location in the **Location** box (optional).
8. In the **Document Modification Options** list, select one of the following options to specify which changes can be made to the document without invalidating the Digital ID:
  - **No changes to the document**
  - **Form-filling, instantiating page templates and signing**
  - **Form-filling, instantiating page templates, signing and annotation changes**

## How digital IDs are validated in Exstream

The digital ID is validated each time you open the output object. If a digital ID is expired, is revoked, or is no longer valid or exportable, you receive a warning message. Because digital IDs

are validated as they are included in the package file, they are not validated at production time. During packaging, Exstream creates a log file that lists all digital IDs that are included in the package file and their valid date ranges. To ensure digital IDs are valid at production time, review the log file.

**Tip:** To include multiple digital IDs in a package file, you can create multiple PDF output objects, each with a different digital ID. Then, place rules on an output queue to select which PDF output object to use.

### 3.9.8 Creating COLD PDF output

If you have licensed the High-Volume Delivery module, you can set up PDF output for sending to a third-party Computer Output to Laser Disc (COLD) system for archiving. Design Manager handles the index record and the system inserts separate entries as a search key. COLD output settings must be set up from both your output queue and your output object.

To create COLD PDF output:

1. In Design Manager, drag an output queue object from the Library to the Property Panel.
2. Click the **Breaks** tab.
3. Select the **Split after event** check box.
4. In the **Customers** box, enter the number of customers your COLD system can support before a break is required.
5. From the **Customer file method** drop-down list, select **Each customer printable**.
6. Create an application search key that the engine inserts as an index record. Set up the search key to use a formula variable with values that coincide with your COLD system's indexing specifications.

For more information about application search keys, see “[Using Application Search Keys](#)” on page 326.

7. From the Library, drag a PDF output object to the Property Panel.
8. Click the **Advanced** tab.
9. From the **Search Keys** drop-down list, select **As Is**.

### 3.9.9 Controlling how bookmark entries are included in output

You can control how bookmark entries are included in output by using the PDF, PDF/A, PDF/VT, or VDX output object properties. The properties allow you to suppress bookmarks even if they are already set up for application objects or design objects. In addition, you can

specify whether bookmarks will be generated based on customers, documents, pages, or objects. You can also use the properties to maintain the existing bookmarks in dynamically imported PDF content when producing Exstream-generated output.

To control how bookmark entries are included in PDF, PDF/A, PDF/VT, or VDX output, you must complete the following steps:

1. In Design Manager, from the Library, drag the output object to the Property Panel.
2. Click the **Basic** tab.
3. In the **PDF bookmarks** area, select one or more check boxes to include bookmarks for the objects in the application:
  - **Customers**—Bookmarks are generated for each customer as specified by the **Customer ID for reporting** variable on the application properties.
  - **Documents**—Bookmarks are generated for documents as specified by the **ID for bookmark** variable on the document properties.
  - **Pages**—Bookmarks are generated for pages as specified by the **ID for bookmark** variable on the document properties.
  - **Objects**—Bookmarks are generated for all text and objects with which you have associated a bookmark.
  - **Use imported bookmarks**—Bookmarks are generated from imported PDF content.

If you do not select any check boxes, no bookmarks will be generated in the output, even if you have set up variables or bookmark markers.

## 3.10 Optimizing PostScript output

This section discusses the following topics:

- “[Using PostScript commands supported in Exstream](#)” on the next page
- “[Using named colors in PostScript and PostScript-based output](#)” on page 218
- “[Using Ghostscript to view PostScript output](#)” on page 218
- “[Using PostScript Encodings in DBCS Applications](#)” on page 218

### 3.10.1 Using PostScript commands supported in Exstream

In Exstream, you can use two types of commands with PostScript output:

- **Media call**—Use this command to specify a type of paper.
- **Line printer remote (LPR)**—Use this command to submit a PostScript print file to a printer across a network.

Each of these commands can be placed in a command or batch file and invoked from the desktop.

For information about running the engine from the command prompt, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

#### PostScript media commands

A media call command specifies a type of paper. These document structuring convention (DSC) commands are used only in PostScript printing.

##### Example of a media command to select paper

```
<< /PageSize [792 612] /MediaType (letter) /MediaColor (White) /MediaWeight  
70.5 >>setpagedevice
```

**Note:** Exstream does not support the /MediaClass parameter available with Level 3 PostScript printers.

#### PostScript LPR commands

You can submit PostScript print files to a printer across a network. On Windows NT or 2000 operating systems, you use LPR commands.

##### LPR command example

```
lpr -S 192.0.0.200 -P PASS -o l %1
```

##### LPR command example

```
lpr -S 13.244.117.14 -P dt61xxhold -o l %1
```

The following table lists some of the parameters available for use in LPR commands:

#### Parameters in LPR commands

Parameter	Function
-S	Identifies the server, followed by the server IP address
-P	Identifies the printer, followed by the printer name
-o 1	Indicates that this is a PostScript job (single command)
%1	Functions as a wildcard command, which generally holds the file name of the PostScript print stream.

### 3.10.2 Using named colors in PostScript and PostScript-based output

If your PostScript or PostScript-based printer has named colors stored on the device, Exstream can call these colors by name, allowing the printer to use its calibrated CMYK values. To use the output color definition, enter the name in the **Output Color Name** box on the Basic tab of the named color object properties.

For information about creating named colors for PostScript output, see *System Administration* in the Exstream Design and Production documentation.

### 3.10.3 Using Ghostscript to view PostScript output

You can use Ghostscript, which is a commercially available PostScript and PDF conversion and rendering tool, as a PostScript file viewer for Microsoft Internet Explorer and IBM WebExplorer. Some printer-specific settings (for example, some printer resource settings) might not be supported by the viewer.

For information about installing Ghostscript, see *Installation and Upgrade Information* in the Exstream Design and Production documentation.

For more information about printer resource settings, see “[Managing Resources for the Print Stream](#)” on page 229.

### 3.10.4 Using PostScript Encodings in DBCS Applications

Exstream automatically detects the encoding based on the font mapping name for any encoding that IBM's International Components for Unicode (ICU) supports. Exstream generates a message in the message file with the font name and the encoding when an application is packaged.

The following table shows how Exstream determines the encoding type, based on the font name:

If font name contains	Encoding will be
UCS2 or UTF 16 in the font name	UTF16 (UNICODE)
RKSJ in the font name	SJIS (SHIFTJIS)
UTF8 in the font name	UTF8 (ASCII)
B5 in the font name	Big5
Unis in the font name	Korean1
All others (unknown encoding)	Defaults to UTF16 (UNICODE)

## 3.11 Optimizing PowerPoint output

This section discusses the following topics:

- “Designing for PowerPoint output” below
- “Text restrictions for PowerPoint output” on the next page
- “Text size considerations for PowerPoint output” on the next page
- “Using tables in PowerPoint output” on page 221

### 3.11.1 Designing for PowerPoint output

Your PowerPoint output might appear different than your original design. This can be due to different resolutions, installed fonts, or software versions of PowerPoint. For best results, keep in mind the following considerations as you design pages in Designer for PowerPoint output:

- **Slide transition and animation effects**—You cannot specify slide transition effects and animation on PowerPoint objects from within Exstream.
- **Resolution**—For the most accurate display, set the design resolution to 96 dots per inch (dpi). At resolutions other than 96 dpi, some text might not appear at the bottom of text boxes, because in PowerPoint, text wraps differently at different resolutions. If you must design at a resolution other than 96 dpi, add extra line feeds at the end of text boxes.
- **Images**—In Designer, do not use patterns to fill objects for PowerPoint output. Patterns do not appear.
- **Color format**—The PowerPoint output driver supports only RGB format.

- **Printing**—Sometimes when printing, PowerPoint changes the color of an element or line to black. Since printing is solely controlled by PowerPoint, it cannot be controlled in Exstream.
- **Hyperlinks**—You can design hyperlinks on pages for specific text or assign them to text boxes and messages. The link addresses can be either static or dynamic (set by variable values).  
For information about creating hyperlinks, see *Designing Customer Communications* in the Exstream Design and Production documentation.
- **Page orientation**—PowerPoint output does not support mixed page orientations. You must either use portrait or landscape orientation for all pages in one output file.

### 3.11.2 Text restrictions for PowerPoint output

PowerPoint output has a few unique text restrictions, including the following:

- **Fonts**—The fonts seen by users depend on the fonts installed on their computers. If you produce output in a font that a computer does not recognize, PowerPoint determines a substitute font. To provide some support for the problem, Exstream tracks the original font and maps every font with an appropriate substitute.
- **Rotated text**—PowerPoint does not support rotated text. The Exstream engine moves all rotated text in your design to a horizontal position during the production run. The only exception is that if you use the SVG format for graphics, the engine rotates text in charts.
- **Tabs**—PowerPoint does not support tabs. Exstream approximates tab use by inserting blank white space, based on a left tab. All other types of tabs (right, centering, and decimal tabs) are not supported.
- **Margins**—PowerPoint does not support text margins on text boxes or table cells. Any margins set in your design are ignored when text is placed. Keep in mind this limitation as you design, especially when placing images and text in the same text box or table cell. The images will have the proper margins, but the text will not.
- **Line spacing**—PowerPoint ignores line spaces in text. Exstream treats all line spaces as a single space in PowerPoint output.

For information about designing text, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 3.11.3 Text size considerations for PowerPoint output

PowerPoint was designed with large text and presentations in mind. Text in PowerPoint output is often larger than the same text in other output files, such as PDF. Though Exstream has no control over the rendering of text in the PowerPoint software, Exstream includes instructions for

font size and attributes in the package file. When you open the file in PowerPoint, PowerPoint determines how to best render the text, and this rendering is often slightly larger than the equivalent text in a PDF file, opened in Adobe Acrobat Reader. This increased text size causes the text to flow outside of text boxes and table cells.

**Tip:** Add extra space (carriage returns) to text boxes and table cells to leave room for the slightly larger text that PowerPoint renders.

You should avoid small font sizes when designing for PowerPoint output. Text rendering can vary from font style to font style. For example, Arial appears correctly at 11-point font size, but not at 10- or 12-point. Most fonts appear correctly at 18-point size or above.

Character spacing differs slightly between Exstream and PowerPoint. If you change the font size for your text, you might need to adjust spacing and alignment in PowerPoint after Exstream creates the output.

Since PowerPoint does not handle small font sizes (7- to 10-point) consistently, this might cause the leader dots in pages from tables of contents imported from RTF files to wrap to the next line. As a workaround, you can increase the point size of the font on the table of contents by one or two levels. This adjustment makes the text and leader dots more manageable in PowerPoint and prevents the text from wrapping.

### 3.11.4 Using tables in PowerPoint output

Tables expand in PowerPoint to include all text inside each cell. This expansion can affect slide designs containing the following features:

- **Text wrapping**—Wrapping text causes an entire table row to grow as text wraps to one or more new lines. If text is too large to fit in a table cell in PowerPoint, the text overflows and prints outside of the cell. Make sure you design variable strings longer than the longest possible values for variables or substituted fonts.

For information about tables, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Font size**—Table rows contract to fit small-sized text, but only to a certain degree. For best results, set text no smaller than 8 points.
- **Cell margins**—If you remove the top and bottom margins inside cells, the PowerPoint output driver automatically inserts some padding above and below the text. Although each space is minimal, the total space can cause the original table to grow on the page.
- **Overlapping objects**—PowerPoint does not support overlapping objects, such as overlapping headers, in tables. If the page contains static or dynamic objects that overlap, then the first object appears in the table and all overlapping objects are ignored. You also receive a warning message.

## 3.12 Optimizing RTF output

This section discusses the following topics:

- “[Adding bookmarks to RTF output](#)” below
- “[Designing objects for editable RTF output](#)” below
- “[Defining the spelling dictionary language in RTF output](#)” on the next page
- “[Viewing RTF output in other programs](#)” on page 225

### 3.12.1 Adding bookmarks to RTF output

If you want to automatically add markers to your RTF output as bookmarks, you can use no-operation (NOP) search keys.

To add bookmarks to RTF output:

1. In Design Manager, drag an application search key object from the Library to the Property Panel.
2. Click the **Basic** tab.
3. In the **Output** area, from the **Type** drop-down list, select either **All outputs** or **Specified PDL**.
4. If you selected **Specified PDL**, select **RTF** from the drop-down list below.
5. From the **Type** drop-down list, select **NOP comments**.
6. In the **Value** area, specify the content of the search key.

For more information about setting up search keys, see “[Adding Non-Printing Data or Production Device Controls with Search Keys](#)” on page 326.

### 3.12.2 Designing objects for editable RTF output

Keep your design simple for RTF output. You can include some graphics, images, and small tables, but use simple flow. Consider using only one flow page and no side-by-side text boxes.

In contrast to Exstream designs, which are page-based, RTF output is document-based. In Designer, you position objects precisely on a page and describe a page layout into which data is applied. RTF, however, is intended to flow across page boundaries. The concept of a page layout does not apply.

This difference means you should avoid using the following in your designs for RTF output:

- **Drawing tools**—Drawing tools such as rectangles and circles are not suited for RTF output since they are expected to go in a specific place on the page.
- **Objects relative to another on the right or left**—Objects should be relative only to objects above them.
- **Side-by-side objects or text boxes**—Objects and text boxes are produced one after another, vertically.
- **Vertical text**—RTF output does not support rotated text. All rotated text is drawn horizontally.
- **Embedded objects**—RTF output does not support embedded objects.
- **Overlapping objects**—Tables, images, and text wraps appear beneath draw objects (all other design objects). For example, if you place a table in front of a shape on a design page, then the shape will appear in front of the table in RTF output.
- **Page numbers at the bottom of a page**—RTF viewing software might insert page breaks that move page numbers to the following page. In these cases, the page numbers are not recalculated.

For more information about designing in Exstream, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## Using Tables in Editable RTF Output

Although Designer lets you create tables with multiple header and footer rows, Microsoft Word does not support footer rows and designates only the first row as a header. To avoid unpredictable results when designing for editable RTF, you should identify only the first row of a flowing table as a repeating header, and you should not use footer rows.

**Caution:** Overlapping headers are not supported in RTF output.

In editable RTF output, table rows split even if you do not select the **Can split** check box on the row properties.

For more information about creating tables in Designer, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 3.12.3 Defining the spelling dictionary language in RTF output

After end users make updates to an RTF file in a text editor (Microsoft Word, for example), they can run spell check to make sure that all of the words in the file are spelled correctly. If the spelling dictionary language and the text language do not match, words are marked as incorrect even if they are spelled correctly because they do not appear in the spelling dictionary. For

example, suppose that the spelling dictionary is English, but the text is in Spanish. Although the words are spelled correctly in Spanish, they are marked as misspelled because they are not in the English dictionary.

The RTF spelling dictionary language is specified in the \deflangN or the \deflangfe tag. The engine automatically populates these tags in RTF output using the specific language defined for a customer or the default language for an entire application. When you create a language, you must select the spelling dictionary to use in Designer. The engine also uses this spelling dictionary to populate the spelling dictionary in an RTF file.

The language for a customer or application can be defined directly using a language or indirectly through a locale. You can define a specific language or a locale for each customer. You can also define a default language or locale for an entire application. Since a language can be defined in more than one way, the engine selects the RTF spelling dictionary language using the priority shown in the following table:

Priority	Value
1	SYS_LanguageCustomer
2	SYS_LocaleCustomer
3	SYS_LanguageDefault
4	SYS_LocaleDefault
5	The default locale specified on the <b>Basic</b> tab of the application properties

**Note:** If the engine cannot populate the RTF spelling dictionary using one of the previous values, as shown in the table, it selects the default language: English (US).

For more information about using languages and locales, see *System Administration* in the Exstream Design and Production documentation.

In some applications, you must create RTF output in a different language for each customer. If you want to change the spelling dictionary each time the language changes, make sure that you use SYS\_LanguageCustomer or SYS\_LocaleCustomer to define the spelling dictionary language. Then use convenience breaks to create a different document for each customer.

For more information about convenience breaks, see *Creating Output* in the Exstream Design and Production documentation.

If you want to specify a spelling dictionary during production, or if you want to specify a spelling dictionary other than what is available in Exstream Design and Production, you can use the SET\_RTF\_DEFAULT\_LANGUAGE engine switch. This switch lets you specify the language name, the hexadecimal ID, or the decimal ID of the spelling dictionary you want to use. If you use the SET\_RTF\_DEFAULT\_LANGUAGE switch, the language name or ID must match exactly the value accepted in the tag. The engine switch overrides any languages specified in the application and you cannot specify a different spelling dictionary for each customer.

For more information about the SET\_RTF\_DEFAULT\_LANGUAGE engine switch, see *Switch Reference* in the Exstream Design and Production documentation.

### 3.12.4 Viewing RTF output in other programs

When creating the print stream during production, Exstream controls object placement with printer-specific output. With RTF output, however, the RTF viewing software (such as Microsoft Word) controls object placement, which can affect both text wrapping and the number of rows per page for tables.

Page breaks inserted by RTF viewing software might move page numbers placed at the bottom of a page to the next page. Page numbers are not recalculated.

## 3.13 Optimizing XML (Content) output

This section discusses the following topics:

- “[How XML \(Content\) output processes flow settings](#)” below
- “[How design objects are used in XML \(Content\) output](#)” on the next page
- “[How tables are used in XML \(Content\) output](#)” on page 227
- “[How charts and barcodes are processed in XML \(Content\) output](#)” on page 228

For more information about the differences between the XML output drivers, see “[XML](#)” on [page 31](#).

### 3.13.1 How XML (Content) output processes flow settings

Since XML (content) output is a report of what is included in your design, this type of output does not take flow into consideration. This means flow pages are not applicable and do not occur in the output. Everything that falls on flow pages is processed and placed with the original design page.

**Tip:** Exstream extracts the reference name for all applicable objects. To make object identification easier, make sure to use unique names when you name Library and design objects.

Repeating headers and footers, which are used only when you have overflow, are included in their design position since flow does not apply to logical content output.

Important information is stored at the beginning of an XML (content) file. If you are using a non-production key, the "Demonstration Powered by Exstream" message appears as a comment at the beginning of the output.

The output also includes information such as the following:

- Encoding type
- Application name
- Database version
- Date of run
- Output queue name

Notice, in the following example, that the engine lists only one page in the output, since there is no flow.

#### Example

```
<!--*- Demonstration Powered by OpenText Exstream 09/27/2016, Version 9.5.301
-*-->
<dlg:Content schemaVersion="1.0" xmlns:dlg="http://www.hplexstream.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.hplexstream.com Content.xsd">
  <dlg:OutputStream>
    <dlg:Run application="BankStatement" version="505301" run-date="2006-01-
27" queue=
key="NAMEXXX">
      <dlg:Customer key="1: Andrew Nixon" number="1">
        <dlg:Document name="BankStatement">
          <dlg:Page name="Statement First Page">
```

### 3.13.2 How design objects are used in XML (Content) output

Objects composed based on page creation, such as tables of contents, indexes, and cross references, are not included in XML (content) output because they are used to map information to page locations. In addition, graphic objects such as lines, backgrounds, chart shapes, and so on are not included in logical content output.

Objects are sorted and processed in top-down order. Exstream does not sort objects from left to right.

Messages normally fill white space. Because there are no physical pages to fill in XML (content) output, messages that qualify for whitespace frames appear at the bottom of the output file.

All objects, including embedded objects, are placed inline at their design location, no matter how large they are. When dynamic objects are embedded, XML (content) output includes embedded

tags to represent the output. However, if an embedded object is static, Exstream creates it as a separate object. Any relation to other objects does not change the output.

### 3.13.3 How tables are used in XML (Content) output

For tables, XML (content) output includes tags for headers and footers. Applicable tags include the following:

Tag	Description
<none>	A static non-repeating row
<row>	A repeating row
<footer>	A footer row used only once
<rpt-footer>	A footer row that repeats on each page
<rpt-footer-not-last>	A footer row that repeats on each page except the last
<footer-subsections>	A footer row added after all lower-level row sections
<header>	A header row used only once
<rpt-header>	A header row that repeats on each page
<rpt-header-not-first>	A header that appears on each page except the first.
<header-first-on-page>	A header used only with table sections. This applies when the table has sections and this is the first occurrence of the section on the page.
<header-not-first-on-page>	A header used only with table sections. This applies when the table has sections and this is not the first occurrence of the section on the page.
<header-top-flow-frame>	A header used only in flow frames. This applies to the headers if the table is located at the top of the flow frame.
<header-middle-flow-frame>	A header used only in flow frames. This applies to the headers if the table is located anywhere but the top of the flow frame.

For more information about headers and footers in tables, see *Designing Customer Communications* in the Exstream Design and Production documentation.

**Table Example**

```
<dlg:Table name="Table">
  <dlg:Row type="header">
    <dlg:Cell span="3">
      <dlg:Paragraph>SAVINGS ACCOUNT</dlg:Paragraph>
      <dlg:Paragraph>Account number S9903061</dlg:Paragraph>
    </dlg:Cell>
  </dlg:Row>
  <dlg:Row type="rpt-header">
    <dlg:Cell>
      <dlg:Paragraph>Date</dlg:Paragraph>
    </dlg:Cell>
    <dlg:Cell>
      <dlg:Paragraph>Transaction</dlg:Paragraph>
    </dlg:Cell>
    <dlg:Cell>
      <dlg:Paragraph>Amount</dlg:Paragraph>
    </dlg:Cell>
  </dlg:Row>
```

### 3.13.4 How charts and barcodes are processed in XML (Content) output

Charts and barcodes are processed using a DataSet/Datum format. This format requires you to have some knowledge of the data to be able to interpret the output. To help identify information in barcodes, you can include a human-readable identifier ([HRI](#)).

For more information about barcodes, see “[Using Barcodes in Exstream](#)” on page 256.

For example, if you have a pie chart that shows a customer's checking and savings accounts, the DataSet would be the pie chart and the Datum would be the individual values of the variables used to fill the labels and slices. Notice in the following example that the values have not been converted into percentages.

**Chart example**

```
<dlg:DataSet set-type="Pie Chart">
  <dlg:Datum id="PieLabel__" data-type="string">Savings</dlg:Datum>
  <dlg:Datum id="PieAmounts__" data-type="float">5844.950000</dlg:Datum>
  <dlg:Datum id="PieLabel__" data-type="string">Checking</dlg:Datum>
  <dlg:Datum id="PieAmounts__" data-type="float">3118.930000</dlg:Datum>
</dlg:DataSet>
```

# Chapter 4: Managing Resources for the Print Stream

This chapter discusses how to manage resources, such as fonts and images, for output in Exstream.

The way that you set up how the output handles reusable resources (for example, fonts, images, and overlays) can affect file size and processing speed.

This chapter discusses the following topics:

- “[Managing Font Resources](#)” below
- “[Managing Image Resources](#)” on page 235
- “[Using Inline Form Definition File in AFP Output](#)” on page 248
- “[Using Overlays in Output](#)” on page 249
- “[Managing Print Resources](#)” on page 249

## 4.1 Managing Font Resources

Depending on how tightly you must control the design and which fonts are included on your output device or viewing software, you might need to either embed some or embed all of the fonts used in an application in the print stream. For more information about which fonts are included, refer to the documentation for your output device or the viewing software.

In addition, if your application produces DBCS output, you can specify the DBCS font characters that are included in the package file, so that they are available when the engine produces the print stream.

This section discusses two ways you can control how Exstream processes fonts:

- “[Referencing All Fonts](#)” below
- “[Embedding Fonts](#)” on the next page
- “[Identifying the DBCS Font Characters to Include in the Package File](#)” on page 233

### 4.1.1 Referencing All Fonts

If you reference all fonts, none of the fonts in an application are included in a package file. Referencing all fonts can help make the print file as small as possible. Keep in mind that when

you reference fonts, your output might look slightly different from your design. For example, if you plan to send files to customers who might not have the same output device or viewing software, these customers might not have the same fonts. Depending on the output driver, their output device or viewing software might either substitute a different font, or they might receive errors.

To assign a unique name for each font size and style already on the output device, you can use a font name table. You can use font name tables only for referenced fonts.

For more information about font name tables, see *System Administration* in the Exstream Design and Production documentation.

**Note:** For best results, embed fonts rather than reference them when working in a DBCS application. For more information about embedding fonts, see “[Embedding Fonts](#)” below.

To reference all fonts in your output:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. In the **Font usage** list, click **Reference all fonts**.
4. Save the output object.

## 4.1.2 Embedding Fonts

Embedding fonts in the print stream helps you make sure the text is printed exactly as designed. Embedding fonts ensures that all font resources you need are included in the output file, which guarantees portability, or that the output can be converted to different formats. For example, if you embed fonts in AFP output, you ensure that the output is viewable in an AFP output viewer and that the output can be converted to a PDF output file.

Keep in mind that the more fonts you embed, the larger the file size. If all the fonts you need are on the output device, you can reference the fonts and not include them in the package file.

**Note:** For best results, embed fonts rather than reference them when working in a DBCS application.

To embed fonts:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.

3. In the **Font usage** list, select one of the following options to specify which fonts to include in the package:
  - **Embed all fonts**—All the fonts used in an application are embedded in a package file. Using this option results in a larger file size, but it does not affect processing time and provides Exstream the most control over the printed output.
  - **Embed only unnamed fonts**—Only the fonts not included on your output device are included in a package file. Use a font name table to assign a unique name for each font size and style already on the output device.
4. In the **Font type** list, specify the font embedding options.

For more information about font type options, see “[Specifying the Type of Font to Embed in the Package File](#)” below.
5. For PDF and PostScript output, select the **Include standard fonts** check box to include Standard (Base 14) fonts in the print stream. These fonts are not normally needed in the print stream. This option is especially helpful for archiving electronic output or for viewing output in a nonstandard reader (for example, if you are viewing PDF output in a program other than Adobe Acrobat). Adding standard fonts increases the output file size.
6. Save the output object.

## Specifying the Type of Font to Embed in the Package File

Exstream supports any font loaded on your system and converts it to the correct font type for your output device. If you choose to embed fonts, for the best results, embed fonts as Type 1 whenever possible. Exstream builds the characters needed for your application (such as East European or Western) as Type 1 fonts and includes them in the package file. This setting provides the following benefits:

- It reduces file size by packaging only the characters required in the application, instead of building and loading all characters of a font.
- It offers enhanced font metrics that can correct spacing errors with another font type, such as TrueType.

If you must build fonts to load into an output device, include the WRITEPS\_PDFFONTS engine switch when you package the application. The engine writes separate .pfb files for all the Type 1 fonts. For example, you can use the .pfb files to reference the exact fonts in other applications.

For information on engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

For more information on the best font embedding choice, consult the documentation for your output device or viewing software. On the **Resource Management** tab, in the **Font type** list, select one of the following options to specify the type of font to embed in the package file, depending on the output driver you are using:

#### Font type options

Output driver	Available options
AFP	<ul style="list-style-type: none"> <li>• <b>Bitmap</b>—Embeds or references bitmap fonts</li> <li>• <b>Outline</b>—Embeds or references outline fonts</li> <li>• <b>OpenType</b>—References OpenType fonts. This option is available only if you select <b>Reference all fonts</b> in the <b>Font usage</b> list on the <b>Resource Management</b> tab. (DBCS applications only)</li> <li>• <b>Subset Bitmap</b>—Embeds only the characters of a font used in an application. (DBCS applications only)</li> </ul> <p><b>DBCS:</b> If you want to use outline fonts in DBCS applications, you must select the <b>Outline</b> option. You must also embed all fonts. For more information about subsets in AFP output, see <a href="#">"Using Character Subsetting in AFP Output (DBCS Applications)" on the next page</a>.</p>
Metacode	<ul style="list-style-type: none"> <li>• <b>Type 1, 4-word FST</b></li> <li>• <b>Type 1, 5-word FST</b></li> <li>• <b>Type 0, 4-word FST</b></li> <li>• <b>Type 0, 5-word FST</b></li> </ul>
PCL	<ul style="list-style-type: none"> <li>• <b>Bitmap (max 300 dpi)</b></li> <li>• <b>Bitmap (high-resolution)</b></li> </ul>
PDF	<ul style="list-style-type: none"> <li>• <b>Use original font format</b>—Embeds all characters of each original font</li> <li>• <b>Build Type 1 fonts</b>—Embeds the characters in the current Windows code page, up to 256 characters per font, as Type 1 fonts</li> <li>• <b>Build Subset fonts (CFF)</b>—Embeds only those characters in each font used in the output file, which reduces the output file size</li> </ul>
PostScript	<ul style="list-style-type: none"> <li>• <b>Build Type 1 Subset fonts (DBCS applications)</b>—Used to support Type 1 font subsetting in DBCS applications. To reduce file size, font subsetting includes only those characters in each font used in the output.</li> <li>• <b>Build Type 1 fonts</b>—This option is the recommended setting. Exstream builds the characters needed for your application (such as East European or Western) as Type 1 fonts and includes them in the package file.</li> <li>• <b>PostScript Type42</b>—Places a PostScript wrapper around TrueType fonts. The target printer must support Type 42 fonts. If your design uses Type 1 fonts, you cannot use this setting. Exstream does not convert Type 1 fonts to Type 42.</li> <li>• <b>Type 1 with hints</b>—Basic hint processing creates smaller files with shorter processing time.</li> <li>• <b>Type 1 with enhanced hinting</b>—Enhanced hinting creates fonts of better quality when printed at sizes of 6 points and below.</li> </ul>

## Building Type 1 Subset Fonts on z/OS for PostScript Output (DBCS Applications)

When you choose to build Type 1 subset fonts, the engine uses temporary files to generate the output. Before you run the engine, if you are using the z/OS platform, you must allocate a temporary file ( DD:TEMP). The space allocated for your temporary file must be at least as large as your main file.

If you run the engine on z/OS, you cannot write your output directly to spool. At run time, the engine stores the body of the PostScript file in the temporary file, writes the header information to the output file, and then appends the body to the header information. If you write the output to spool, the header information becomes unavailable before the engine can append the body to it.

## Using Character Subsetting in AFP Output (DBCS Applications)

When printing to AFP using output queues, the embedded font characters are subset when packaging. Subsetting is a technique which includes only those characters of a font that are actually used in the output. This creates a more efficient AFP print stream because it ensures that your embedded fonts include only the characters actually used. Packaging times can vary based on your application, but the files are smaller than they would be if you did not use character subsetting.

The application must include a single AFP output queue. Other output queues required must be generated separately from the AFP output that utilizes font subsets for embedded fonts.

### 4.1.3 Identifying the DBCS Font Characters to Include in the Package File

When you package an application that produces DBCS output, use the font resource management feature to ensure that all of the DBCS font characters that your application requires are included in the package file, and that they are available when you then run engine to produce output.

When you design an application that produces DBCS output, you have the following options for determining the DBCS font character set that will be included in the package file:

- **Package all font characters**—When you select the **Package all font characters** check box, the packaging process initiates a pre-packaging engine run in order to determine the set of DBCS characters to include in the package file. The pre-packaging engine run adds the characters from the test data files that are used in your application design into the package file. When the engine then uses the package file and your production data files to produce output, formula calculations or rules can sometimes result in DBCS characters that are required in the output, but were not used in the pre-packaging engine run. As a result, keep in mind that you must make sure that all of the DBCS characters that your application might

require are contained within the test data files that are used in your application.

**Note:** To avoid unpredictable results due to formula calculations or rules, you can clear the **Package all font characters** check box and use the **Additional font character ranges to package** selection area to specify the DBCS font characters that you want to include in the package file.

- **Additional font character ranges to package**—You can use the **Additional font character ranges to package** selection area to specify one or more Unicode character code charts to include in the package file.

**Note:** Keep in mind that when you use the **Additional font character ranges to package** selection area to specify the DBCS characters to include in the package file, all of the DBCS characters in the specified Unicode character sets are included in the package file.

- **FONTDIRECTORIES switch**—You can use the FONTDIRECTORIES engine switch to specify the directory paths for TrueType fonts that are used in content that is dynamically imported into your design. The engine searches the specified directory paths at run time to dynamically include fonts from those locations. For more information about the FONTDIRECTORIES engine switch, see *Switch Reference* in the Exstream Design and Production documentation.

To specify the DBCS font characters that you want to include in the package file, complete one of the following tasks:

To	Do this
Include the set of DBCS characters that the pre-packaging engine run assembles from the test data files that are used in your application in the package file	<ol style="list-style-type: none"><li>1. From Design Manager, drag an application to the Property Panel.</li><li>2. Click the <b>Font Resources</b> tab.</li><li>3. Select the <b>Package all font characters</b> check box.</li><li>4. From the Menu bar, select <b>Edit &gt; Save</b>.</li></ol> <p><b>Note:</b> Keep in mind that you must make sure that the pre-packaging engine run includes all of the DBCS font characters that are required in subsequent engine runs that use the package file.</p>

To	Do this
Include a subset of DBCS font characters in the package file by specifying one or more Unicode character code charts to include in the package file	<ol style="list-style-type: none"> <li>From Design Manager, drag an application to the Property Panel.</li> <li>Click the <b>Font Resources</b> tab.</li> <li>Clear the <b>Package all font characters</b> check box. The <b>Additional font character ranges to package</b> area becomes available.</li> <li>In the <b>Additional font character ranges to package</b> area, click . The <b>Select the script</b> dialog box opens.</li> <li>From the <b>Select a group</b> drop-down list, select a Unicode character code group.</li> <li>From the <b>Select a script</b> list, select a Unicode font character range.</li> <li>Click <b>OK</b>. The <b>Select the script</b> dialog box closes and the script is added to the <b>Unicode code chart</b> list.</li> <li>Repeat step 4 through step 7 as needed.</li> <li>From the Menu bar, select <b>Edit &gt; Save</b>.</li> </ol>

When your application design produces DBCS output, keep in mind the following additional considerations :

- You must specify the alphabetic script and the numeric script associated with the DBCS language.

For more information about configuring language settings for DBCS applications, see *System Administration* in the Exstream Design and Production documentation.

- If your application imports text content at run time, you can use the font resource management feature to ensure that the fonts that are used in the imported content are available for the print stream.

For more information about font resource management, see *Importing External Content* in the Exstream Design and Production documentation.

## 4.2 Managing Image Resources

This section discusses information about managing image resources. You can set options for both static images and imported image resources.

To specify how you want the engine to handle image resources, complete the following tasks as needed:

- “[Specifying Where to Place Static Images in the Print Stream](#)” below
- “[Converting Images for Optimized Viewing](#)” on the next page
- “[Specifying How Image Resolution Settings Are Processed](#)” on page 238
- “[Specifying the Internal Format to Use for Images](#)” on page 239
- “[Using AFP Color Management Architecture Functionality with Image Resources in the Print Stream](#)” on page 241
- “[Referencing External Images for PostScript and VIPP Output](#)” on page 243
- “[Converting Charts and Shapes into Images \(AFP Output Only\)](#)” on page 244
- “[Referencing and Storing Images for HTML, HTML \(Email\), and Multi-Channel XML Output](#)” on page 244

With PPML output, forms and images are raster image processed (ripped) before production starts to enhance the amount of data transferred to the printer. This language inserts images in the print stream by external references (for example, <EXTERNAL\_DATA Src=ftp://exstream.com/logo.eps>). For HTML output, hyperlinks and images can also be encoded in MIME and included in the data transfer.

## 4.2.1 Specifying Where to Place Static Images in the Print Stream

Charts and other dynamically composed images are always placed on each page on which they appear, because they are unique for each customer. Static images, however, can be placed at different places in the print stream. The options you use to specify where static images are placed depend on the output device or viewing software you will use. For example, some output devices can read resources from the top of the file into memory and then reuse them each time the image is called.

To specify where to place static images:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. From the **Image processing** drop-down list, select one of the following options:
  - **Include on each page**—Includes a copy of all the images used in the application in your output file on each page where they occur
  - **Place at top of print stream**—Includes a copy of the images at the top of the print stream for insertion as needed. This option can increase production processing speed.

- **Reference external images**—References images for the application in your print stream. If you use this option, output files will be smaller, but your output device must be able to find the image library.
  - **Make external image files**—Creates external image files. For some printers, this might increase printing speed. This option is available only for PPML output.
4. From the Menu bar, select **Edit > Save**.

For PPML output, if you are using static JPEG images, you must include the resolution information. If this information is not included, and you select both the **Make external image files** option from the **Image** drop-down list and the **Only convert if original format is unsupported** option from the **Image conversion action** drop-down list, then static JPEG images might appear incorrectly in PPML output.

## 4.2.2 Converting Images for Optimized Viewing

You can set up to convert all images to the resolution settings or format that you specify, or you can set up to convert only those images with formats that are not supported in Exstream output. These settings apply only to static images.

To convert static images:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. From the **Image conversion action** drop-down list, select one of the following options:
  - **Convert all images using settings below**—Images are converted based on the options you select from the **Resolution method** drop-down list and the **Format** drop-down list. Use this option if you want all images in the output to be formatted in the same way, based on the format and resolution settings.
  - **Only convert if original format is unsupported**—Images are converted based on the **Resolution method** and **Format** settings only if the original image format is not supported by Exstream for the specified output. Use this option if you want to retain the resolution settings or image format on some images in your output.

For more information about specifying the format of converted images, see “[Specifying the Internal Format to Use for Images](#)” on page 239.

4. From the Menu bar, select **Edit > Save**.

For more information about the image file formats that are supported in Exstream, see *Importing External Content* in the Exstream Design and Production documentation.

### 4.2.3 Specifying How Image Resolution Settings Are Processed

When you import images into Designer, for the best results, make sure the image resolution settings match the output resolution settings. If the image resolution settings are different from the output resolution settings, you will get larger files and experience other problems, such as image quality issues.

For more information about output resolution settings, see “[Managing output resolution in Exstream](#)” on page 37.

To specify how the engine processes image resolution settings:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. In the **Resolution method** list, select one of the following options:
  - **Automatic**—The resolution is calculated using properties set on the **Basic** tab, the **System Settings**, and the image properties.
  - **Printer resolution**—The dots per inch (dpi) setting you select from the **Resolution** drop-down list on the **Basic** tab is used for all resolution settings.
  - **Half resolution**—The print stream resolution is set to one-half of the driver specification. In most drivers, you experience little or no reduction in print quality, but the processing time is reduced.
  - **Specific resolution**—Images are created at the resolution you specify.
4. If you selected the **Specific resolution** option in the **Resolution method** list, specify a resolution for images in the **Specific resolution(DPI)** list.
5. From the Menu bar, select **Edit > Save**.

### Specifying a Custom Image Resolution

If the image resolution options available in the **Resolution method** drop-down list do not meet your business needs, you can specify a custom output resolution for some output drivers.

You can specify a custom image resolution for the following output drivers:

- PDF
- PDF/VT
- PostScript
- PPML

- TOP
- VDX
- VIPP
- VPS

To specify a custom output resolution:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. In the **Resolution method** list, select **Specific resolution**.
4. In the **Specific resolution (DPI)** list, select **Custom**.
5. In the **Custom** box, enter the custom image resolution you want to use. The minimum valid resolution you can set is 72, and the maximum valid resolution is 2400.
6. From the Menu bar, select **Edit > Save**.

#### 4.2.4 Specifying the Internal Format to Use for Images

The internal format you can use for images is dependent on the output driver you use and the color mode you selected. The following table lists the options available for each output driver, depending on the color mode option you select:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. In the **Format** list, depending on the output driver you are using, specify the format to use for images:

Output driver(s)	Options
AFP, if you are printing in black and white, with highlight colors, or with AFP IOCA color output	<ul style="list-style-type: none"><li>• IOCA G4 compressed</li><li>• IOCA B&amp;W uncompressed</li><li>• IM1 (old PSEG)</li></ul>
AFP, if you are printing in full color	<ul style="list-style-type: none"><li>• FS45 (JPEG)</li><li>• FS45 (lossless)</li></ul>

Output driver(s)	Options
HTML <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <b>Note:</b> You can manage image resources for HTML output only if you are delivering container designs in the output.             For more information about delivering container designs in HTML output, see "<a href="#">Setting Up HTML Output</a>" <a href="#">on page 91</a>.         </div>	<ul style="list-style-type: none"> <li>• <b>JPEG</b></li> <li>• <b>GIF</b></li> </ul>
HTML (email)	<ul style="list-style-type: none"> <li>• <b>JPEG</b></li> <li>• <b>GIF</b></li> </ul>
PDF or PostScript, if you are printing in black and white, in grayscale, or with highlight colors	<ul style="list-style-type: none"> <li>• <b>B&amp;W uncompressed</b></li> <li>• <b>CCIT-group 4</b></li> </ul>
PDF or PostScript, if you are printing in full color	<ul style="list-style-type: none"> <li>• <b>JPEG</b></li> </ul>
XML (multi-channel)	<ul style="list-style-type: none"> <li>• <b>JPEG</b></li> <li>• <b>GIF</b></li> </ul>

For more information about color mode, see "[Specifying the Way Color is Produced in Output](#)" [on page 40](#).

The older IM1 (PSEG) image format is supported for AFP printers. Select the **IM1 (old PSEG)** option if your output device cannot support IOCA formats.

4. From the Menu bar, select **Edit > Save**.

## Forcing G4 Compression for Images

By default, the engine calculates whether to compress each image depending on whether the image will be smaller by using G4 compression. If it will not be smaller, then the engine does not compress the image. However, some third-party viewers and production software require that all images be compressed.

This task applies to the following output drivers:

- AFP
- PDF
- PostScript

To force G4 compression:

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. For AFP output, select the **IOCA G4 compressed** option in the **Format** list. For PDF or PostScript output, select the **CCITT – Group 4** option.
4. If you want to compress all images, regardless of image size, select the **Force G4 on all images** check box.
5. Save the output object.

For AFP output, if you also select the **Convert charts and shapes into images** check box on the **Resource Management** tab, Exstream will compress the charts and shapes unless the result of the compression is that the image is larger. In this case, non-compressed images will be used to optimize the size of the print stream.

For more information about converting charts and shapes into images, see “[Converting Charts and Shapes into Images \(AFP Output Only\)](#)” on page 244.

#### 4.2.5

### Using AFP Color Management Architecture Functionality with Image Resources in the Print Stream

If you are printing AFP output in full color mode, you can specify that Exstream include images as object container resources in the AFP print stream. Object containers are used to pass original image data to the printer, which allows the printer to handle images natively and typically results in the highest possible quality output. By including container resources, you can also embed International Color Consortium (ICC) profiles to manage color for images. If an image contains an ICC profile, Exstream creates an inline Audit Color Conversion Color Management Resource (CMR) for that ICC profile and applies the CMR to the image in which it is embedded. CMR color conversion helps the printer render the image correctly.

**Important:** Keep in mind that not all AFP devices support the inclusion of native PDF content that was imported into the output at run time.

For more information about output devices that do not natively support PDF content, see *Importing External Content* in the Exstream Design and Production documentation.

To manage AFP image resources in the print stream:

1. In Design Manager, from the Library, drag an AFP output object to the Property Panel.
2. Click the **Basic** tab.

3. From the **Color mode** drop-down list, select **Full color**.
4. Click the **Resource Management** tab.
5. To control how both design-time images and run-time images are included in the AFP print stream, do one of the following:

To	Do this	Details
Disable AFP Color Management Architecture™ (ACMA™)	Clear the <b>Use AFP color management architecture</b> check box.	<ul style="list-style-type: none"> <li>• All imported images use the following import conversions:           <ul style="list-style-type: none"> <li>• By default, PDF images are converted to JPEG. If you want PDF images to maintain the PDF format in an object container, then you can include the PDF_PASSTHROUGH_AS_PDF engine switch in your control file. For additional information about the PDF_PASSTHROUGH_AS_PDF engine switch, see <i>Preparing Applications for Production</i> in the Exstream Design and Production documentation.</li> </ul> </li> <li>• All other images are either wrapped natively in the FS45 format (such as JPEG, EPS, or TIFF) or are ignored.</li> </ul>
Enable ACMA	Select the <b>Use AFP color management architecture</b> check box.	<ul style="list-style-type: none"> <li>• All imported images are placed in AFP object containers which carry non-object content architecture objects (OCA).</li> <li>• This option is suggested for high quality printing because the original image data is passed directly to the printer.</li> <li>• The following import formats are supported for full-color AFP output: EPS, JPEG, PDF, and TIFF.</li> </ul>

6. To specify where to place static images that are imported at design time, do one of the following:

To	Do this	Details
Include design-time images at the top of the print stream	In the <b>Image management</b> area, from the <b>Image processing</b> list, select <b>Place at top of print stream</b> .	This option can improve processing speed. Keep in mind, however, that this option can also increase file sizes.
Reference design-time images for the application in the print stream	In the <b>Image management</b> area, from the <b>Image processing</b> list, select <b>Reference external images</b> .	If you use this option, output files will be smaller. Keep in mind, however, that your output device must be able to find the image library.
Include a copy of the design-time images in the output file on each page where the images occur	In the <b>Image management</b> area, from the <b>Image processing</b> list, select <b>Include on each page</b> .	This option is useful where caching limits apply. Keep in mind, however, that this option can also increase the size of the output file and can cause slower processing speeds.

7. To specify where to place dynamic images that are imported at run time, do one of the following:

To	Do this	Details
Include all run-time images in the print stream inline	In the <b>File resource management</b> area, from the <b>Resource inclusion</b> list, select <b>Default</b> .	As a best practice, use this option if you are importing large quantities of unique images.
Include only used run-time images in the print stream at the top of the file	In the <b>File resource management</b> area, from the <b>Resource inclusion</b> list, select <b>Used resources only</b> .	This option can help to reduce file sizes and optimize printing speed. This option is useful for applications that contain reused images.

## 4.2.6 Referencing External Images for PostScript and VIPP Output

If you want to reference external images for PostScript and VIPP output, you must create image resource placeholder variables. You must also enable the GetTiff function on the printer. You can enable the GetTiff function only if you have licensed the Dynamic Content Import module.

For more information about placeholder variables, see *Importing External Content* in the Exstream Design and Production documentation.

For information about using the GetTiff function, consult the documentation for your output device.

To reference external images for PostScript and VIPP output:

1. Make sure all the placeholder variables you use in the application are set correctly:
  - a. In Design Manager, from the Library, drag a placeholder variable to the Property Panel.
  - b. Click the **Placeholder** tab.
  - c. From the **Placeholder method** drop-down list, make sure the option **Image resource** is selected.
  - d. Save or close the placeholder variable.
2. In Design Manager, from the Library, drag a PostScript or VIPP output object to the Property Panel.
3. Click the **Resource Management** tab.
4. From the **Placeholder handling** drop-down list, select one of the following options:
  - **None**—Specifies file path names to have TIFF placeholder variables import specific images into the print stream
  - **Use GetTiff**—Specifies the path name of a resource so the printer imports TIFF images at run time. For this option to work correctly, the placeholder variable type must have been set to **Image resource** when the variable was created, and you must map the

variable to the correct path name for the TIFF resource.

5. From the Menu bar, select **Edit > Save**.

**Caution:** Because the engine inserts the path name of the images in the print stream as printer-specific resources, referenced images do not appear in the Exstream Viewer.

## 4.2.7 Converting Charts and Shapes into Images (AFP Output Only)

If a chart or table row designed to have gray shading appears in color or with unexpected patterns, or if you cannot use Graphics Object Content Architecture (GOCA) objects in your AFP print stream, you can specify that charts and shapes are converted to images. If you use this feature, Exstream creates graphic drawings as rasterized patterns rather than as GOCA graphics. You could also choose to create each page as an image, although this feature significantly increases the size of the print file.

For more information about creating pages as images, see “[Creating Pages as Images](#)” on [page 44](#).

If you use highlight colors in your output, you cannot convert charts and shapes into images.

To convert charts and shapes into images:

1. In Design Manager, from the Library, drag an AFP output object to the Property Panel.
2. Click the **Resource Management** tab.
3. Select the **Convert charts and shapes into images** check box.
4. From the Menu bar, select **Edit > Save**.

## 4.2.8 Referencing and Storing Images for HTML, HTML (Email), and Multi-Channel XML Output

When you include images in your design for HTML, HTML (email), and Multi-Channel XML output, you can specify the way that images are referenced and stored for the output.

If you want to reference images at an external location, you can either specify unique locations for each image or specify a single location that applies to all images in the design. You can specify these reference locations using a relative path, absolute path, or a variable.

If you do not want to store the images at an external location, you can include images from the application in the output directory with the output file. When you include images with the output file, the engine includes the image names in the output file, creates the image files in your application, and places the image files in the output directory.

You can manage image resources for HTML output only if you are delivering container designs in the output.

For more information about including container designs in HTML output, see “[Setting Up HTML Output](#)” on page 91.

This section discusses the following topics:

- “[Design Considerations for Referencing and Storing Images](#)” below
- “[Specifying a Directory Path or URL for All Images](#)” below
- “[Specifying a Directory Path or URL for Individual Images](#)” on the next page
- “[Including All Images with the Output File](#)” on page 248

## Design Considerations for Referencing and Storing Images

Keep the following in mind when referencing and storing image resources for HTML, HTML (email), and Multi-Channel XML output:

- If you reference images in your application, production runs will be much faster than if you create and place the images in the output directory.
- Charts, shapes, and barcodes are automatically converted to images for HTML, HTML (email), and Multi-Channel XML output. The output resolution setting is applied to these images, and they are included with the output file in the specified output directory path. Because internally created images will not have paths associated with them in the output file, these images can be referenced with relative paths if the output files are moved after production.
- If you use placeholder images, only JPEG and PNG images are passed through in their original format. PDF is converted to JPEG, and all other formats are unsupported.
- In Multi-Channel XML output, if you specified a URL, the URL for an image is placed in HTML tags for HTML output and in plain text for plain text output. For example, for output that will be transformed into an HTML email, the URL will be in HTML tags so that the email client can pull the image into the email. For plain text Multi-Channel XML output, the URL will be included in plain text and can be copied into the address bar.

## Specifying a Directory Path or URL for All Images

You can specify a default path or URL that is applied to all of the images for HTML, HTML (email), or Multi-Channel XML output. By setting a default path (either as a static value or as a variable), you can enter the location for images once rather than entering the same information in each individual image. Specifying a directory path or URL for all images can save time if your application contains many images or if all images must be stored in the same location. If you specify a default path or URL for all images, the engine uses the location that you specified for all images and ignores any settings that are applied to individual images.

For more information about specifying a directory path or URL from individual images, see [“Specifying a Directory Path or URL for Individual Images” below.](#)

To specify a directory path or URL for all images:

1. In Design Manager, from the Library, drag an HTML, HTML (email), or XML (multi-channel) output object to the Property Panel.

The output properties open in the Property Panel for you to define.

2. Click the **Resource Management** tab.

3. To define the default path for all images, complete one of the following sets of steps:

To	Do this
Specify a URL, absolute path, or relative path to an existing image	<ol style="list-style-type: none"><li>From the <b>Image path</b> list, select <b>Use a static value to define the default path</b>.</li><li>In the <b>Location path</b> box, enter the path to use as a default location for all images.</li></ol>
Specify a variable to define the directory path at run time	<ol style="list-style-type: none"><li>From the <b>Image path</b> list, select <b>Use a variable to define a default path</b>.</li><li>In the <b>Location path variable</b> box, enter the path to use as a default location for all images.</li></ol>

4. From the Menu bar, select **Edit > Save**.

## Specifying a Directory Path or URL for Individual Images

If you want more control over where each individual image is stored, you can specify a directory path or URL for each individual image that you include in HTML, HTML (email), or Multi-Channel XML output.

For more information about specifying a directory path or URL for all images, see [“Specifying a Directory Path or URL for All Images” on the previous page.](#)

To specify a directory path or URL for individual images:

1. In Design Manager, from the Library, drag an HTML, HTML (email), or XML (multi-channel) output object to the Property Panel.

The output properties open in the Property Panel for you to define.

2. Click the **Resource Management** tab.

3. From the **Image path** drop-down list, select **Use paths from individual images**.

4. From the Menu bar, select **Edit > Save**.

5. In Designer, right-click an image and select **Image Properties**.

The **Image Properties** dialog box opens.

6. Click the **Image** tab.
7. Do one of the following to specify the how images are handled in the output:

To	Do this
Instruct Exstream to place the image files at the same location as the output file and include only the image name in the output file	<ol style="list-style-type: none"> <li>a. From the <b>Image path location</b> list, select <b>Use output directory path</b>.</li> <li>b. In the <b>Image name</b> box, enter a name for the image. Make sure you use a name that will help you easily identify the image that is being used in the output file.</li> </ol>
Specify a URL or an absolute or relative path to an existing image	<ol style="list-style-type: none"> <li>a. From the <b>Image path location</b> list, select <b>Use specified URL/path</b>.</li> <li>b. In the <b>User-specified path</b> box, enter the URL or directory path you want to use.</li> <li>c. If the URL or directory path you specified does not already include the file name of the image, enter the file name in the <b>Image name</b> box. This file name is appended to the specified URL or path. (If the URL or path already contains the file name, leave this box blank.)</li> </ol>
Specify a variable to indicate the directory path at run time	<ol style="list-style-type: none"> <li>a. From the <b>Image path location</b> list, select <b>Use specified variable</b>.</li> <li>b. In the <b>Path variable</b> box, click .</li> <li>c. Select the variable that you want to use to specify the directory path.</li> <li>d. If the URL or path specified by the variable does not already include the file name of the image, enter the file name in the <b>Image name</b> box. This file name is appended to the URL or path specified by the variable. (If the URL or path already contains the file name, leave this box blank.)</li> </ol>

**Tip:** If you select **Use specified URL/path** or **Use specified variable**, you can specify or use a variable that specifies the Adaptive Media Delivery (AMD) URL for an image from OpenText Media Management (OTMM).

For more information about using AMD with OTMM, see the OpenText Media Management documentation.

8. Click **OK**.

The **Image Properties** dialog box closes.

9. Repeat step 5 through step 8 for all images intended for HTML, HTML (email), or Multi-Channel XML output.
10. From the Menu bar, select **File > Save**.

## Including All Images with the Output File

For HTML, HTML (email), and Multi-Channel XML output, you can specify that the engine place image files with the output file. If you place image files with the output file, the engine ignores any image path settings on the image properties in Designer.

To include all images in an application with the output file:

1. In Design Manager, from the Library, drag an HTML, HTML (email), or XML (multi-channel) output object to the Property Panel.
2. Click the **Resource Management** tab.
3. Select the **Always include images with output** check box.
4. From the Menu bar, select **Edit > Save**.

## 4.3 Using Inline Form Definition File in AFP Output

Form definition (formdef) files are resources that define the general characteristics of the physical printed page, including duplexing, static overlays, and the number of copies. You can create an inline formdef in Exstream and include it at the top of the AFP print stream.

If you choose to create output with both simplex and duplex pages, you must create an inline formdef.

For more information about simplex and duplex options, see “[Using Simplex and Duplex in Output](#)” on page 39.

To create an inline form definition file:

1. In Design Manager, from the Library, drag an AFP output object to the Property Panel.
2. Click the **Resource Management** tab.
3. Select the **Create and include Formdef in print file** check box.
4. In the **Formdef Name** box, enter the name for the formdef file. The formdef file name cannot exceed eight characters. You can also use the default name of FORMDEFR.
5. Save the output object.

## 4.4 Using Overlays in Output

When you use overlays, the engine writes recurring static objects only at the top of the file and places them where needed. These overlays can enhance processing speed because the engine needs to perform fewer disk writes. Also, objects are reused so the engine calculates and draws fewer objects.

Use overlays if pages are reused in the print file in a large run. If you produce unique output for each customer, overlays increase file size and processing time with no benefit. Using overlays enables the printer to save top-of-stream images to its hard drive during an output run, which improves print performance.

PCL and Metacode output do not use overlays.

### 4.4.1 Specifying Overlay Settings

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. From the **Overlay** drop-down list, select one of the following options:
  - **Do not create overlays**—The engine creates pages for each customer.
  - **Create and include in print file**—The engine creates virtual overlays and stores them at the top of the print stream for use where needed. If you use overlays, then you cannot use accessibility tags in the same PDF output.
4. In the **Maximum overlays** box, enter the maximum number of overlays to allow on the output device to control your file size and processing time more precisely. If your output device can handle a specific number of overlays, enter a number between 1 and 99,999. If there is no limit, use the default setting of 99,999.
5. Save the output device.

## 4.5 Managing Print Resources

With Exstream, you can choose to include all available resources (for example, fonts, images, and overlays) in your output, or you can include only the resources that are used.

You can set up how to use print resources in the following ways:

- **Include only used resources in output with the resource inclusion feature**—You can choose either to include all resources (which might increase engine performance), or to include only those resources you use in your application (which can result in a much smaller output file).
- **Manage print resources with a Resource Manager file**—For some output drivers, you can use Resource Manager (ResManager) files to include resources.
- **Store named files on a PostScript output device**—For some PostScript-based output devices, you can name and then write resource files to an internal hard drive on the printer instead of using production environment memory.

### 4.5.1 Including Only Used Resources in Output

1. In Design Manager, from the Library, drag an output object to the Property Panel.
2. Click the **Resource Management** tab.
3. To control which resources are included in your output, select one of the following options from the **Resource inclusion** drop-down list:
  - **Default**—All available resources that are designed in your application are included in the output file, even if they are not used. For example, if you have used rules to exclude some content, the resources in that content are still included in the output file.
  - **Used resources only**—Only the resources used by your application are included in the output file. If you select this option, programs can reference resources as needed rather than loading all available resources for each page. The **Used resources only** option can slow the engine run; however, it can also reduce the file size, depending on the number of resources used by your application. If you select the **Used resources only** option, you do not need to use a Resource Management file.
  - **Used resources at page level**—Include used resources in your output on a page-by-page basis so that programs can reference resources as needed rather than loading all of the available resources for each page. As a best practice, if you intend to use overlays in your application or if you intend to deliver pages that were previously imported into Design Manager, select the **Used resources at page level** option. This option is available only for PDF, PDF/A, and VDX output.
4. Save the output object.

**Caution:** If you select **Used resources only**, the 'SYS\_BytelnQueue', 'SYS\_BytelnBreak', 'SYS\_CustomerBegByte', 'SYS\_RecordInQueue', 'SYS\_RecordInBreak', and 'SYS\_CustomerBegRecord' system variables do not produce valid values during an engine run.

## Using Resource Inclusion on z/OS

When you use the **Used resources only** option in Design Manager, the processing engine uses temporary files to generate the output. If you are using the z/OS platform, you must allocate a temporary file ( DD : TEMP) before running the engine. The space allocated for your temporary file must be at least as large as your main file.

If you select the **Used resources only** option and run the engine on z/OS, you cannot write your output directly to spool. At run time, the engine stores the body of the AFP file in the temporary file, writes the header information to the output file, and then appends the body to the header information. If you write the output to spool, the header information becomes unavailable before the engine can append the body to it.

### 4.5.2 Managing Print Resources with a Resource Manager File

A Resource Manager (ResManager) file lets you control which resources the engine places at the top of the print stream when you use external content in an application. This file lets you fine-tune an application for better performance and file size. It is especially useful if you have dynamically imported image files and page, message, or template overlays.

For more information about using external content, see *Importing External Content* in the Exstream Design and Production documentation.

For more information about overlays, see “[Using Overlays in Output](#)” on page 249.

Typically, images on the page are automatically placed at the top of the output file. Images for placeholder variables, however, are handled when the page is composed. This means that the images are placed on each page used, which can potentially result in large files.

In a Resource Manager file, you list placeholder variables that will be pulled during the run. At run time, the engine reads the list and places these images at the top of the print stream. If a placeholder image is not found in the list, then the engine places the image on the page.

**Note:** PPML output does not support scaling (resizing) of dynamic reusable objects. Images imported using a ResManager file will be placed at 100 percent of their original size.

For HTML output, ResManager files can prevent duplicate images from being created in the output directory. If you do not use a ResManager file, then each time the same referenced image file is encountered during an engine run, a new duplicate image is created.

For more information about referencing images, see *Importing External Content* in the Exstream Design and Production documentation.

## Creating a Resource Manager File

The ResManager file is a standard new-line terminated text file. Each line is consistent with the following format: C:\Path\filename.EXT[,objecttype]. If you have licensed the Output Sorting and Bundling module, you can automatically create a ResManager file using the CREATE\_RESMANAGERFILE engine switch.

For more information about using engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

The path name and file name are case-sensitive across all platforms. The following table describes the available object parameters for each output driver:

Output driver	Object parameter and description
HTML	<ul style="list-style-type: none"><li>DYNAMIC—Specifies a black-and-white PNG file name. This parameter is the default setting.</li><li>JPEG_PT—Specifies a file name for JPEG images that can be dynamically imported at run time (JPEG pass-through).</li><li>PNG_PT—Specifies a file name for PNG images that can be dynamically imported at run time (PNG pass-through).</li><li>PDF_PT—Specifies a file name for PDF content that can be dynamically imported at run time (PDF pass-through).</li></ul>
IJPD5	<ul style="list-style-type: none"><li>DYNAMIC—Specifies a black-and-white PNG or TIFF file name. This parameter is the default setting.</li><li>JPEG_PT—Specifies a file name for JPEG images that can be dynamically imported at run time (JPEG pass-through).</li><li>MESSAGE—Uses a message as an overlay</li><li>PAGE—Uses a page as an overlay</li><li>TEMPLATE—Uses a template as an overlay</li><li>TIFF_PT—Specifies a file name for TIFF images that can be dynamically imported at run time (TIFF pass-through).</li><li>PDF_PT—Specifies a file name for PDF content that can be dynamically imported at run time (PDF pass-through).</li></ul>
Metacode	<ul style="list-style-type: none"><li>DYNAMIC—Specifies a black-and-white TIFF file name</li><li>PDF_PT—Specifies a file name for PDF content that can be dynamically imported at run time (PDF pass-through).</li></ul>

Output driver	Object parameter and description
PostScript and PostScript-based	<ul style="list-style-type: none"> <li>• DYNAMIC—Specifies a black-and-white PNG or TIFF file name. This parameter is the default setting.</li> <li>• EPS_PT—Specifies a file name for EPS content that can be dynamically imported at run time (EPS pass-through).</li> <li>• JPEG_PT—Specifies a file name for JPEG images that can be dynamically imported at run time (JPEG pass-through).</li> <li>• MESSAGE—Uses a message as a form</li> <li>• PAGE—Uses a page as a form</li> <li>• TEMPLATE—Uses a template as a form</li> <li>• TIFF_PT—Specifies a file name for TIFF images that can be dynamically imported at run time (TIFF pass-through).</li> <li>• PDF_PT—Specifies a file name for PDF content that can be dynamically imported at run time (PDF pass-through).</li> </ul>
XML (composed)	<ul style="list-style-type: none"> <li>• DYNAMIC—Black-and-white PNG or TIFF file name. This parameter is the default setting.</li> <li>• EPS_PT—Specifies a file name for EPS content that can be dynamically imported at run time (EPS pass-through).</li> <li>• JPEG_PT—Specifies a file name for JPEG images that can be dynamically imported at run time (JPEG pass-through).</li> <li>• MESSAGE—Uses a message as a form</li> <li>• PAGE—Uses a page as a form</li> <li>• PNG_PT—Specifies a file name for PNG images that can be dynamically imported at run time (PNG pass-through).</li> <li>• TEMPLATE—Uses a template as a form</li> <li>• TIFF_PT—Specifies a file name for TIFF images that can be dynamically imported at run time (TIFF pass-through).</li> <li>• PDF_PT—Specifies a file name for PDF content that can be dynamically imported at run time (PDF pass-through).</li> <li>• AFPPSEG_PT—Specifies a file name for a PSEG (AFP page segment) that can be dynamically imported at run time (AFP PSEG pass-through).</li> </ul>

## Referencing Images in the Resource Manager File

Any image requested by the print stream that does not appear in the ResManager file is placed on the page each time it is used. The first 10 times this happens, the engine generates an information message. If you list dynamic images in the ResManager file, they must meet the following criteria:

- Have absolute paths
- Match object names exactly
- Be reachable from the print stream

- Be in an accessible format for the output device or viewing software
- Be set up with placeholder variables

## Referencing Forms in the Resource Manager File for PostScript and XML (Composed)

For PostScript and XML (composed) output drivers, the engine embeds all form types unless you list them in the ResManager file. If you list forms, the engine embeds only the types of forms listed.

Improper form references can result in missing static page content or an unusable print stream. Depending on the output driver you use, if an engine run requests a form that does not appear at the top of the file, the output driver might continue processing with missing static content or the output driver might stop.

## Referencing Overlays in the ResManager File

The engine embeds all overlay types unless you list them in the ResManager file. If you list overlays, the engine embeds only the types of overlays listed.

Improper overlay references can result in missing static page content or an unusable print stream. If an engine run requests an overlay that does not appear at the top of the file, some output drivers continue processing with missing static content, but other drivers stop.

## Using a Resource Manager File

To identify which ResManager file to use, use the RESMANAGERFILE engine switch. You must include the absolute path (unless the file is in the same directory as the engine). This file must contain a sequential list of images with image paths that match exactly to the values of the placeholder variables you use. Otherwise the images will be placed on the page and not at the top of the print stream.

The engine reads the list of images exactly and adds all images to the top of the print stream. If you use 100 images but list 500, then all 500 images appear at the top of the print stream.

**Tip:** Use the REPORT\_RESMAN engine switch to add detail to the engine message file regarding any failure to import files in your Resource Manager list. This switch works with both the local and production engines.

If you set up for output sorting and bundling with two engine runs, both pre-sort and post-sort, you must define the RESMANAGERFILE switch for the post-sort engine run.

For more information about output sorting and bundling, see “[Setting Up Output for Sorting and Bundling](#)” on page 335.

### 4.5.3 Storing Named Files on a PostScript-Based Output Device

For some PostScript-based output devices, you can name and then write files to an internal hard drive on the printer instead of using the production environment memory. The Exstream engine creates named files that are stored either on the printer hard drive or in memory files, depending on the output driver. Only PostScript output devices can store the named files on both the printer hard drive and in the memory files. The following table lists which type of output device can use which storage location type:

Output driver	Storage location
PostScript	<ul style="list-style-type: none"><li>Printer hard drive</li><li>Production environment memory</li></ul>
PPML	Printer hard drive only
TOP	Production environment memory only
VIPP	Production environment memory only
VPS	Printer hard drive only

To store named files on the hard drive of the PostScript-based output device:

1. In Design Manager, from the Library, drag a PostScript output object to the Property Panel.
2. Click the **Resource Management** tab.
3. Select the **Supports named files** check box. If you clear this check box, Exstream uses the production environment memory files.
4. Save the output object.

# Chapter 5: Using Barcodes in Exstream

A barcode is a standard method used to represent data to be read by processing and tracking equipment. In Exstream, barcodes support high-speed routing, delivery, and verification services to improve business communications. You can create and customize many different barcode types to work with your scanning equipment and to fulfill your business needs. Perhaps the most familiar use for barcodes are the various barcodes used by postal services around the world. In Exstream, you can also use barcodes for more specialized tasks, such as controlling inserter devices and facilitating tracking features.

For more information about inserters, see [“Using Inserter Objects to Add Messages and Inserts During Print Production” on page 290](#).

For more information about tracking, see *Managing Marketing Messages* in the Exstream Design and Production documentation.

This chapter discusses the following topics:

- “Barcode Types” below
- “Creating Barcodes” on page 267
- “Specifying How a Barcode is Generated” on page 269
- “Specifying the Placement and Appearance of a Barcode Object” on page 271
- “Specifying the Content of a Barcode Object” on page 281
- “Previewing and Printing a Barcode” on page 286
- “Using Barcodes in Your Design” on page 287

## 5.1 Barcode Types

The various types of barcodes supported by Exstream have different requirements for properties such as size of bars, thickness (or weight) of bars, position on the page, and support of readable characters. To help you determine which types are best for your business needs, the following sections describe each barcode type that Exstream offers:

- “3 of 9” on the next page
- “Code 128-Based” on the next page
- “DataMatrix 2D” on page 259
- “EAN-8” on page 259
- “Four State” on page 260

- “General Purpose” on page 261
- “Interleaved 2 of 5” on page 261
- “Japanese Postal” on page 262
- “Modified Plessey” on page 263
- “OMR and GBR OMR” on page 264
- “PDF 417” on page 265
- “POSTNET” on page 265
- “QR Code” on page 266
- “UPC” on page 266

### 5.1.1 3 of 9

3 of 9 barcodes are often called 39 or Code 39 barcodes. These barcodes are widely used and can be decoded with virtually any barcode reader. 3 of 9 barcodes consist of five bars per character and can include alphanumeric characters 0–9 and capital letters from A–Z, plus some symbol characters (for example: \* or +) used to mark the start and end of the barcode.

#### 3 of 9 barcode example



\*0123456789ABCDEF\*

A standard length of the 3 of 9 barcode is 15 bytes, consisting of three Content Type identities, each five digits long. Integer values are right-justified in the barcode and padded with leading zeros to fill in the proper number of digits. The second group is a string consisting of all alphabetical characters. Exstream pads this data with trailing spaces (~) and prints the characters from right to left. Any text larger than the number of digits is truncated from the left. The third group is a string consisting of mixed alphabetical and numeric characters.

### 5.1.2 Code 128-Based

Code 128-based barcodes produce variable-length codes using three different symbol sets: two (A and B) that represent the full ASCII character set and a third (C) for paired digits. Exstream provides the ability to use both Code 128 barcodes and EAN 128 barcodes. EAN 128 barcodes are a subtype of Code 128.

This section discusses the following topics:

- “Code 128” below
- “EAN 128” below

## Code 128

All Code 128 barcodes must begin with a specific start character corresponding to the barcode character set (start A, start B, or start C), and must end with a stop code. You can manually switch between barcode subsets in the middle of a barcode by including the start symbol of a different character set.

### Code 128 barcode example



12345670

Symbol set A is used to print uppercase letters, numbers, and the standard ASCII control characters. Symbol set B is used to print uppercase and lowercase letters, numbers, and punctuation. Symbol set C encodes pairs of numbers 00 through 99.

In the barcode, each character is three bars and three spaces. This means that in A and B modes, each set of three bars and three spaces represents one character; however, in C mode, each set of three bars and three spaces represents two digits. This makes C mode operation twice as dense as A or B mode.

Mode	Supports	Data representation
A	Alphanumeric: uppercase letters, numbers, ASCII control characters	Each set of 3 bars and 3 spaces represents one character in the data.
B	Alphanumeric: uppercase and lowercase letters, numbers, and punctuation	Each set of 3 bars and 3 spaces represents one character in the data.
C	Numeric: numbers 00 through 99	Each set of 3 bars and 3 spaces represents two digits in the data.

## EAN 128

EAN 128 barcodes (also known as GS1-128 barcodes) are a variation of the Code 128 barcode. EAN 128 barcodes add a Function 1 (FNC 1) character after the start code to uniquely identify this barcode as a shipping code.

The FNC 1 requirement is the only major difference between EAN 128 and Code 128 barcodes in Exstream. The FNC 1 character enables scanners and processing software to automatically discriminate between EAN-128 and other barcode symbologies. This enables scanners and processing software to process only relevant data.

The required map string must have least three characters to identify the start, stop, and FNC1 characters. You can map the complete character set, but you must always begin with the start, stop, and FNC1 characters.

The composed EAN 128 barcode consists of a leading quiet zone, a Code 128 Start character (A, B, or C), an FNC 1 character, data (an application identifier plus a data field), a checksum character, a stop character, and a trailing quiet zone.

### 5.1.3 DataMatrix 2D

DataMatrix 2D barcodes, also known as 2D DataMatrix barcodes, are two-dimensional barcodes that have greater capacity for storage than standard barcodes. The size of the square or rectangular barcode varies depending on your output device resolution and scanner capabilities. These barcodes are especially useful for marking small items.

Note: One type of Datamatrix 2D barcode, the Royal Mail Mailmark barcode, is produced in Exstream by using the MAILMARK engine switch. For more information about the MAILMARK engine switch, see *Switch Reference* in the Exstream Design and Production documentation.

#### DataMatrix 2D barcode example



These barcodes are made up of cells (also called modules or data regions) that are arranged in a square or rectangular pattern. The bottom border and left border of the barcode are solid lines, forming what is called the "finder pattern." Barcode readers use the finder pattern to locate and orient the symbol. The top border and right border are broken into alternating dark and light modules, forming what is called the "timing pattern." Barcode readers use the timing pattern to read the row and column count of the cells in the symbol, which contains the data. The cells in the symbol are surrounded by a border called a "quiet zone," which is not read by the barcode reader.

DataMatrix barcodes require a 2D scanner. They cannot be read using a linear barcode scanner.

### 5.1.4 EAN-8

EAN-8 barcodes encode seven digits of numeric message data, along with a single check digit, for a total of eight digits of barcode data. EAN-8 barcodes are designed to use as little space as possible and are most commonly used to identify small packages for retail purposes (for example, a package of chewing gum).

#### EAN-8 barcode example



EAN-8 barcodes include four digits in each of the left and right halves of the barcode. The first two or three digits identify the numbering authority, and the remaining digits identify the product.

The numbering authority and product codes are printed just below the barcode.

### 5.1.5 Four State

The Four State barcode, also called the Four-State Customer Barcode (4CB), combines the sorting and tracking capabilities offered by a number of different barcodes into one. Several standards exist for this versatile barcode type, all making use of four distinct bar sizes. The particular Four State Barcode generated by Design Manager complies with the requirements set by the United States Postal Service.

#### Four State barcode example



The Four State barcode's distinctive four-bar composition significantly expands the amount of information active in a minimum amount of space.

The 65-bar Four State Barcode is only three bars wider than the 11-digit POSTNET version, yet this barcode can uniquely identify up to a billion mail pieces per mailing with support for multiple handling services. For manual mail handling, you can add an optional [HRI](#).

For information about the content requirements of a Four State barcode, see ["Four State Barcode Content Requirements"](#) below.

### Four State Barcode Content Requirements

Four State barcodes support numbers. The engine does not accept spaces, dashes, symbols, or letters. The information you supply in variables must be 20, 25, 29, or 31 (the maximum) digits long and cannot be fewer than 20 digits. If you try to save the barcode with more than 31 digits or fewer than 20 digits, you receive an error message.

No checksum options are necessary. The system conducts error checks during the encoding.

The barcode contains the following two sections:

- **Customer tracking code**—This section is made up of 20 digits. Leading or trailing zeros can be added to fill the spaces.

- **Routing code**—This code is made up of 0, 5, 9, or 11 digits. You cannot use leading or trailing zeros to fill the spaces.

The following table describes the digit sequence for the tracking code in Four State barcodes:

Digit sequence for the tracking code in Four State barcodes

Digits	Description	Limitations
First 2	Barcode identifier	The second digit must be between 0 and 4, as assigned by the United States Postal Service.
Next 3	Special services	The valid range is 000–999, as assigned by the United States Postal Service.
Next 6	Mailer identifier	The valid range is 000000–999999, as assigned by the United States Postal Service for each customer.
Next 9	Sequence number	The valid range is 00000000–999999999, as assigned by mailer for tracking mail pieces.

The following table describes the digit sequence for the routing code in Four State barcodes:

Digit sequence for the routing code in Four State barcodes

Digits	Description	Limitations
None	No ZIP Code	--
5	5-digit ZIP Code	The valid range is 00000–99999.
9	9-digit ZIP Code	The valid range is 000000000–999999999.
11	11-digit ZIP Code	The valid range is 00000000000–99999999999.

## 5.1.6 General Purpose

General Purpose barcodes are useful when other barcode types available in Design Manager do not meet your needs. You can define a General Purpose barcode according to the specifications of a particular barcode reader.

General Purpose barcode example



## 5.1.7 Interleaved 2 of 5

Interleaved 2 of 5 barcodes encode pairs of digits, using five bars to encode one of the digits, and the spaces between the five bars to encode the other. It gets its name not only from the fact that the spaces are interleaved between the bars, but also because two out of each set of five bars are wide. This configuration compresses information into a small area. Exstream supports

Interleaved 2 of 5 barcodes because they are more adaptable than the older Standard 2 of 5 and Industrial 2 of 5 barcodes. Interleaved 2 of 5 barcodes are often used in warehouse and industrial applications.

#### Interleaved 2 of 5 barcode example



12345670

The character set used for 2 of 5 barcodes uses the numbers 0–9 and framing characters, with values ranging from 00–99. The start code uses the pattern narrow bar/narrow space/narrow bar/narrow space, and the stop code uses the pattern wide bar/narrow space/narrow bar.

Because Interleaved 2 of 5 barcodes can encode only an even number of digits, an odd number of digits requires that a zero be added as the first digit. You can also use a checksum as the last digit.

### 5.1.8 Japanese Postal

The Japanese Postal barcode is font-based, as is the USPS POSTNET barcode. Japanese Postal barcodes use 21 characters. Each symbol consists of three bars, one or two of which are extended above and below the middle point. The number of extensions on both sides is always the same. Letters are created using three shifts. The shifts change the meaning of the symbol that follows. Japanese Postal barcodes are usually 0.35 cm tall. You must adjust the size depending on your font and output.

The Japanese Postal barcode generated by Design Manager complies with the requirements set by the Japanese Postal system.

#### Japanese Postal barcode example



Design Manager automatically truncates or expands the first 20 digits of the Japanese Postal barcode to fit the specification size of 21 digits (7 digits for the postal code plus 13 for the street address plus 1 for the check digit). This ensures that the production environment does not produce invalid Japanese Postal barcodes.

The first seven digits can be only numbers (0–9). The next thirteen can be any of the following:

- Numbers (0–9)
- A dash (-)
- Letters (A–Z)

With Japanese Postal barcodes, you do not have to enter the control characters. Exstream automatically inserts them for you.

## 5.1.9 Modified Plessey

Modified Plessey, or MSI, is a continuous, non-self-checking symbology. While the length of a Modified Plessey barcode can vary, normally each individual application implements a fixed-length code. Modified Plessey barcodes are composed of a start character (a wide bar followed by a narrow space), the encoded data, a check digit(s), and a stop character (narrow bar, wide space, narrow bar). Modified Plessey barcodes are used primarily for inventory purposes.

### Modified Plessey barcode example



Each data character in the Modified Plessey symbology is made up of four black or white modules. The Modified Plessey symbology is designed to encode digits zero through 9, but can also encode other characters. Each decimal digit is represented by a Binary Coded Decimal (BCD) number of 4 bits, which represent the four modules of bars and spaces.

### Modified Plessey symbology

Digit	Binary coded decimal
0	0000
1	1000
2	0100
3	1100
4	0010
5	1010
6	0110
7	1110
8	0001
9	1001
Start	110

Modified Plessey symbology, continued

Digit	Binary coded decimal
Stop	<ul style="list-style-type: none"><li>• 1001</li><li>• 0101</li><li>• 1101</li><li>• 0011</li><li>• 1011</li><li>• 0111</li><li>• 1111</li></ul>

### 5.1.10 OMR and GBR OMR

This section discusses the following topics:

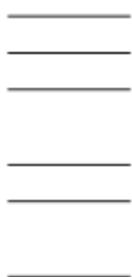
- “OMR” below
- “GBR OMR” on the next page

Optical Mark Reader (OMR) barcodes are often used to control an inserter. They are also used for collecting data (for example, from surveys and tests) and inventory control. Exstream provides the ability to use both OMR barcodes and GBR OMR barcodes. GBR OMR barcodes are a subtype of OMR.

## OMR

OMR barcodes can be up to eight lines that are all the same thickness. Lines can be either present or absent. A present line represents 1; an absent line represents zero.

### OMR barcode example



## GBR OMR

GBR OMR barcodes are a type of OMR barcode that uses a combination of broken and solid lines. A broken (or dashed) line represents zero and a solid line represents 1.

### GBR OMR barcode example



## 5.1.11 PDF 417

Portable Data File 417 (PDF 417) barcodes are two-dimensional, stacked barcodes that can encode over 1KB of data per label. They are mostly used when it is impractical to store item information in a database or the database is not accessible when and where the barcode is read. Because a PDF 417 barcode can store so much data, item data such as the content of a shipping manifest or equipment maintenance history can be carried on the item, without requiring access to a remote database.

### PDF 417 barcode example



Each PDF 417 barcode is a stack of vertically aligned rows. The barcode can contain between three and 90 rows. Each row includes between one and 30 symbol characters, excluding start, stop, and row indicator columns. There is a quiet zone on all four sides.

A PDF 417 barcode is created as a bitmap that is drawn to the page during engine run time. PDF 417 barcodes require a 2D scanner. They cannot be read using a linear barcode scanner.

## 5.1.12 POSTNET

POSTNET barcodes, also called USPS or 3 of 5 barcodes, conform to standards set by the United States Postal Service and consist of evenly spaced bars of two different heights. POSTNET barcodes must have tall bars in the start and stop positions. Each character is represented by five bars (two tall, and three short). The character set includes the digits 0–9.

**POSTNET barcode example**



POSTNET barcodes can contain a 5-digit ZIP Code, a 9-digit ZIP+4 code, or an 11-digit Delivery Point Code. Typically, these codes add a Modulo 10 checksum after the ZIP Code and before the ending frame bar to create totals in multiples of 10. Each number in a ZIP Code must consist of two long bars and three short bars.

### 5.1.13 QR Code

Quick Response (QR) Code barcodes are two-dimensional barcodes that have greater capacity for storage than standard barcodes. QR Code barcodes are also resistant to damage or poor resolution. QR Code barcodes are a good choice for tasks where a document might be scanned or copied, such as sending a fax. A common practice is also to embed web addresses to be read by mobile phones.

**QR barcode example**



QR Code barcodes can handle up to 7000 numeric characters, 4296 alphanumeric characters, and 2953 bytes of binary data. With QR Code, you can encode characters, numbers, text, and actual bytes of data. The size of the square barcode varies depending on your output device resolution and scanner capabilities.

QR Code barcodes require a 2D scanner. They cannot be read using a linear barcode scanner.

### 5.1.14 UPC

UPC-A barcodes encode 11 digits of numeric message data, along with a trailing check digit, for a total of 12 digits of barcode data. UPC-A barcodes are used to identify unique products for retail or tracking purposes.

**UPC barcode example**



The number system digit is typically printed just to the left of the barcode, the checksum digit just to the right of the barcode, and the manufacturer and product codes are printed just below the barcode.

In UPC-A barcodes, the number system code is a single digit from 0–9.

## 5.2 Creating Barcodes

In the design environment, you can create a barcode in three different ways, depending on your production needs:

- **Create a barcode object in Design Manager**—Use this method if you want to create a reusable barcode object to store in the Library. These objects can be placed on the design page where necessary.
- **Create a barcode from fonts in Designer**—Use this method if you want to create a single, unchanging barcode or a barcode controlled by a single variable.
- **Import scanned barcodes to Designer**—Use this method if you want to use a single, static barcode throughout your application.

### 5.2.1 Creating a Barcode Object in Design Manager

Design Manager barcodes are stored as objects under the **Barcodes** heading in the Library. These barcodes can be inserted where necessary into your designs.

Creating a barcode object in Design Manager lets you create a reusable barcode object that can support specific production settings. This method enables your barcode to change, depending on preset actions. If you want to embed a barcode, you must create it in Design Manager. After you create the barcode object in Design Manager, you can add it to the design page in Designer.

Barcodes created in Design Manager use data that can change in a variety of ways with the use of any combination of variables or processing events. You can define your barcode to do any of the following:

- Include information from data files.
- Control post-processing equipment and customer responses.
- Include information such as the customer number or sheet count.

To create a barcode object in Design Manager:

1. In Design Manager, in the Library, right-click the **Barcodes** heading and select **New Barcode**.
2. In the **Name** box, enter a name for the new barcode. In the **Description** box, enter a

description (optional).

3. Click **Next**.
4. From the list of barcodes, select the type of barcode you want to create.
5. Click **Finish**.

The barcode object opens in the Property Panel for you to define.

6. In the **Objects that can reference this barcode** list, you can limit the objects that can use the barcode you created.

Select from the following options:

- **Any type**—Any object can use the barcode.
- **Messages only**—Only messages can use the barcode.
- **Pages only**—Only pages can use the barcode.
- **Templates only**—Only templates can use the barcode.
- **Inserters only**—Only inserters can use the barcode.

**Note:** If you do not want to limit the object that can incorporate the barcode, select **Any type**. The option you select in the **Objects that can reference this barcode** list determines the options that are active in the **Content Type** list on the **Contents** tab.

For more information about choosing the type of content to include in the barcode, see “[Content Type Options for Barcodes](#)” on page 282.

7. Save the barcode.

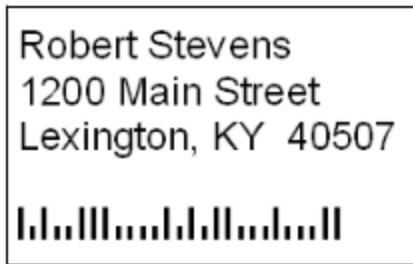
## 5.2.2 Creating a Barcode from Fonts in Designer

You can create a font-based barcode by applying a barcode font to a variable in your application. Creating a barcode from fonts in Designer lets you use a streamlined approach to basic barcode use. This method is recommended if you must use a single, unchanging barcode or a barcode controlled by a single variable.

For example, in an address block, you can insert the variables for customer name, address, city, state, and ZIP Code. On the next line, you can insert a formatted postal code variable and format it with the USPS barcode font. When you run the engine, Exstream creates a POSTNET barcode that changes with the ZIP Code for each customer.

For more information about using variables in your application, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

**Barcode created from font**



To create a barcode from fonts:

1. In Designer, place the user-created variable you want to use on the design page. As in the previous example, you can use a variable that provides postal code information.
2. Highlight the variable and apply the font you want to use. As in the previous example, you can apply the USPS barcode font to create a barcode for mailing in the United States.
3. Save the page.

### 5.2.3 Importing a Scanned Barcode to Use in Designer

If you want to use a single, static barcode throughout your application, you can import an image of a previously created barcode and insert the image into your application.

For more information about importing images, see *Importing External Content* in the Exstream Design and Production documentation.

You can import a pre-created barcode that you capture with a digital scanner into Designer as an image, but the information is static on each barcode. The only change you can make on customer data is with inclusion/exclusion rules. One option is to create rules that select a particular barcode from among multiple scanned barcode images.

## 5.3 Specifying How a Barcode is Generated

You can specify that Exstream generate a barcode in the following ways:

- **Generate a barcode with a specified font**—Use this option if you are including the barcode fonts in your print stream.
- **Generate a barcode by drawing the lines**—Use this option if you do not want to include the barcode fonts in your print stream, or if your barcode font does not provide the precision you need for the requirements of your barcode reader.

### 5.3.1 Generating a Barcode with a Specified Font

If you have the barcode font you need for the requirements of your barcode reader, you can add the font to Design Manager and use it to create barcodes. This method requires that you include the font in the print stream.

For more information about using fonts in Exstream, see *System Administration* in the Exstream Design and Production documentation.

To generate a barcode with a specified font:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. Click .  
The **Select Font** dialog box opens.
4. Select the font that you want to use.
5. Click **OK**.
6. Save the barcode object.

### 5.3.2 Generating a Barcode by Drawing the Lines

If you do not want to include the barcode fonts in your print stream, or if your barcode font does not provide the precision you need for the requirements of your barcode reader, you can specify that Exstream individually draw the lines of a barcode. This method also gives you more tight control over how the barcodes appear when printed. This can be especially helpful if you are using an inkjet printer. Ink run during printing can cause barcodes to scan improperly or not at all. Drawing the lines can create a more precise barcode that helps ensure proper scanning.

You can use this option for the following barcodes:

- 3 of 9
- Interleaved 2 of 5
- Code 128
- EAN 128
- Japanese Postal
- Modified Plessey
- UPC
- EAN-8

When a barcode is drawn, the map string character position is used to specify the type of bars drawn. If you choose to draw the lines of your barcode, make sure that you use the map string needed for your barcode. You must also set the size of bars and spaces based on dots per inch (dpi) settings.

For more information about the map string, see “[Customizing the Barcode Character Set](#)” on [page 273](#).

For drawn barcodes, the size of the bars are measured by the number of dots the bar or space occupies within the resolution settings of your output. For example, two dots in output with the resolution set at 300 dpi are larger than two dots in output with the resolution set at 900 dpi. You set the resolution of your output using both your output device and your application settings.

To generate a barcode by automatically drawing the lines:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. On the **Basic** tab, select the **Draw Lines** check box.
3. In the **Barcode sizes in dots** area, in the **Spaces** boxes, specify the size of the space between the bars.
4. In the **Bars** boxes, specify the size of the bars.
5. In the **Size** boxes, order the bars from narrow (starting in the **Size 1** box) to wide (in the subsequent boxes) as needed.
6. Save the barcode object.

For more information specifying a map string, see “[Specifying the Placement and Appearance of a Barcode Object](#)” below.

## 5.4 Specifying the Placement and Appearance of a Barcode Object

In Design Manager, you can specify exactly where you want to place your barcode on the page and how you want the barcode to look in the output.

For information about placing objects on a page, see *Designing Customer Communications* in the Exstream Design and Production documentation.

This section discusses the following topics:

- “[Specifying Where the Barcode is Placed on the Page](#)” on the next page
- “[Adjusting the Size of a Barcode](#)” on the next page
- “[Rotating a Barcode](#)” on [page 273](#)
- “[Customizing the Barcode Character Set](#)” on [page 273](#)

- “Specifying the Size of Bars and Spaces for Four State, Japanese Postal, and OMR Barcodes” on page 274
- “Adding a Human-Readable Identifier to a Barcode” on page 275
- “Using the BCOCA Standard (AFP Output Only)” on page 277
- “Specifying 2D Barcode Properties” on page 278

## 5.4.1 Specifying Where the Barcode is Placed on the Page

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. To help you manually position a barcode in Designer, you can enter a design sample in the **Design sample** box. Enter the same number of characters as the final barcode.

**Note:** Design samples are not supported in DBCS applications.

4. In the **Position** area, to specify the starting point for the position measurements, select one of the following options from the **Placement reference** drop-down list:
  - To measure barcode placement from the upper left corner of a portrait page, select **Upper left (portrait)**.
  - To measure barcode placement from the upper left corner of a landscape page, select **Upper left (landscape)**.
5. In the **Bottom** box, specify the distance from the top of the page to the bottom of the barcode. The default is 1 inch (2.54 cm).
6. In the **Left** box, specify the distance from the left of the page to the left of the barcode. The default is 1 inch (2.54 cm).
7. Save the barcode object.

## 5.4.2 Adjusting the Size of a Barcode

In Exstream, you can adjust the height of the tallest bars in the barcode. When you run the engine, Exstream automatically determines the width of your barcode based on the height you specify and the number of characters in your barcode.

To adjust the size of a barcode:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Position** area, in the **Height** box, specify the height of the tallest bars for this code. The default is 0.500 inch (1.27 cm).
4. Save the barcode object.

### 5.4.3 Rotating a Barcode

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. In the **Position** area, select an option from the **Orientation** drop-down list. The option you select determines in which direction the barcode is read.

Select from the following options:

- **Normal**—The barcode is positioned as designed.
  - **Face left**—The bottom of the barcode faces the left side of the page. The barcode is rotated 90 degrees.
  - **Face up**—The bottom of the barcode faces the top of the page. The barcode is rotated 180 degrees.
  - **Face right**—The bottom of the barcode faces the right side of the page. The barcode is rotated 270 degrees.
4. Save the barcode object.

### 5.4.4 Customizing the Barcode Character Set

If you are using a custom barcode font, you might need to alter the character set used to map the barcode characters.

**Important:** It is recommended that you do not alter the default map string unless you are using a customized barcode font.

To specify a character set other than the default:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. On the **Basic** tab, in the **Map string** box, enter the characters that you want to use to

define the barcode. If you are using start and stop characters, you must include them.

**Note:** The characters that you enter in the **Map string** box are mapped to each barcode value in the order that they are entered, so make sure that you list them in the proper order based on the barcode font that you are using.

3. If you want to view characters in hexadecimal format, select the **Hex** check box. The string in the **Map string** box changes to hexadecimal.
4. For 3 of 9 barcodes, you can specify a numerical base to customize the map string.
  - a. In the **Base** box, enter the numerical base required for your barcode. Common base values include binary (2), decimal (10), hexadecimal (16), and duotrigesimal (32).
  - b. In the **Map string** box, update the existing string to use the specified numerical base.
5. Save the barcode object.

## 5.4.5 Specifying the Size of Bars and Spaces for Four State, Japanese Postal, and OMR Barcodes

For barcodes that use bars (as opposed to those that use 2D properties), you must specify the size of the space between the bars and the thickness of the bars. You set the properties of some barcodes based on dpi settings and others are based on other measurements (such as inches).

You must adjust settings according to the requirements of your reading equipment.

To specify the size of bars and the space between the bars for Four State, Japanese Postal, and OMR barcodes:

1. In Design Manager, from the Library, drag a barcode object that has been assigned the type Four State, Japanese Postal, and OMR barcode to the Property Panel.
2. Click the **Basic** tab.

3. Depending on the type of barcode you are using, complete the following tasks:

To	Do this
Specify the size of bars and spaces for the following barcodes: <ul style="list-style-type: none"><li>• Four State</li><li>• Japanese Postal</li></ul>	In the <b>Barcode sizes</b> area: <ol style="list-style-type: none"><li>a. To specify the size of the space between the bars, in the <b>Space</b> box, enter or select the size of the space you want.</li><li>b. To specify the thickness of the bars, in the <b>Thickness</b> box, enter or select the thickness you want.</li></ol>
Specify the size of bars and spaces for the following barcodes: <ul style="list-style-type: none"><li>• OMR</li><li>• GBR OMR</li></ul>	In the <b>OMR bar sizes</b> area: <ol style="list-style-type: none"><li>a. To specify the size of the space between the bars, in the <b>Space</b> box, enter or select the size of the space you want.</li><li>b. To specify the thickness of the bars, in the <b>Thickness</b> box, enter or select the thickness you want.</li></ol>

4. Save the barcode object.

For information about specifying the size of bars and spaces for other barcodes, see “[Generating a Barcode by Drawing the Lines](#)” on page 270.

## 5.4.6 Adding a Human-Readable Identifier to a Barcode

You can add a human-readable identifier (HRI) to a barcode so a production operator can read the barcode. This is helpful if an inserter stops when a barcode is incorrect or out of sequence. The human operator can read the HRI that accompanies the barcode to help determine the problem.

In the following example of a UPC barcode, the numbers below and next to the bars are the HRI.

### HRI example



To add an HRI:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. On the **Basic** tab, in the **Human readable (HRI)** area, in the **Placement** list, specify where you want to place the HRI. Select from the following options:

To	Do this
Place the HRI above the barcode	Select <b>Above</b> .
Place the HRI below the barcode	Select <b>Below</b> .
Specify the exact placement of the HRI on the output page	<ol style="list-style-type: none"><li>Select <b>Positioned</b>.</li><li>Enter the exact X/Y coordinates for the HRI:<ul style="list-style-type: none"><li>• In the <b>Bottom</b> box, enter the Y coordinate.</li><li>• In the <b>Left</b> box, enter the X coordinate.</li></ul></li><li>To offset the HRI relative to the barcode, select the <b>Relative</b> check box. Otherwise, the HRI is offset relative to the page margins.</li></ol>
Disable HRI settings	Select <b>None</b> .

3. In the **Orientation** list, specify the direction the HRI characters face:
  - **Normal**—The HRI is positioned so that it is read from left to right.
  - **Face left**—The HRI is positioned so that it is read from bottom to top.
  - **Face right**—The HRI is positioned so that it is read from top to bottom.
  - **Face up**—The HRI is positioned so that it is read from right to left.
4. Specify the style of the HRI:

To	Do this
Specify the font	<ol style="list-style-type: none"><li>In the <b>HRI font</b> box, click .</li><li>The <b>Select Font</b> dialog box opens.</li><li>From the <b>Font</b> list, select a font.</li><li>Click <b>OK</b>.</li></ol>
Specify the font size	Enter a font size in the <b>Size</b> box.
Make the font bold	Select the <b>Bold</b> check box.
Make the font italic	Select the <b>Italic</b> check box.

**Tip:** If you want to add special characters to the HRI, such as a hyphen, you can place a text box on the design page just below the barcode.

5. Save the barcode object.

## 5.4.7 Using the BCOCA Standard (AFP Output Only)

The Bar Code Object Content Architecture (BCOCA) standard lets you print barcodes without special barcode fonts or images. The BCOCA standard embeds data control commands in the print stream to generate a barcode for an AFP output device in a specified type, size, and orientation.

BCOCA-generated barcodes are composed only for AFP output. With any other output driver, the barcode is created as a font barcode. When you configure the properties for the barcode, keep in mind that the print stream determines the final dimensions of BCOCA barcodes at run time.

BCOCA settings are not available for OMR-based barcodes.

### Creating a Barcodes with BCOCA Settings

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. To enable the use of BCOCA, select the **Generate AFP BCOCA barcode** check box.

For DataMatrix 2D, Four State, PDF 417, POSTNET, and QR Code barcodes, this is the only BCOCA option available. When you select the **Generate AFP BCOCA barcode** check box, you must select a font so that Exstream has some spacing metrics to reference during composition. The actual AFP structured fields generated are for a BCOCA barcode object, not for barcodes using a font.

4. In the **BCOCA Width** box, enter the approximate width needed for the barcode in your output. The actual printed width can vary from your width measurement due to run-time processing that ensures the barcode stays in the barcode specifications for spacing and bar thickness.
5. If you want to use a check digit, select the **Use BCOCA Check Digit** check box. When you select this check box, the modifier byte (byte of 23) of the Barcode Data Descriptor record is set in the engine. As a result, the barcode is generated with a check digit. A BCOCA check digit is automatically included for POSTNET, UPC, Code 128, EAN-8, and EAN 128 barcodes.

**Tip:** You can also enable the use of a check digit by including a checksum in your data.

6. Save the barcode object.

## Adding a Human Readable Identifier to a BCOCA Barcode

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. Select the **Use BCOCA HRI** check box.
4. In the **Human readable (HRI)** area, in the **Placement** list, specify where you want to place the HRI characters in reference to the barcode.

Select from the following options:

  - **Above**—The HRI is positioned above the barcode.
  - **Below**—The HRI is positioned below the barcode.
  - **Default**—The characters are placed according to the default settings of the output device.
5. If you do not want to include an asterisk at the beginning and end of the character string, select the **Do not include asterisk** check box.
6. Save the barcode object.

For more information about adding an HRI, see “[Adding a Human-Readable Identifier to a Barcode](#)” on page 275.

### 5.4.8 Specifying 2D Barcode Properties

Two-dimensional barcodes include DataMatrix 2D, QR Code, and PDF 417 barcodes. The 2D properties vary for each barcode type.

#### Specifying the 2D Properties for DataMatrix 2D Barcodes

1. In Design Manager, from the Library, drag a DataMatrix 2D barcode to the Property Panel.
2. On the **Basic** tab, in the **Pixel block size** box, enter a value to control the size of each black or white square in the barcode. Most barcode readers require a specific block size.
3. To specify the size of the barcode (in pixel blocks), do one of the following:

To	Do this
Let the engine specify the most efficient size (the smallest number of pixel blocks) when creating the barcode. The size of the barcode might vary.	In the <b>Dimension HxW</b> list, select <b>Automatic</b> .

To	Do this
Maintain a standard size for the barcodes across your output	<ul style="list-style-type: none"> <li>In the <b>Dimension HxW</b> list, select one of the standard options.</li> <li>Click in the <b>Dimension HxW</b> box and enter the size you want.</li> </ul>

If the size you select or enter is not big enough for the encoded data, you receive an error. Keep in mind that the actual size of the barcode in the output depends on your output driver.

4. If you want to use error correction, in the **Error correction level** list, select an industry standard error correction level that corresponds to the specifications of the equipment that will be used to read the barcode.
  - If you select **None**, error correction is not used and any damage to the barcode can render it unreadable.
  - If you select **ecc 200** (the default), which has the most error correction, less than half of the barcode is required to remain readable. You must select this option if you selected the **Generate AFP BCOCA barcode** check box.
5. In the **Data type** list, select an option to modify the size of the barcode that is necessary to hold a given amount of data. Choose the smallest character set possible. For example, if a barcode holds only numeric data, then select **Numeric** rather than **Full 128 ASCII set**.

**Note:** Keep in mind that if you select or accept the default selection of **ecc 200** in the **Error correction level** list, the **Data type** list becomes inactive and the engine automatically determines the barcode's **Data type** setting based on the barcode content.

6. Save the barcode object.

## Specifying the 2D Properties for PDF 417 Barcodes

1. In Design Manager, from the Library, drag a PDF 417 barcode object to the Property Panel.
2. On the **Basic** tab, in the **Module width** box, specify width of the narrowest bar or space in a barcode symbol.
3. In the **Module height** box, specify the height of each row in the stack symbol.
4. In the **Error correction level** list, set the error correction level. Since barcodes can become damaged in handling, PDF 417 barcodes have error-correcting symbology. The level that you select determines the number of EC codewords that are added to the printed symbol.

5. In the **Compaction type** list, specify the compaction needed for the barcode:

Option	What can be encoded	Maximum compaction
<b>Text</b>	All printable ASCII characters (Values 32–126 and select control characters)	Two characters per codeword
<b>Byte</b>	All ASCII characters (Values 0–127). This option also provides international character set support.	1.2 bytes per codeword
<b>Numeric</b>	Long digit strings	2.9 digits per codeword

6. In the **Rows** box, specify the number of rows used in the barcode. If you want Exstream to automatically specify the number of rows, enter 0.
7. In the **Columns** list, specify the number of columns used in the barcode. If you want Exstream to automatically specify the number of columns, select 0. PDF 417 barcodes generally have an aspect ratio of 3:1 (width:height). When you enable the automatic selection of rows and columns, Exstream uses this default aspect ratio to calculate the number of rows and columns.
8. Save the barcode object.

## Specifying the 2D Properties for QR Code Barcodes

1. In Design Manager, from the Library, drag a QR Code barcode object to the Property Panel.
2. On the **Basic** tab, in the **Module width** box, specify the size for each individual QR Code module (or block).
3. To control the number of modules that compose a QR Code barcode, do one of the following:

To	Do this
Let Exstream encode the data in the smallest version possible. This is the recommended option for the best performance.	In the <b>Symbol Version (size)</b> list, select <b>Auto</b> .
Specify the number of modules	<ul style="list-style-type: none"><li>• In the <b>Symbol Version (size)</b> list, select one of the standard options.</li><li>• Click in the <b>Symbol Version (size)</b> box and enter the size you want.</li></ul>

4. In the **Error correction level** list, select an error correction value to control the amount of error recovery data in the barcode. Select an option that corresponds to the specifications of the equipment that will be used to read the barcode.
5. In the **Mask Pattern** list, select an option to specify the mask pattern settings you want to

use. If you want Exstream to determine the best choice for the mask pattern settings, select **Automatic**. Otherwise, select a number from **0** to **7**.

6. Save the barcode object.

## 5.5 Specifying the Content of a Barcode Object

You can assign variables, processing events, or other content to the various segments of a barcode object using the **Contents** tab. For example, if you are using inserters, you can select which bin is controlled by the barcode. When the barcode is read, the insert driven by the barcode is pulled from the correct bin. You can also use variables to control processes, such as starting a new envelope.

Your barcode reader might need specific barcode input to use your barcodes correctly. Before you configure your barcode content, make sure that you know the requirements of your barcode reader.

To define the content of a barcode object:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Contents** tab.

Each row of the **Contents** tab represents a successive segment of the barcode. The length of each segment is determined by the value in the **Digits** column. The order of the digits from top to bottom determines the order of the barcode segments from left to right.

3. To assign content to a segment of your barcode, complete the following steps:
  - a. In the **Digits** column, enter the length for that segment of the barcode.
  - b. Use the **Content type** list and **Value** boxes to specify the data source used to populate the barcode segment indicated by the number of digits in the **Digits** box.

To	Do this
Use a variable data source	<ol style="list-style-type: none"><li>i. In the <b>Content type</b> list, select <b>Variable</b>.</li><li>ii. In the <b>Value</b> box, click  to open the <b>Select Variable</b> dialog box, and select the variable that provides the value.</li></ol>

To	Do this
Use a static value	<p>i. In the <b>Content type</b> list, select <b>Text</b>.</p> <p>ii. In the <b>Value</b> box, enter the static value that you want to use.</p> <p>For example, if you have been provided with a number that is required in the barcode, enter that number in the <b>Value</b> box.</p>
Use a different data source	<p>In the <b>Content type</b> list, select one of the other options. These options vary depending on which objects you have allowed to reference the barcode.</p> <p>For more information about setting up which objects can reference the barcode, see "<a href="#">Creating a Barcode Object in Design Manager</a>" on page 267.</p>

**Note:**

- Make sure that the number of digits that you entered in the **Digits** box is sufficient to contain the information that is specified in the **Content type** list. If you enter too few digits to contain the information, digits are truncated from the left.
- The **Content type** options vary based on where your content is placed in your output. For more information about barcode positioning, see "[Specifying Where the Barcode is Placed on the Page](#)" on page 272.

For more information about choosing the type of content to include in the barcode, see "[Content Type Options for Barcodes](#)" below.

4. If you are creating an OMR-based barcode, you can select the **Zero** check box if the corresponding digit position supports a value of zero. With OMR barcodes, a zero value in the barcode results in no line at the respective position in the barcode. If the **Zero** check box is cleared and the engine encounters a zero value, you receive an error message.
5. If you are creating a 3 of 9 barcode, you can specify that Exstream convert integer values to different base number formats by selecting the **Map** check box for multiple digits. Exstream converts the value of the content that you specified based on the base number that you entered in the **Map string** box on the **Basic** tab.
6. Save the barcode object.

## 5.5.1 Content Type Options for Barcodes

You add content to barcodes on the **Content** tab of the barcode object properties in Design Manager. This section provides a reference for the options available in the **Content type** list. These options are dependent on the type of barcode you are creating and which objects you have allowed to reference the barcode.

For more information about setting up which objects can reference the barcode, see “[Creating a Barcode Object in Design Manager](#)” on page 267.

For more information about adding content to a barcode, see “[Specifying the Content of a Barcode Object](#)” on page 281.

The following table demonstrates the availability of options in the **Content type** list, based on the option you select in the **Objects that can reference this barcode** list on the **Basic** tab of the barcode object properties:

#### Content type availability

Options in the <b>Content type</b> list	Any Type	Messages Only	Pages Only	Templates Only	Insters Only
General content types	X	X	X	X	X
Sheet, document, page, and queue content types	X		X	X	X
Bin content types	X		X	X	X
Message tracking ID	X	X		X	X
Checksum and checkchar types	X	X	X	X	X
GBR 8 bit OMR	X				
GBR 16 bit OMR	X			X	X

## General Content Types

General content types include the following:

#### General content types for barcodes

Option	Available with these barcode types	Description
<b>Empty</b>	All	The digit information is generated using empty spaces. Select this option if your barcode is not used.
<b>Variable</b>	All	The digit information is generated using the variable defined in the <b>Variable</b> box.
<b>Benchmark (always 1)</b>	<ul style="list-style-type: none"><li>OMR and GBR OMR</li><li>PDF 417</li></ul>	The digit information is generated at a value of 1.

## Sheet, Document, Page, and Output Content Types

Sheet, document, page, and output queue content types let you base barcode content on where the barcode is positioned at run time. These content types are available for all barcodes except for the Four State barcode. Select one of the options in the following table:

Sheet, document, page, and queue content types

Option	Description
<b>Total Sheets in Doc</b>	The digit value is equal to the total sheets in your document. The value is determined by the system variable 'SYS_SheetTotalInDocument'.
<b>Total Pages in Doc</b>	The digit value is equal to the total pages in your document. The value is determined by the system variable 'SYS_PageTotalInDocument'.
<b>Sheets in Document</b>	The digit value is equal to the number of the sheet on which the barcode is placed. The value is determined by the system variable 'SYS_SheetInDoc'.
<b>Pages in Document</b>	The digit value used is equal to the current number of counted pages within the document being written. The value is determined by the system variable 'SYS_PageInDoc'.
<b>Page in Queue</b>	The digit value is equal to the current page number in the current queue.
<b>Page in Stream</b>	The digit value is equal to the current page number within the current break of the current queue. The value is determined by the system variable 'SYS_PageInBreak'.
<b>First Sheet</b>	The value for the first sheet is 1. All other pages have a value of zero.
<b>Last Sheet</b>	The value for the last sheet is 1. All other pages have a value of zero.
<b>Not First Sheet</b>	The value of all sheets except the first sheet is 1. The first sheet has a value of zero.
<b>Not Last Sheet</b>	The value of all sheets except the last sheet is 1. The last sheet has a value of zero.
<b>Sheet in Queue</b>	The digit value is equal to the current sheet number in the current output queue. The value is determined by the system variable 'SYS_SheetInQueue'.
<b>Sheet in Break</b>	The digit value is equal to the current sheet number in the current break of the current output queue. The value is determined by the system variable 'SYS_SheetInBreak'.
<b>Document in Queue</b>	The digit value is equal to the current customer number being written into the current output queue. The value is determined by the system variable 'SYS_DocumentInQueue'.
<b>Document in Break</b>	The digit value is equal to the current customer number within the current break of the current output queue. The value is determined by the system variable 'SYS_DocumentInBreak'.

## Bin Content Types

Bin content types let you base barcode content on a specified inserter bin at run time. Select the bin number you want to use. Bin content types are available for all barcodes except for the Four

State barcode.

## Checksum and Checkchar Content Types

Checksum or checkchar content types let you include a checksum or checkchar character specific to your barcode type. Select one of the following options:

Checksum and checkchar content types

Option	Available with these barcode types	Description
<b>Message Tracking ID</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>3 of 9</li><li>Interleaved 2 of 5</li><li>DataMatrix 2D</li><li>Japanese Postal</li><li>Code 128</li><li>Modified Plessey</li><li>UPC</li><li>PDF 417</li><li>QR Code</li></ul>	Adds an ID to track messages
<b>Checksum: Base</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>3 of 9</li><li>POSTNET</li></ul>	Adds a character used to calculate the base
<b>Checksum: Base Sum</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>3 of 9</li><li>POSTNET</li></ul>	Adds a character used to calculate the base sum
<b>Checksum: 2of5</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>Interleaved 2 of 5</li></ul>	Adds checksum digits used with Interleaved 2 of 5 barcodes
<b>Checkchar: Code 128</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>Code 128</li></ul>	Adds the checksum digit used with Code 128 barcodes. Code 128 barcodes require a checksum character that is printed after your data and before the stop character in your barcode. The checksum character is expected to be a single character. The checkchar is created using information from the encoded data and the current character set. The default values of the characters used in the checksum are based on Code 128 barcode character tables (according to character set A, B, or C, as determined by the start character). If the complete character set is mapped in the map string, the values are based on the mapped locations instead.

#### Checksum and checkchar content types, continued

Option	Available with these barcode types	Description
<b>Checksum: Jpostal</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>Japanese Postal</li></ul>	Adds the checksum digit used with Japanese Postal barcodes. Japanese Postal barcodes require a checksum character as the 21st character in the string. The default values of the checksum are based on the default Japanese Postal barcode character table. If the complete character set is mapped in the map string, the values are based on the mapped locations instead.
<b>Checksum: MSI Mod 10</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>Modified Plessey</li></ul>	General Purpose Modified Plessey Adds a checksum digit calculated using the MSI Mod 10 method
<b>Checksum: MSI Mod 11</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>Modified Plessey</li></ul>	Adds a checksum digit calculated using the MSI Mod 11 method
<b>Checksum: UPC</b>	<ul style="list-style-type: none"><li>General Purpose</li><li>UPC</li></ul>	Adds checksum digits used with UPC barcodes
<b>Checksum: OMR odd</b>	<ul style="list-style-type: none"><li>OMR and GBR OMR</li></ul>	Adds an extra line if the number of lines is even
<b>Checksum: OMR even</b>	<ul style="list-style-type: none"><li>OMR and GBR OMR</li></ul>	Adds an extra line if the number of lines is odd

## 5.6 Previewing and Printing a Barcode

As you change the properties of a barcode, the **Preview** area on the **Basic** tab of the Property Panel is updated to reflect changes to the barcode properties. You can use this area to see how changes you make to the barcode will affect the physical appearance of the barcode.

After you have defined the basic properties of a barcode, you can print a copy of the barcode preview to your local printer. The ability to print a barcode from Design Manager allows you to create a sample of the barcode that you can use to verify that the barcode properties meet your requirements without requiring you to package the application to see the barcode. Depending on the DPI settings of your local printer, the appearance of the printed barcode might vary slightly from the appearance of the barcode in the **Preview** area.

To print a barcode:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Basic** tab.
3. Below the **Preview** area, click **Print**.

The **Print** dialog box opens.

4. Make the necessary selections and click **OK**.

A sample of the barcode is printed according to the printer options you selected.

## 5.7 Using Barcodes in Your Design

In Exstream, barcodes are most often used with inserters to control inclusion of pre-printed inserts loaded on an inserter device and to facilitate the tracking features. In addition to incorporating barcodes in your design, you can also add barcodes to banner pages, or for bundled output, you can add barcodes to cover pages or trailer pages to control various production equipment (for example, inserter devices).

For more information about banner pages, see “[Using Banner Pages to Mark Breaks in Output](#)” on page 295.

For more information about cover pages and trailer pages, see “[Using Bundling \(Householding\) to Group Customer Information](#)” on page 343.

You can add barcodes to your design page for other purposes, such as for authorizing entrance to an event on ticket stubs. You can also use barcodes in flowing tables to create a pick list to track inventory.

This section discusses the following topics:

- “[Adding a Barcode Object to a Page in Designer](#)” below
- “[Creating an Inventory or Pick List with Barcodes in a Table](#)” on the next page

### 5.7.1 Adding a Barcode Object to a Page in Designer

After you set up a barcode object in Design Manager, you can add the barcode to a page in Designer.

To add a barcode to a page in Designer:

1. From the Menu bar, select **Insert > Drawing Object > Barcode**.
2. In the **Folder** box, select the folder that contains the barcode object.
3. In the **Barcode** list, select the barcode object you need.

The **Barcode** list is populated with the barcode objects you have created in Design Manager.

4. To enable a human-readable identifier (HRI), select the **Include text (HRI)** check box.
5. Format the tracking ID in one of the following ways:
  - If you want Exstream to provide a tracking ID for this barcode, select the **Use automatic ID value** check box.
  - If you want to specify a tracking ID for this barcode, clear the **Use automatic ID value** check box and enter the ID in the **Barcode value** box.
6. Click **OK**.

The barcode appears on the page.

For information about positioning a barcode object on the design page, see *Designing Customer Communications* in the Exstream Design and Production documentation.

## 5.7.2 Creating an Inventory or Pick List with Barcodes in a Table

You can create a table with repeating rows that contains barcodes, each with different values. This method is useful for creating a long list of barcodes, such as for a pick list for inventory control. To do this without creating a separate barcode for each value, you can use array variables.

For more information about tables and embedding objects, see *Designing Customer Communications* in the Exstream Design and Production documentation.

To create an inventory or pick list with barcodes in a table:

1. In Design Manager, from the Library, drag a barcode object to the Property Panel.
2. Click the **Contents** tab and make sure you select an array variable to control the dynamic barcode content.
3. In Designer, open the design page with the table in which you want to embed the barcode.
4. Click inside an automated row on the table.
5. From the Menu bar, click **Insert > Drawing Object > Barcode**.
6. In the **Barcode** list, select the barcode object you need.

The **Barcode** list is populated with the barcode objects you have created in Design Manager.

7. To enable a human-readable identifier (HRI), select the **Include text (HRI)** check box.
8. Format the tracking ID in one of the following ways:

- If you want Exstream to provide a tracking ID for this barcode, select the **Use automatic ID value** check box.
  - If you want to specify a tracking ID for this barcode, clear the **Use automatic ID value** check box and enter the ID in the **Barcode value** box.
9. Click **OK**.
10. The barcode appears on the page.  
The barcode data will be updated as the table is generated.
11. If you want to generate data in an adjacent column that matches the value of the barcode, then add the same array variable you used to generate barcode data.  
The content of the array variable prints next to each barcode.

For more information about array variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

# Chapter 6: Setting Up for High-Volume Production

Exstream offers solutions for high-volume production that let you manage the creation of output to multiple print and electronic channels to maximize the efficiency of print/mail centers.

This chapter discusses the following topics:

- “[Using Inserter Objects to Add Messages and Inserts During Print Production](#)” below
- “[Using Banner Pages to Mark Breaks in Output](#)” on page 295
- “[Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects](#)” on page 300
- “[Controlling and Setting Up High Volume Production Options with Output Queues](#)” on page 315
- “[Adding Non-Printing Data or Production Device Controls with Search Keys](#)” on page 326

## 6.1 Using Inserter Objects to Add Messages and Inserts During Print Production

You can set up inserter objects so that the contents of inserter devices are reflected in Exstream. Using inserter objects with inserter devices lets you supplement your printed output during the production run with inserts, such as pre-printed advertisements or return envelopes.

For example, suppose you print billing statements and they include both a return envelope and two pre-printed insert ads. You can set up an inserter object in the Library for each insert and inserter device configuration. Suppose you also print welcome letters that include a brochure for a different production run, using the same inserter device. You can create another inserter object for this configuration, even though both inserter objects refer to the same device. If you do not create separate inserter objects, you must update the inserter object when the contents of an inserter bin change.

After you have created an inserter object, you can attach a barcode to the inserter object to control when inserts are included.

To add insert information to insert objects, you can use the following two message types:

- **Insert messages**—Insert messages reference a pre-printed message (for example, postcards, coupons, or teaser notes) in an inserter bin. To see a thumbnail of a pre-printed message in the Property Panel for a visual reference, you can import a scanned copy of the insert into Designer. If insert messages are included in output, they are considered when calculating package weight totals for postage.

- **Graphic/insert messages**—Graphic/insert messages are a backup option you can use to ensure that an insert message is included, even if a pre-printed message is not available. If the pre-printed message is unavailable, Exstream automatically composes the message you have either designed or imported as a scanned image in Designer. The message is placed on an available page during packaging, but the weight is not added to the total calculated weight for the output.

For more information about creating insert and graphic/insert messages, see *Managing Marketing Messages* in the Exstream Design and Production documentation.

This section discusses the following topics:

- “Configuring an Inserter Object” below
- “Including Inserts Only if the Language Matches the Customer” on the next page
- “Using a Barcode to Control an Inserter Object” on page 293
- “Specifying a Weight Selection Method for Carrier Envelopes” on page 293
- “Specifying the Way an Output Queue Handles an Inserter Object” on page 294

## 6.1.1 Configuring an Inserter Object

Before you can set up an inserter object, you must have created an insert message or a graphic/insert message.

For more information about creating messages, see *Managing Marketing Messages* in the Exstream Design and Production documentation.

To set up and configure an inserter object:

1. In Design Manager, in the Library, right-click the **Inverters** heading and select **New Inserter**.

The **Name** dialog box opens.

2. In the **Name** box, enter a name.

3. In the **Description** box, enter a description (optional).

4. Click **Finish**.

The inserter object opens in the Property Panel.

5. In the **Model** box, enter an identifying text string for each inserter device (optional). This option is especially useful if you must differentiate between two different inserter devices at the same location.

6. From the **Bins** drop-down list, select the number of inserter bins on your printer. You can define up to 32 inserter bins.

The bins appear on the right side of the Property Panel.

7. For each bin, select either **None** or the name of the message. Each drop-down list shows all messages that are either insert messages or graphic/insert messages. If available, a thumbnail of the message you select appears to the left of the drop-down list.
8. From the Menu bar, select **Edit > Save**.

## 6.1.2 Including Inserts Only if the Language Matches the Customer

You can set up an inserter object so that inserts are sent only if the language of the insert matches the language of the customer. For example, suppose you want to send a specific offer to customers in France. You can set the insert so that it is sent only to customers who are defined as speaking French.

Before you can assign a language to an insert message, you must have set up a language in the **System Settings**.

For more information about setting up languages, see *System Administration* in the Exstream Design and Production documentation.

To specify a language for an inserter object:

1. In Design Manager, from the Library, drag an inserter object to the Property Panel.
2. Specify the language settings:

To	Do this
Specify a language for an insert	<ol style="list-style-type: none"><li>a. Select the <b>Send inserts only if language matches customer</b> check box.</li><li>b. In the adjacent box, specify the language for each bin. The drop-down list is populated with the language layers available in the database. Use a different bin for each language setting of an insert. If there is no specific language for the message, leave the default setting as <b>Generic</b>.</li></ol> <p>For information about setting up a language, see <i>System Administration</i> in the Exstream Design and Production documentation.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"><p><b>Note:</b> Generic or default languages are treated as specific languages. Any customer with a defined language (such as English or French) does not receive a default language message.</p></div>
Disable language settings for an inserter object	From the <b>Language matching</b> drop-down list, select <b>Ignore language</b> .

3. From the Menu bar, select **Edit > Save**.

### 6.1.3 Using a Barcode to Control an Inserter Object

Exstream uses barcodes to control inserter object and inserter device configurations. The barcode that you add controls bin selection so that only the intended inserts go in a customer's envelope. Other information can be included in the barcode, as well, such as a variable that is set up to display the customer number, count the sheets in the document, or track other processing equipment information. You can assign up to 10 barcodes to an inserter object.

For more information about barcodes, see ["Using Barcodes in Exstream" on page 256](#).

To use a barcode to control an inserter object:

1. In Design Manager, from the Library, drag an inserter object to the Property Panel.
2. In the **Barcodes** area, from the **Name** box, select the barcode you want to use with this inserter object configuration.
3. From the **Placement** drop-down list, specify on which side of a sheet or in which multiple-up frame you want to place the barcode. You can set other placement settings, such as the direction of the barcode, in the properties of the barcode.
4. From the Menu bar, select **Edit > Save**.

### 6.1.4 Specifying a Weight Selection Method for Carrier Envelopes

You can set up how to specify the weight of carrier envelopes you use for production to help you control postage costs. If you use a variable to specify the weight selection method, you must have created a floating or integer variable.

For more information about setting up variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

To specify a weight selection method:

1. In Design Manager, from the Library, drag an inserter object to the Property Panel.
2. In the **Carrier Envelope** area, specify the weight of the envelope:

To	Do this
Use a static weight value	<ol style="list-style-type: none"><li>a. From the <b>Weight selection method</b> drop-down list, select <b>Static weight value</b>.</li><li>b. In the <b>Weight</b> box, enter the weight of the envelope. To determine the unit of measurement, use either the <b>oz</b> (ounces) or <b>g</b> (grams) abbreviation.</li></ol>

To	Do this
Use a variable weight value	<p>a. From the <b>Weight selection method</b> drop-down list, select <b>Variable weight value (ounces)</b> or <b>Variable weight value (grams)</b>.</p> <p>b. In the <b>Weight</b> box, click  to select a floating or integer variable to control the weight selection of the insert.</p>

- From the Menu bar, select **Edit > Save**.

## 6.1.5 Specifying the Way an Output Queue Handles an Inserter Object

To use an inserter object to control your inserter device, you must add the inserter object to an output queue. This task assumes that an output queue object has already been created.

For more information about output queues, see “[Controlling and Setting Up High Volume Production Options with Output Queues](#)” on page 315.

If you add multiple inserter objects to an output queue, you can identify a particular inserter object at run time with the SETQUEUEINSERTER engine switch. If you do not use the SETQUEUEINSERTER engine switch, the engine uses the first inserter object in the inserter object list in the output queue properties as a default.

For more information about engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

To add an inserter object to an output queue:

- In Design Manager, from the Library, drag an output queue object to the Property Panel.
- Click the **Inserter** tab.
- To add or remove an inserter object, do the following:

To	Do this
Add an inserter object	<p>a. Click  , and from the <b>Select an item to add to the list</b> dialog box, select an inserter object.</p> <p>b. Click <b>OK</b>.</p> <p>The inserter object name appears in the inserter object list.</p>
Remove an inserter object	In the inserter object list, select the inserter object you want to remove, and click  .

4. In the inserter object list, specify the order in which you want to process the inserter objects in the output:

To	Do this
Move an inserter object up in the inserter objects list	Select the inserter object you want to move and click  .
Move an inserter object down in the inserter objects list	Select the inserter you want to move and click  .

5. If you are using multiple queues and the output queue is not the first queue listed in an application, you can do the following:

To	Do this
Prevent the engine from composing blank pages on the backs of the printed sheets, if the first output queue in the application is set up to be duplex and subsequent output queues are not	Select the <b>Remove blank pages if primary is duplex</b> check box.
Duplicate the inserter object settings of the first output queue	Select the <b>Use same inserter configuration as primary</b> check box.  <b>Note:</b> If the <b>Use same inserter configuration as primary</b> check box is selected on the first output queue in the application, your output might not select the correct inserter objects. If you choose to change the order of your output queues, verify that the correct inserters are defined in the first output queue.

6. From the Menu bar, select **Edit > Save**.

## 6.2 Using Banner Pages to Mark Breaks in Output

With banner pages, you can mark breaks in the output and include current or summary information about the production run. These pages help print floor operators or manual insertion personnel handle a high volume of output by visually marking points in the output for separate mailings or by separating documents for different finishing operations (for example, if you are printing information for multiple projects in the same production run).

Before you can create a banner page, you must have designed a page with the information you want to appear on the banner page. To aid in splitting and sorting the finished output (for example, by vendor name or number of pages in the job), you add variable information to a banner page when you design it. When you add variables, keep in mind the engine adds banner pages during queue processing but before customer information is processed. So that banner

pages are easily recognized as a stopping point or as a change in processing, your banner page should be designed to be significantly different from the other pages in the output.

For more information about creating a page object, see *Designing Customer Communications* in the Exstream Design and Production documentation.

For more information about engine timing, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

Since banner pages are not a part of customer output, Exstream applies the default language layer. Banner pages are also not included on multiple-up sheets. If you include banner pages with multiple-up output, they are printed as stand-alone pages, using the dimensions of the paper type associated with the banner page.

This section discusses the following topics:

- “[Creating a Banner Page](#)” below
- “[Specifying When to Add a Banner Page in Output](#)” on the next page
- “[Specifying Where to Place a Banner Page in Output](#)” on the next page
- “[Adding an Envelope Banner Page to Split Customer Documents into Separate Envelopes](#)” on page 298
- “[Specifying the Way an Output Queue Handles Banner Pages](#)” on page 299

## 6.2.1 Creating a Banner Page

Before you can create a banner page object, you must have created a page.

To create a banner page object:

1. In Design Manager, in the Library, right-click the **Banner Pages** heading and select **New Banner**.

The **New Banner Page** dialog box opens.

2. In the **Name** box, enter a name.
3. In the **Description** box, enter a description (optional).

4. In the **Page** box, click .

The **Select Page** dialog box opens.

5. Select the page object you want to use as a banner page and click **OK**.

The page name appears in the **New Banner Page** text box.

6. Click **OK**.

The new banner page opens in the Property Panel.

For information about creating an envelope banner page, see “[Adding an Envelope Banner Page to Split Customer Documents into Separate Envelopes](#)” on the next page.

## 6.2.2 Specifying When to Add a Banner Page in Output

You can add banner pages either before or after a specified number of pages, documents, or customers, or before or after a combination of these three options.

To specify when to add a banner page in output:

1. In Design Manager, from the Library, drag a banner page object to the Property Panel.
2. In the **When to add banner page** area, set up the following options, as applicable:

To	Do this
Add the banner page before a specified number of pages	In the <b>Pages</b> box, enter the number of pages before which you want to add the banner page.
Add the banner page before a specified number of documents	In the <b>Documents</b> box, enter the number of documents before which you want to add the banner page.
Add the banner page before a specified number of customers	In the <b>Customers</b> box, enter the number of customers before which you want to add the banner page.
Add the banner page before each change in data based on a variable	In the <b>Variable</b> box, specify a variable to control when the banner page is included. For more information about variables, see <i>Using Data to Drive an Application</i> in the Exstream Design and Production documentation.
Add the banner page based on a rule	In the <b>Rule</b> box, set up a rule to control when the banner page is included. For more information about rules, see <i>Using Logic to Drive an Application</i> in the Exstream Design and Production documentation.

3. If you want to place the banner page after the options you set, rather than before, select the **Place after event** check box.
4. From the Menu bar, select **Edit > Save**.

## 6.2.3 Specifying Where to Place a Banner Page in Output

You can specify where you want the banner page to appear in the output. For example, you might want to place banner pages at the beginning of each break in the print stream.

To specify where to place a banner page in output:

1. In Design Manager, from the Library, drag a banner page object to the Property Panel.
2. In the **Where to put banner page** area, select one or more of the following check boxes:

To	Do this
Place the banner page at the beginning of each convenience break in the print stream	Select the <b>Beginning of each queue break</b> check box.
Place the banner page at the end of each convenience break in the print stream	Select the <b>End of each queue break</b> check box.
Place the banner page at the beginning of each print queue	Select the <b>Beginning of each queue</b> check box.
Place the banner page at the end of each print queue	Select the <b>End of each queue</b> check box.

For more information about convenience breaks, see “[Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages](#)” on page 318.

3. In the **Copies** box, enter the number of copies of the banner page to include in the output.
4. From the Menu bar, select **Edit > Save**.

## 6.2.4 Adding an Envelope Banner Page to Split Customer Documents into Separate Envelopes

Envelope banner pages are a sub-customer level banner page that can be placed after a specific number of pages. Envelope banner pages can be used to easily split a customer document into multiple envelopes. For example, if you can include only nine pages per envelope but a customer has 12 pages, you can create an envelope banner page to include the customer's address for the second envelope, which will contain the extra three pages.

With the envelope banner page in the Property Panel, use the envelope banner page properties to identify and customize how your envelope banner page is used.

If you are using the sorting and bundling capabilities in Exstream, you can use envelope banner pages for one pre-sort run or for multiple pre-sort runs. When you use multiple pre-sort engine runs, you must use the bundling feature in order for the envelope banner page to appear in the output.

For more information about sorting, see “[Setting Up Output for Sorting and Bundling](#)” on page 335.

To create an envelope banner page:

1. In Design Manager, in the Library, right-click the **Banner Pages** heading and select **New Banner**.

The **New Banner Page** dialog box opens.

2. In the **Name** box, enter a name. In the **Description** box, enter a description (optional).

3. In the **Page** box, click .

The **Select Page** dialog box opens.

4. Select the page object you want to use as a banner page and click **OK**.

The page name appears in the **New Banner Page** text box.

5. Select the **For customer enveloping** check box.

6. Click **OK**.

The new banner page opens in the Property Panel for you to define.

7. To control where the page count starts, select one of the following:

To	Do this
Start the page count at the first page of each bundle	Select the <b>First page of each envelope</b> radio button.
Start the page count at the last page of each bundle	Select the <b>Last page of each envelope</b> radio button.

8. If you want to use the envelope banner page for all envelopes, including the first, select the **Include in first envelope** check box.
9. In the **Pages** box, enter the number of pages you want between each envelope banner page. You must enter a value of **2** or higher.
10. From the Menu bar, select **Edit > Save**.

## 6.2.5 Specifying the Way an Output Queue Handles Banner Pages

To use a banner page in production, you must add the banner page object to an output queue. This task assumes that an output queue object has already been created.

To specify the way an output queue handles banner pages:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Banners** tab.
3. To select a banner page to add, click  . From the **Select banner page** dialog box, select a banner page.
4. Click **OK**.

The banner page appears in the banner page box.

5. From the Menu bar, select **Edit > Save**.

For more information about output queues, see “[Controlling and Setting Up High Volume Production Options with Output Queues](#)” on page 315.

## 6.3 Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects

With multiple-up options, you can create documents that require more than one design page on one printed page. For example, you can create booklets and brochures with multiple-ups. You can also print multiple design documents on one page to reduce production costs per sheet. For example, you can print multiple postcards or tickets on one large sheet.

If multiple-up output contains an odd number of pages for any customer, Exstream inserts blank pages in the empty multiple-up frames to break the output evenly. However, these blank pages are not considered part of the customer's output.

If you add search keys to multiple-up sheets, the engine processes search keys only for the first multiple-up frame that is filled. Because Exstream can insert blank pages in empty multiple-up frames, any search key that has been set to be placed at the end of each customer is placed after all customer pages.

For more information about search keys, see “[Adding Non-Printing Data or Production Device Controls with Search Keys](#)” on page 326.

Keep in mind that multiple-up output is not optimized for using accessibility tags. If you specify multiple-up options for an output queue with a PDF or PDF/A driver, accessibility tags are automatically disabled during the engine run, and you will receive a message. If the output driver is PDF/A-1a or PDF/A-2a, the output file will be compliant only with PDF/A-1b specifications.

For more information about PDF/A, see “[PDF/A](#)” on page 27.

This section discusses the following topics:

- “[Creating a Multiple-Up Object](#)” on the next page
- “[Setting Basic Options for a Multiple-Up Object in Design Manager](#)” on page 302
- “[Specifying a Dynamic Paper Type for a Multiple-Up Object](#)” on page 302
- “[Allowing Multiple Languages for Multiple-Up Objects](#)” on page 303
- “[Adding Content to Multiple-Ups with Multiple-Up Frames in Designer](#)” on page 303
- “[Using Bleeds to Prevent Unprinted Edges in Trimmed Multiple-Up Content](#)” on page 305
- “[Using Imposition to Specify the Page Order for Pages in a Booklet](#)” on page 309

- “Setting the Margins for Pages in a Booklet Using the Booklet Creep Option” on page 310
- “Specifying How to Order Customers on Multiple-Up Sheets” on page 311
- “Specifying the Way an Output Queue Handles Multiple-Ups” on page 314
- “System Variables You Can Use with Multiple-Ups” on page 314

### 6.3.1 Creating a Multiple-Up Object

Before you can create a multiple-up object, you must have set up a paper type object.

For more information about paper types, see *System Administration* in the Exstream Design and Production documentation.

To create a multiple-up object:

1. Depending on whether you are creating a multiple-up object in Design Manager or Designer, do one of the following:

To	Do this
Create a multiple-up object in Design Manager	In Design Manager, in the Library, right-click the <b>Multiple-ups</b> heading and select <b>New Multiple-up Sheet</b> . The <b>New Multiple-Up Sheet</b> dialog box opens.
Create a multiple-up object in Designer	a. In Designer, on the Standard toolbar, click  to create a new page. b. From the <b>What to create</b> list, select <b>Multiple-Up</b> . The <b>New Multiple-Up Sheet</b> dialog box opens.

2. In the **Name** box, enter a name.
3. In the **Description** box, enter a description (optional).
4. From the **Paper Type** drop-down list, select a paper type.
5. Click **Finish**.

In Design Manager, the multiple-up object opens in the Property Panel for you to define.

In Designer, the new multiple-up sheet opens in your design window.

**Note:** If you create a multiple-up object in Designer, remember to open the object in Design Manager to customize the properties.

## 6.3.2 Setting Basic Options for a Multiple-Up Object in Design Manager

1. In Design Manager, from the Library, drag a multiple-up object to the Property Panel.
2. From the **Orientation** drop-down list, select a page orientation for the multiple-up sheet. Make sure the orientation you select is compatible with the paper type you selected for the multiple-up object. You can select one of the following options:
  - **Portrait**
  - **Landscape**
  - **Portrait reversed**
  - **Landscape reversed**

**Note:** Inkjet output drivers (MIBF and IJPDS) require that multiple-up sheets be in portrait orientation.

3. If you want the multiple-up to print on both sides, select the **Duplex** check box.
4. If you want to specify a resolution, enter a resolution in the **Design resolution** box. For more information about design resolution, see *Designing Customer Communications* in the Exstream Design and Production documentation.
5. If you want to specify a bottom margin to retain on the sheet, enter a margin in the **Bottom margin** box.
6. From the Menu bar, select **Edit > Save**.

## 6.3.3 Specifying a Dynamic Paper Type for a Multiple-Up Object

You can specify more than one paper type for a multiple-up object to accommodate multiple production requirements. To specify multiple paper types for a multiple-up, use paper types that are dynamic (that is, they change at run time based on the value of a specified variable). For example, if one multiple-up object will be printed on two different printers that use different paper stocks, you can design the multiple-up one time. You can then use dynamic paper types to define the paper properties for each printer so that the postage calculations are correct for each paper stock.

For more information about dynamic paper types, see *Designing Customer Communications* in the Exstream Design and Production documentation.

If you specify a dynamic paper type, you must add the paper type to your application.

To specify a dynamic paper type for a multiple-up object:

1. In Design Manager, from the Library, drag a multiple-up object to the Property Panel.

2. In the **Dynamic paper type** box, click .

The Variable Palette opens.

3. From the variable list, select a non-array string variable that contains the dynamic paper type name.

4. Click **OK**.

5. From the Menu bar, select **Edit > Save**.

### 6.3.4 Allowing Multiple Languages for Multiple-Up Objects

While it is technically possible in certain scenarios to use multiple language layers on a MUP object, doing so can cause unexpected results in your MUP output. Therefore, as a best practice, you should not use language layers on MUP objects. You should instead place design objects only on the default layer, and then use language layers on individual components in your design (such as pages, paragraphs, and messages).

For information about using language layers in your design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 6.3.5 Adding Content to Multiple-Ups with Multiple-Up Frames in Designer

To add content to a multiple-up sheet, you must open the multiple-up object in Designer and add a multiple-up frame. Each multiple-up frame represents a full designed page. The order in which the frames are filled is specified by the fill order assigned to each frame.

Typically, multiple-up frames fill the entire sheet (one frame for each page), but you can add other objects to the multiple-up sheet, as well. For example, if the actual sheet area is larger than all of the combined multiple-up pages, you can add custom elements such as registration marks, color bars, and trim marks. Optionally, you can create these as components in Designer and save them in the Library. The only message frames you can use on multiple-up sheets are multiple-up frames.

For information about other types of frames, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Because multiple-up frames must fit precisely on a flat sheet, you must specify the exact positioning of the frame on the properties in Designer. Otherwise, you can receive error messages about the frame being off the page.

**Note:** Keep in mind that design layers within a multiple-up (MUP) sheet are processed by the engine before the engine processes any customer data. This timing difference means that customer data is not available to the engine when the engine processes the design layer content. Therefore, for best results, do not use rules on design layers to control the inclusion or exclusion of objects from customers within a MUP sheet. If you want to use customer data to include or exclude objects from customers within a MUP sheet design, you must place those objects directly into your design and use Library rules to control the inclusion or exclusion of those objects.

To design a multiple-up frame:

1. In Designer, open a multiple-up page.

2. Click .

The **Insert Frame** dialog box opens.

3. From the **Page paper types allowed in frame** drop-down list, select **Any** to allow any paper type to fill the frame, or select a specific paper type.
4. In the **Fill Order** box, enter a specific fill order for this frame in relation to other frames on the sheet. Exstream places pages on the multiple-up sheet according to the fill order you specify. There are additional settings on output queues that control page placement.

For more information about multiple-ups settings on output queues, see “[Specifying the Way an Output Queue Handles Multiple-Ups](#)” on page 314.

5. If this frame represents the first page in the second series of pages running from the top to the bottom of the sheet, select the **Starts (second) north/south group** check box.

For more information about the north/south setup for customers on multiple-up pages, see “[Specifying How to Order Customers on Multiple-Up Sheets](#)” on page 311.

6. From the **Draw Orientation** drop-down list, specify the orientation for the design page so that it fits in the multiple-up frame as you want it to appear in the output. You can select one of the following options:
  - **Portrait**—Rotates the page so that it is placed in the frame in portrait orientation
  - **Landscape**—Rotates the page so that it is placed in the frame in landscape orientation
  - **Portrait reversed**—Rotates the page so that it is placed in the frame in reverse portrait orientation
  - **LandScape reversed**—Rotates the page so that it is placed in the frame in reverse landscape orientation
  - **Based on paper type**—Places the page in the frame as it is designed
7. To create a border for the frame:

- a. Click inside the **Border** box.
  - b. In the **Offset** box, specify how much space to leave empty between the content of the frame and the border.

Negative numbers place the border inside the frame.
  - c. In the **Line properties** area, specify the line style, color, and weight.
  - d. Click **OK**.
  - e. To add lines for the border, click the edges of the **Border** box.
8. Click **OK**.

The frame appears on the page.

For more information about formatting frames, see *Designing Customer Communications* in the Exstream Design and Production documentation.

### 6.3.6 Using Bleeds to Prevent Unprinted Edges in Trimmed Multiple-Up Content

Pages in a multiple-up sheet are often cut apart or trimmed during post-production. Because paper can shift during the cutting process, you can create pages with design objects (such as images or banners) that extend beyond the edge of the design area and bleed into the margin. When these pages are included in a multiple-up, the bleed margin can be cut away during post-production to ensure that no unprinted edges appear in the final customer output.

Bleeds are supported only in PDF, PostScript, and VDX output. If you generate content that uses bleeds on any other output device, you can receive unexpected results.

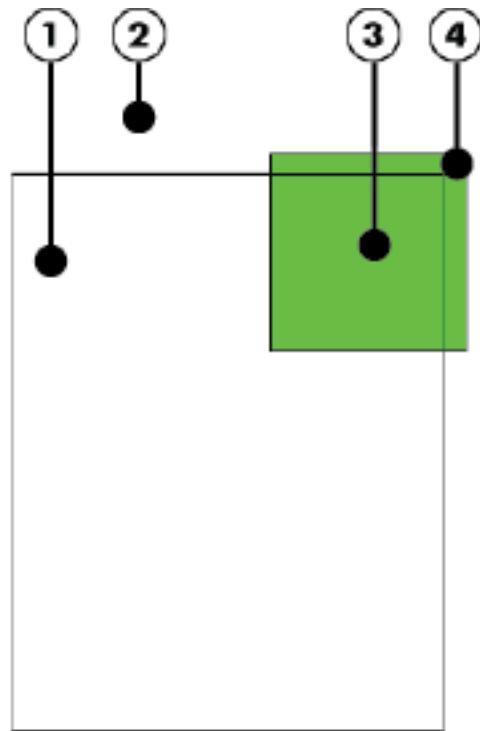
This section discusses the following topics:

- “[Design Considerations for Creating Pages That Will Be Included in a Multiple-Up with Bleeds](#)” below
- “[Setting Up Multiple-Up Frames to Accommodate Pages with Bleeds](#)” on page 307

### Design Considerations for Creating Pages That Will Be Included in a Multiple-Up with Bleeds

If you want to create pages with content that will be allowed to bleed when it is placed into a multiple-up, keep the following in mind:

- If you want a design to use bleeds when it is imported into a multiple-up, the original design must contain content that overlaps the design area and the pasteboard area.

**Example of a page with content that is intended for use with bleeds**

<b>1</b>	The design area
<b>2</b>	The pasteboard area
<b>3</b>	An object that overlaps both the design area and the pasteboard area
<b>4</b>	The bleed that will be contained in the bleed margin of the multiple-up frame

- You do not have to match the size of the bleed in the original page design to the size of the bleed margin that is set in the multiple-up frame; the bleed margin that is set on the multiple-up frame controls the amount of bleed that is applied to the final output. For example, suppose that the original design overlaps the edge of the page by 24 points, but you want to conserve ink and space in the final output. You can set the bleed margin on the multiple-up frame to 12 points to limit how much of the original design is allowed to bleed in the final output.

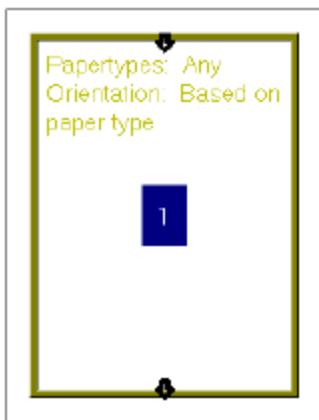
For more information about setting up a multiple-up frame with the appropriate bleed margin, see “[Setting Up Multiple-Up Frames to Accommodate Pages with Bleeds](#)” on the next page.

- As you design pages with bleeds, you can force the page border to be visible above design objects by clicking  on the Standard toolbar.
- Bleeds are supported only in PDF, PostScript, and VDX output. If you generate content that uses bleeds on any other output device, you can receive unexpected results.
- Pages with content that overlaps the pasteboard area can be generated with bleeds only if the page is generated as output using a multiple-up frame that allows bleeds. If the same pages are generated individually or generated in a multiple-up frame that does not allow bleeds, the final output is automatically cropped at the design area border.
- When you create duplex page designs, the front and back of the page are arranged side-by-side in Designer with a half-inch gutter between the two designs. Because of how the designs are arranged, bleeds in duplex designs can extend from either the front- or back-page design and into the gutter between the two designs; however, you should avoid designing bleeds that span the gutter to overlap both the front and back of the page. Bleeds that overlap both the front and the back of the page can produce unexpected results. If you want more space to accommodate bleeds between the front and back of a page, you can design the front page and back page as unique design pages and import each page separately into the multiple-up.

## Setting Up Multiple-Up Frames to Accommodate Pages with Bleeds

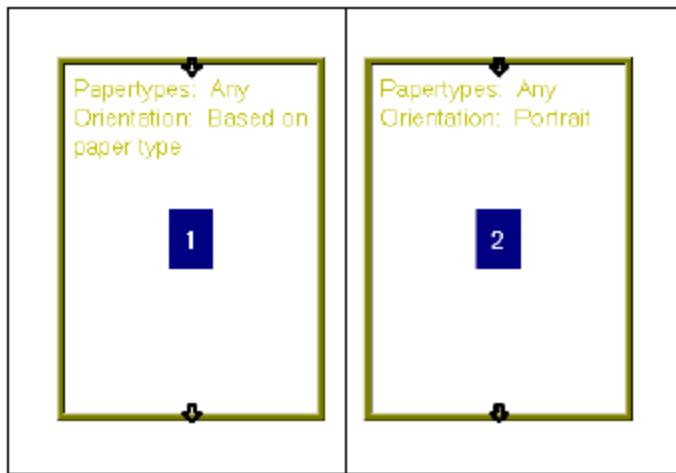
If you want to accommodate pages with bleeds in a multiple-up, you must set up the multiple-up frames to allow bleeds and specify the default size of the bleed margin to apply around each frame. The bleed margin controls how much of the original design content is allowed to bleed into the margin around the multiple-up frame. For example, even if the bleed in the original page design is 24 points, you can use the bleed margin to limit the bleed to 12 points to conserve ink and space in the final output. The default bleed margin that you specify is applied to all four sides of the frame and is indicated in Designer as a black border around the frame.

### Example of a multiple-up frame with a default bleed margin



As you arrange frames in your multiple-up, Designer automatically adjusts the bleed margin of intersecting frames to ensure that bleeds do not overlap in the final output. For example, suppose that you have set all your frames to have a 24-point bleed margin. If the total available space between each frame is 24 points, there would not be enough space to accommodate a 24-point bleed from two different frames. In this example, Designer automatically adjusts the bleed margins of each frame to 12 points to ensure that the bleeds from each frame do not overlap in the final output.

**Example of two multiple-up frames where the default bleed margin is automatically adjusted to prevent overlapping bleeds**



To set up multiple-up frames to accommodate pages with bleeds:

1. In Designer, open a multiple-up page.
2. Select a multiple-up frame.
3. Click .

The **Frame Properties** dialog box opens.

4. To allow the page that is added to the frame to bleed, select the **Allow bleed** check box.
5. In the **Bleed margin** box, specify the default size of the bleed margin to apply around the frame.
6. Click **OK**.

The **Frame Properties** dialog box closes and the bleed margin appears as a black border around the frame.

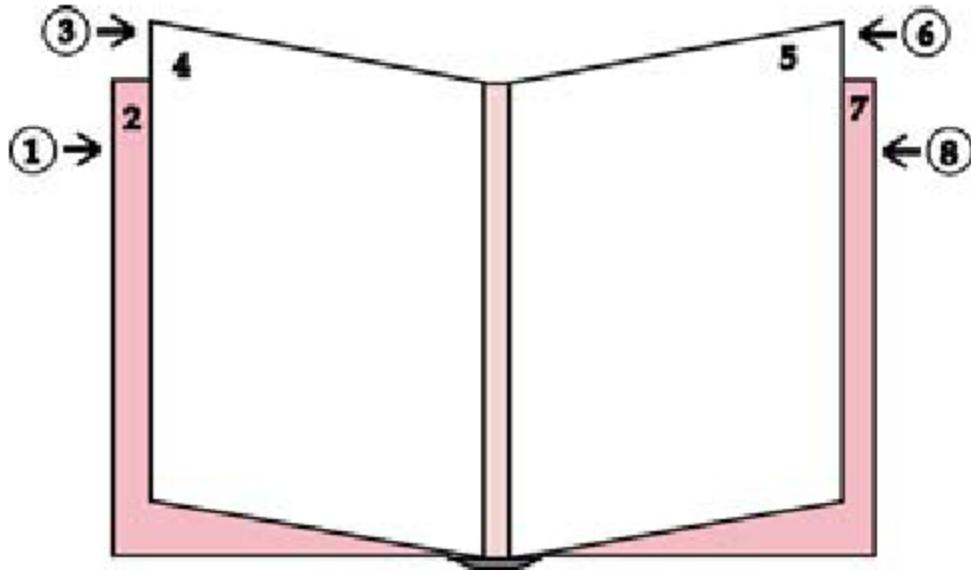
7. Repeat step 2 through step 6 for each multiple-up frame that must accept pages with bleed.
8. Review and adjust the placement of frames in the multiple-up as needed to accommodate the bleed margins that you specified.

### 6.3.7 Using Imposition to Specify the Page Order for Pages in a Booklet

You can arrange the pages for a multiple-up (MUP) object so that the pages are printed exactly the way that you want. For example, you can specify that they print in reverse order, or you can use imposition for binding in post-processing.

With imposition, the engine takes pages from a MUP sheet and arranges them in the correct order for binding in post-processing. In the following small booklet example, the outer duplex sheet is printed with pages 1 and 8 on the back and pages 2 and 7 on the front. The inner duplex sheet is printed with pages 3 and 6 on the back and pages 4 and 5 on the front. The engine automatically selects these various pages from a multiple-up sheet in the correct order.

#### Imposition example on a booklet



To specify the page order for a booklet:

1. In Design Manager, drag an output queue object from the Library to the Property Panel.
2. Click the **Sorting and Bundling** tab.
3. To arrange the pages, select one of the following options from the **Page sorting** drop-down list:
  - **2-up duplex imposition**—Arrange pages from a MUP sheet as duplex pages in the correct order for binding in post-processing.
  - **2-up duplex imposition with back cover page**—Arrange pages from a MUP sheet

as duplex pages in the correct order for binding in post-processing and add an additional page so that the back cover falls on the correct page.

- **2-up duplex imposition with 2 back cover pages**—Arrange pages from a MUP sheet as duplex pages in the correct order for binding in post-processing and add two additional pages so that the back cover falls on the correct page.
- **Reverse 2-up duplex imposition**—Arrange pages from a MUP sheet as duplex pages in reverse order for binding in post-processing.
- **Reverse 2-up duplex imposition with back cover page**—Arrange pages from a MUP sheet as duplex pages in reverse order for binding in post-processing and add an additional page so that the back cover falls on the correct page.
- **Reverse 2-up duplex imposition with 2 back cover pages**—Arrange pages from a MUP sheet as duplex pages in reverse order for binding in post-processing and add two additional pages so that the back cover falls on the correct page.
- **Reverse order**—Print all pages in succession from the last to the first.
- **None**—Disable imposition.

**Note:** When you select either **2-up duplex imposition with back cover page** or **2-up duplex imposition with 2 back cover pages**, the pages added by the engine must be the same paper type as the first inside page, not the outside page (cover), since the outside cover can use a different paper stock.

4. From the Menu bar, select **Edit > Save**.

### 6.3.8 Setting the Margins for Pages in a Booklet Using the Booklet Creep Option

After trimming, booklets you bind in the center of a sheet visually seem to have a gradual shift (or creep) in the outside margins from one page to the next. The booklet creep feature counteracts this shift by nudging the text towards the gutter on each page to give consistent outer margins when you trim the booklet in post-processing. Normally, the number you specify here is the thickness of the paper.

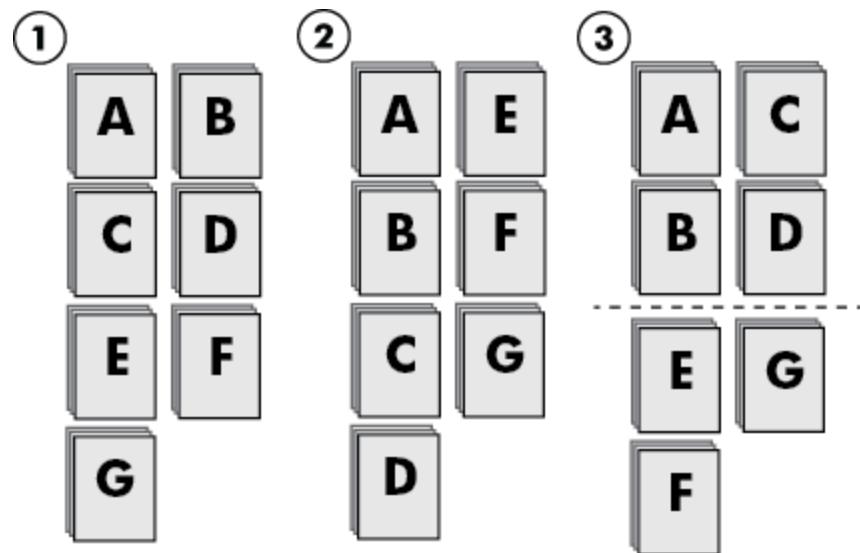
To set margins to accommodate for booklet creep:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Multiple-up** tab.
3. In the **Booklet creep** box, enter the thickness of the paper you are using for output.
4. From the Menu bar, select **Edit > Save**.

### 6.3.9 Specifying How to Order Customers on Multiple-Up Sheets

When you set up multiple-up (MUP) objects on the output queue, you must specify where to place customers and how to order them on the MUP sheets. You can place one or more customers on a sheet. If you place multiple customers on a sheet, you can set up the customers so that when the MUP sheets are cut, the customers are ordered in a specific manner. You can order the customers from east to west or from north to south. If you order customers from north to south and use post-sort processing, you can use convenience breaks to create groups of customers ordered north to south. For a simplified example of each method of ordering customers, see the following illustration.

**Customer ordering examples**



①	East-to-west customer ordering
②	North-to-south customer ordering
③	North-to-south customer post-sort ordering, with convenience breaks

For more information about post-sort processing, see “[Setting Up Post-Sort Processing](#)” on page 345.

For more information about convenience breaks, see “[Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages](#)” on page 318.

If you are using search keys after each customer and you want to place customers in a specific order, the engine divides your output and places each customer on a separate MUP sheet.

If you combine customers on individual MUP sheets, keep in mind that you cannot use the system variables intended to count MUP sheets (such as 'SYS\_MupPageInDocument') to count customer pages. The values that those variables return reflects only the number of MUP sheets in a run, not the number of customer pages.

For more information about counting- and numbering-related system variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

To specify how to order customers on MUP sheets:

1. In Design Manager, drag an output queue object from the Library to the Property Panel.
2. Click the **Multiple-up** tab.

3. To select the method in which to order the customers, do one of the following:

To	Do this
Start a new sheet for each customer	From the <b>Multiple-up method</b> drop-down list, select <b>Each customer starts a new sheet</b> .
Print multiple customers on each sheet, printing from left to right on each row of MUP frames	From the <b>Multiple-up method</b> drop-down list, select <b>Combine customers-east/west</b> .
Reorder customers during post-sort processing and print multiple customers on each sheet, printing from top to bottom on each column of MUP frames  To do this, you must license the Output Sorting and Bundling module.	<ul style="list-style-type: none"> <li>a. On the <b>Basic</b> tab, select <b>For postsort processing only</b> from the <b>Use</b> drop-down list.</li> <li>b. On the <b>Multiple-up</b> tab, select <b>Combine customers-north/south</b> from the <b>Multiple-up method</b> drop-down list.</li> <li>c. Select the <b>Sequentially reorder customers in sort index file</b> check box.</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <b>Caution:</b> If your application contains multiple post-sort output queues, be aware of the ordering method for each post-sort output queue. The engine cannot preserve the sort index order of the post-sort output queues that do not use north-to-south ordering. The engine will use north-to-south ordering for all post-sort output queues.       </div>
Reorder customers during post-sort processing and print groups of customers on each sheet, printing from top to bottom on each column of MUP frames  To do this, you must license the Output Sorting and Bundling module.	<ul style="list-style-type: none"> <li>a. On the <b>Basic</b> tab, select <b>For postsort processing only</b> from the <b>Use</b> drop-down list.</li> <li>b. On the <b>Breaks</b> tab, enter the number of customers you want in each group.</li> <li>c. Select the <b>Split after event</b> check box. If you clear the check box, the break occurs before the last customer in the group. For example, if you enter 12 in the <b>Customers</b> box, the break is before the twelfth customer, meaning that there are 11 customers in the group.</li> <li>d. On the <b>Multiple-up</b> tab, select <b>Combine customers-north/south</b> from the <b>Multiple-up method</b> drop-down list.</li> <li>e. Select the <b>Sequentially reorder customers in sort index file</b> check box.</li> <li>f. Select the <b>Use Customer Convenience Break as MUP Group</b> check box.</li> </ul> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <b>Caution:</b> If your application contains multiple post-sort output queues, be aware of the ordering method for each post-sort output queue. The engine cannot preserve the sort index order of the post-sort output queues that do not use north-to-south ordering. The engine will use north-to-south ordering for all post-sort output queues.       </div>
Print multiple customers on each sheet, printing from top to bottom on each column of multiple-up frames  To do this, you must license the Output Sorting and Bundling module.	<ul style="list-style-type: none"> <li>a. From the <b>Multiple-up method</b> drop-down list, select <b>Combine customers-north/south</b>.</li> <li>b. If you want specific customers in each column, make sure that your sorting program rearranges the sort index file. For example, suppose that you have 100 customers and you want customers 1–50 on the left and customers 51–100 on the right. Your sorting program must rearrange the sort index file so that customers alternate: 1, 51, 2, 52, 3, 53, and so on.</li> </ul>

4. From the Menu bar, select **Edit > Save**.

### 6.3.10 Specifying the Way an Output Queue Handles Multiple-Ups

To create multiple-up output, you must add a multiple-up object to an output queue. This task assumes that an output queue object has already been created.

If you add more than one multiple-up object to an output queue, keep in mind that the order of the multiple-up objects is important. When the engine selects a multiple-up object for a page, the first multiple-up that can use the paper type assigned to the page is used.

To specify the way an output queue handles multiple-ups:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Multiple-up** tab.
3. To add or remove a multiple-up object, do one of the following:

To	Do this
Add a multiple-up	<ol style="list-style-type: none"><li>a. Click  , and from the <b>Select an item to add to the list</b> dialog box, select a multiple-up.</li><li>b. Click <b>OK</b>.</li></ol> <p>The multiple-up name appears in the multiple-up list.</p>
Remove a multiple-up	Click 

4. To specify the order of multiple-ups in output, do one of the following:

To	Do this
Move a multiple-up object up in the list	Select a multiple-up and click 
Move a multiple-up object down in the list	Select a multiple-up and click 

5. From the Menu bar, select **Edit > Save**.

### 6.3.11 System Variables You Can Use with Multiple-Ups

Exstream provides several system variables designed to help you complete certain tasks when you create multiple-ups. You can use system variables with multiple-ups to add page numbers,

to track how many pages you print, and to set up rules. For example, you might want to count the actual pages and sheets used in creating the output for a specific customer.

For more information about system variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

- 'SYS\_MupPageInBreak'
- 'SYS\_MupPageInDocument'
- 'SYS\_MupPageInQueue'
- 'SYS\_MupPageInRun'
- 'SYS\_MupPagePhysicalInDocument'
- 'SYS\_MupPageTotalInDocument'
- 'SYS\_MupPageTotalPhysicalInDocument'
- 'SYS\_MupSheetInBreak'
- 'SYS\_MupSheetInDocument'
- 'SYS\_MupSheetInQueue'
- 'SYS\_MupSheetInRun'
- 'SYS\_MupSheetTotalInDocument'

## 6.4 Controlling and Setting Up High Volume Production Options with Output Queues

In Exstream, output queues function as containers for production settings, including those for output drivers, inserter objects, banner pages, and multiple-ups. Output queues are also known as production queues, or just queues. You can use multiple output queues to produce output with a variety of high-volume settings and output driver settings, including a mix of print and electronic drivers, while running the engine only once in Exstream. If you have licensed the Output Sorting and Bundling module, you use some options on an output queue specifically for the sorting and bundling capabilities within Exstream.

For more information about setting up for sorting and bundling, see [“Setting Up Output for Sorting and Bundling” on page 335](#).

After you have created an output queue object, you add it to the application object to which you want to apply the production settings. When you add an output queue to an application, you must consider the order of the output queues. By default, Design Manager processes output queue objects in the order in which they appear in the Library. The first output queue listed in an application is sometimes called the primary queue.

As you consider the order of output queues in the application, keep in mind that the output driver associated with the primary queue controls not only the settings for the primary queue, but also most settings for the secondary queues. You can, however, control settings for character fidelity and resource management on a per-queue basis. For example, if the character fidelity settings in the output driver for a secondary queue differ from those on the output driver for the primary queue, the engine honors the settings for the secondary queue.

For more information about how text quality is affected when you use multiple output queues, see [“Text quality considerations in multiple queue production” on page 192](#).

This section discusses the following topics:

- [“Creating an Output Queue Object” below](#)
- [“Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages” on page 318](#)
- [“Adding a Unique Name to Each Record in an Output File” on page 321](#)
- [“Offsetting Stacks of Printed Pages with Jogging” on page 322](#)
- [“Assigning a Connector to an Output Queue” on page 322](#)
- [“Generating Queue Report Files for Tracking and Quality Control” on page 323](#)
- [“Adding Information to the Output Manifest for Output Delivery” on page 324](#)

## 6.4.1 Creating an Output Queue Object

Before you can save an output queue, you must have created an output object.

For more information about creating an output object, see [“Creating an Output Object” on page 34](#).

To create an output queue object:

1. To select an application mode, complete one of the following tasks:

To	Do this
Create a new output queue object in SBCS mode	In Design Manager, in the Library, right-click the <b>Output Queues</b> heading and select <b>New Output Queue</b> . The <b>New Output Queue</b> dialog box opens.
Create a new output queue object in DBCS mode	In Design Manager, in the Library, right-click the <b>Output Queues</b> heading and select <b>New Output Queue</b> . The <b>New Output Queue</b> dialog box opens.
Create a new output queue object in SBCS/DBCS mode	<ol style="list-style-type: none"> <li>In Design Manager, in the Library, expand the <b>Output Queues</b> heading.</li> <li>Right-click the type of output of which you want to create a new output object. For example, select <b>SBCS Output Queues</b> for a new SBCS output queue object.</li> </ol> The <b>New Output Queue</b> dialog box opens.

2. In the **Name** box, enter a name.
3. In the **Description** box, enter a description (optional).
4. Click **Finish**.

The output queue object opens in the Property Panel for you to define.

5. From the **Output** box, click  . From the **Select output** dialog box, select an output object.
6. If you want to process the output queue but do not want to create actual output, select the **Do not create print file** check box. For example, you might want to create only a report file.

**Tip:** When you package an application, Exstream generates a small text file with a reminder that you disabled printing. To disable the reminder, use the **DISABLEPRINT=YES\_AND OMIT\_REMINDER** engine switch.

For more information about engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

7. In the **Rule** area, you can set a rule that determines whether to apply output queue settings for a package run based on specified criteria. For example, suppose you have an application where customers can receive anywhere from a one-page statement to a ten-page statement. Because the postage is significantly different for customers who receive more than three pages, you can add two queues to the application, one with a rule to apply settings for customers who receive three pages or fewer and one with a rule to apply settings for customers who receive more than three pages.

For more information about creating rules, see *Using Logic to Drive an Application* in the Exstream Design and Production documentation.

8. In the **Test file** box, specify a local output file for the application. By default, the engine writes this file to the Exstream program folder. To store the file somewhere else, click  to specify the full path. If you use convenience breaks to create multiple output files, the engine adds a break number to the file name for each file.

For more information about convenience breaks, see “[Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages](#)” below.

9. For electronic output, in the **Production file** box, specify a production file. If you use the same production file on more than one platform, you can specify a symbolic file name and identify specific file names in the control file by using the FILEMAP engine switch. If you use convenience breaks to create multiple output files, the engine adds a break number to the file name for each file.

For more information about engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

10. From the Menu bar, select **Edit > Save**.

To use the output queue settings for production, you must add the output queue object to an application object.

For more information about adding objects to an application, see *Designing Customer Communications* in the Exstream Design and Production documentation.

**Note:** If you choose to create a new output queue object by selecting **Insert > Environment > Delivery > Output Queue**, you must specify the application mode from the **Application mode** drop-down list in the **New Output Queue** dialog box.

## 6.4.2 Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages

You can use the convenience breaks feature to divide output into multiple files (sometimes called document subsets) based on various criteria. You can also use breaks to optimize postage bundling and for archival purposes. For example, you can create a separate output file for each customer, or you can separate the output into files of 5000 pages each. For electronic output, you might want to create a separate file for each customer or to separate files to meet file size requirements. For printed output, however, you might want to set up a variable to break the files at a specified number of pages for tray breaks. When this variable reflects a change, the engine breaks the output so that the printer operator can change the tray.

If you are using recipient copies, keep in mind that each recipient copy in the output is included in the customer count. Therefore, if you separate the output by customers, the engine counts both customers and recipients, and a break occurs when the specified count is reached. You can alternatively set up a variable or a rule to group recipient copies with associated customers or otherwise customize breaks in output with recipient copies.

For more information about using convenience breaks with recipient copies, see *Designing Customer Communications* in the Exstream Design and Production documentation.

You can also use convenience breaks during post-sort processing to divide output based on the number of customers. If you are using multiple-up (MUP) objects to place more than one design page on a printed sheet and you are ordering the customers placed on those sheets from north to south, you can use convenience breaks to create separate groups of north-to-south ordered customers. The engine places customers on the MUP sheets until it reaches the last customer in the convenience break. The north-to-south ordering restarts at the first customer of the next convenience break.

For more information about post-sort processing, see “[Setting Up Post-Sort Processing](#)” on [page 345](#).

For more information about MUP sheets and customer ordering, see “[Specifying How to Order Customers on Multiple-Up Sheets](#)” on [page 311](#).

If you are creating duplex output, keep in mind that if you break output into groups with an odd number of pages (for example, five pages each), a blank back will be added to make the page count even, resulting in a higher page count.

If you use multiple break options (for example, if you choose to split the output both at 1000 pages and when a variable changes), the breaks occur when either condition is true. If you want to split the output only when both are true, you must create a rule to test both conditions.

For more information about creating a rule, see *Using Logic to Drive an Application* in the Exstream Design and Production documentation.

To create a convenience break:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Breaks** tab.
3. In the **When to create convenience breaks** area, specify how you want to place a convenience break:

To	Do this
Break by page count	In the <b>Pages</b> box, enter the number of pages per file.
Break by document count	In the <b>Documents</b> box, enter the number of documents per file.
Break by customer count	In the <b>Customers</b> box, enter the number of customers per file.

To	Do this
Break when a variable changes	In the <b>Variable change</b> box, click  and select a variable.
Break when a rule condition is true	<p>a. Click in the <b>Rules</b> area. The <b>Rule</b> dialog box opens.</p> <p>b. Create a rule to determine when to divide the file. You can specify as many as 250 different conditions on a single rule, each on a separate line. To add more than 250 conditions, click <b>code</b>, and enter a rule in the Code Panel.</p> <p>c. Click <b>OK</b>.</p> <p>For more information about using rules, see <i>Using Logic to Drive an Application</i> in the Exstream Design and Production documentation.</p>

4. From the **Customer file method** drop-down list, select one of the following options:
  - **Normal**—The engine places customer documents in one or more output files as specified by other options.
  - **Each customer printable**—When multiple customers are placed in an output file, they retain their identity so they can be easily found and separated out later. The output file contains index records to help find each user. For example, this option could be useful if you want to simplify finding a particular customer inside a larger Computer Output to Laser Disc (COLD) archive file.
5. To divide the output after the break criteria is met (this is also called a break event), select the **Split after event** check box.
 

By default, the file is separated before the break event. For example, suppose you want to divide the output at each customer so that each customer is in a separate file. By default, Design Manager splits the output before the break event at every customer, resulting in one customer per file. In this case, the break event is a new customer. If you select the **Split after event** check box, Design Manager does not split the output after the first customer because the first customer starts the output. The second customer triggers the first break event, which means that the first two customers will be in the same file. If you are creating groups, the break occurs before the last object in the group. For example, if you enter 12 in the **Customers** box, the break is before the twelfth customer, meaning that there are 11 customers in the group.
6. If you want to allow documents for a customer to be split into multiple output files, select the **Can split customers** check box. This option is especially useful for electronic output where you want each page to be a separate file.
7. If you are producing electronic output files, do one of the following:

To	Do this
Use a variable	<p>In the <b>Variable for file naming</b> box, select a variable.</p> <p>Exstream appends an incremental integer to the name of each file in the same run. For example, if the value of the variable is <b>File</b> and you are creating PDF files, then the files might be named as the following sequence: <b>File1.pdf, File2.pdf, File3.pdf</b>, and so on. If you want more unique file names, you can use an array variable to name the files.</p> <p><b>Important:</b> If you produce output using the Exstream engine in the Communications Server, you must select a variable to name your file. You must also set your output path correctly in the output connector settings in your Communications Builder project. In Communications Builder, in the output connector settings, make sure that the <b>Create directories</b> check box is selected, and that your file path is in the format &lt;output_file location&gt;\%. The % character indicates to the engine that your output files should be placed in a directory that is named using the job ID from the engine run.</p>
Use the test or production file name	<ol style="list-style-type: none"> <li>Clear the <b>Variable for file naming</b> box.</li> <li>In the <b>Test file</b> box, specify a local output file for the application.</li> <li>In the <b>Production file</b> box, specify a production file.</li> </ol> <p>Exstream appends an incremental integer to the end of the name you enter in the <b>Test file</b> box or the <b>Production file</b> box, depending on the type of output you are creating. For example, if you enter <b>C:\Working_Folder\Output\exstream_PDF.pdf</b>, the first file is named <b>exstream_PDF.pdf</b>, the second file is named <b>exstream_PDF2.pdf</b>, and so on.</p>

- From the Menu bar, select **Edit > Save**.

### 6.4.3 Adding a Unique Name to Each Record in an Output File

If you want to add variable text at the beginning of each record in a print file, you can assign a variable to the output queue. For example, if you want to track when your output is printed, you can apply a date and time stamp at the beginning of each record.

To add a unique name to each record in an output file:

- In Design Manager, from the Library, drag an output queue object to the Property Panel.
- Click the **Basic** tab.
- In the **Record prefix** box, select a variable.
- From the Menu bar, select **Edit > Save**.

## 6.4.4 Offsetting Stacks of Printed Pages with Jogging

For printed output, you can use the jogging feature to offset groups of pages by 0.5 inches. Offsetting groups of pages, also known as offset stacking, provides visual cues for production personnel to easily separate output. For example, you might want to separate output by customer or at each banner page. Offsetting pages is especially useful when you are using envelope banner pages.

For more information about envelope banner pages, see “[Adding an Envelope Banner Page to Split Customer Documents into Separate Envelopes](#)” on page 298.

To specify how to offset stacks of pages:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Jogging** drop-down list, select one of the following options:
  - **Customer**—Groups pages by customer
  - **Banner page**—Groups pages before each banner page
  - **Bundle**—Groups pages by bundle. You can use this option only if you have licensed the Output Sorting and Bundling module and if you set up for bundling.
  - **None**—Disables the jogging feature

For more information about bundling, see “[Using Bundling \(Householding\) to Group Customer Information](#)” on page 343.

4. From the Menu bar, select **Edit > Save**.

## 6.4.5 Assigning a Connector to an Output Queue

You assign a connector object to an output queue so that a DDA routine can route output from the engine. You must assign a connector to an output queue so the engine can use the information it receives to determine what type of output to produce. This kind of routine generally routes output as a whole unit to an external connector module (for example, WSMQ Connector). In contrast to routines on an output object, a "routing" DDA on an output queue does not process data in chunks or return output to the engine.

For more information about assigning a connector to an output queue, see *Configuring Connectors* in the Exstream Design and Production documentation.

## 6.4.6 Generating Queue Report Files for Tracking and Quality Control

If you want to create audit or quality control files, you can create a custom report file (often a called queue report or a custom queue report) with an output queue. Unlike regular report files, queue reports generate queue-specific information rather than global information about an application. For example, you might want to track information about how many customers are processed through the queue or how many documents are sent through the queue. The report is generated as a composed text file and can be imported into other programs for formatting and data manipulation. You must have created a report file object before you can add it to the output queue.

For more information about creating report file objects, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

If you want to create multiple types of reports for an output queue, you can clone the queue, change the properties so that it does not create output, and then change the report settings as necessary.

For more information about cloning objects in Design Manager, see *Getting Started* in the Exstream Design and Production documentation.

To create a custom report for an output queue:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Reports** tab.
3. From the **Layout** drop-down list, select a report file object to provide variable mapping and the layout for the output queue report.

Only report file objects with the following I/O times are available for use with output queue report files:

- **After initial customer data**
  - **After initialization files**
  - **At end of customers, before campaign**
  - **Queue break and end of queue**
  - **Completion of customers**
4. To identify the file to use when testing on a local machine, click  next to the **Test file** box, and browse for the file you want to use.
  5. In the **Production file** box, enter the name you want to use for the report. The name must conform to naming conventions on your production platform.
  6. If you have set up convenience breaks for output, you can set the report file to break at the

same time by selecting **When queue breaks** from the **Break method** drop-down list. To disable this option, select **None**.

7. If you chose to break the report file when the output breaks, you can use a variable to name your report file(s). In the **Variable for file naming** box, select a variable. If you do not specify a variable, a serial number is appended to the name as files are created to ensure that your report file is not overwritten each time it breaks.

For more information about convenience breaks, see “[Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages](#)” on page 318.

8. If you use a z/OS platform for production, specify the expected size for report files in kilobytes in the **MVS file size (KB)** box. If you do not specify a value, the size of each file is set at 200KB.
9. From the Menu bar, select **Edit > Save**.

## 6.4.7 Adding Information to the Output Manifest for Output Delivery

The output manifest file is an XML-based data file that uses a pre-defined schema to provide a comprehensive list of the output composed in an engine run. The manifest file includes information from your Exstream Design and Production application, such as output queue sources, that can be used to deliver output to customers. If you want to include additional information in the output manifest, such as custom metadata for an output queue, you can provide this information on the **Delivery** tab on the output queue properties. If you are producing output for use with an email delivery connector, you can also specify email addresses and related information on the **Delivery** tab.

At engine run time, the output manifest file is created using the STDDELIVERYFILE engine switch. The information in the manifest file is then used to connect Exstream engine output to the output connectors in the Exstream platform environment.

**Important:** Do not manually include the STDDELIVERYFILE switch if you are using Communications Server for engine orchestration in the Exstream platform. The switch is passed automatically when the Communications Server application invokes the Exstream engine.

However, if you are not using the platform orchestration features, you can use the STDDELIVERYFILE switch to generate the manifest file and use the output delivery information for your specific requirements.

For more information about the STDDELIVERYFILE switch, see *Switch Reference* in the Exstream Design and Production documentation.

To add information to the output manifest for an output queue:

1. In Design Manager, in the output queue properties, click the **Delivery** tab.
2. If you are producing output for use with an email delivery connector, you can specify information that can be used to deliver the emails. You can enter static text or variables in the following boxes:
  - **To**
  - **From**
  - **Subject**
  - **CC**
  - **BCC**
  - **Attachment**
3. To add custom metadata fields to the output manifest:
  - a. Click the **Metadata** button.
  - b. In the Metadata dialog box, in the **Available Metadata** list, select a metadata object and then click  to apply it to the object.
  - c. In the **Applied Metadata** list, select the metadata object that you just applied to the output object.
  - d. In the **Metadata value** area, do one of the following:

To	Do this
Use a static value for the metadata object	<ol style="list-style-type: none"><li>i. Enter a text value in the <b>Value</b> field.</li><li>ii. Click <b>OK</b>.</li></ol>
Use a variable value for the metadata object	<ol style="list-style-type: none"><li>i. Click .</li><li>ii. In the <b>Select Variable</b> dialog box, select a variable from the list and click <b>OK</b>.<p><b>Note:</b> You should use a string variable when assigning a variable value to a metadata object. Other variable types can produce unexpected results in the output.</p></li><li>iii. Click <b>OK</b>.</li></ol>

## Related information

For information about configuring the Exstream platform and using the platform orchestration features, see the following documentation:

- *System Administration* in the Exstream Design and Production documentation
- *Preparing Applications for Production* in the Exstream Design and Production documentation
- *Creating Exstream Engine plug-ins* in the *OpenText Exstream: Communications Builder Configuration Guide*
- *Setting up Communications Server applications* in the *OpenText Exstream: Communications Builder Configuration Guide*

## 6.5 Adding Non-Printing Data or Production Device Controls with Search Keys

You can add data to the print stream for both electronic and print output using the search key feature. You can add data such as comments, searchable tags, instructions to print devices, and more.

Since you can customize the output driver, content, timing, and other properties, search keys can be used to define unique methods of counting or as a way of inserting output device or post-processing controls (for example, specifying the stapling options). Another way of using search keys is to add them so that you can later prove a named document was sent (for example, a government-required regulation notice). You can also use them to insert comments in PDF output in the form of No Operation comments (NOPs) and to create both Tagged Logical Elements (TLEs) and NOPs in AFP output.

You can create the following two types of search keys:

- **Application search keys**—If you know you are creating a search key that will be used in multiple locations, you can create an application search key and store it in the Library in Design Manager. Application search keys are placed in the print stream around content.
- **Locator search keys**—If your search key is for only one application, you can create a locator search key and place it directly in the content of the design in Designer. Locator search keys are placed in the print stream within content.

### 6.5.1 Using Application Search Keys

Application search keys are added to an application object in Design Manager, and can be placed in the print stream around content.

If you give a search key a name, Exstream prepends the search key name with ": " (a colon and a space) before the variable data. For AFP output, Design Manager uses the search key name as the TLE name without the colon and space. TLEs are the only exception, because TLEs are designed as true searchable keys, whereas NOPs and other comment types are not designed to be used for searching.

This section discusses the following topics:

- “[Creating an Application Search Key](#)” below
- “[Specifying Where to Place an Application Search Key](#)” below
- “[Specifying the Content of an Application Search Key](#)” on page 329
- “[Specifying Which Output Drivers Can Use the Search Key](#)” on page 329
- “[Specifying the Type of Application Search Key to Use](#)” on page 330
- “[Setting Up an Application Search Key for Sorting](#)” on page 331
- “[Adding an Application Search Key to an Application](#)” on page 331

## Creating an Application Search Key

1. In Design Manager, in the Library, right-click the **Search Keys** heading and select **New Search Key**.  
The **Create a New Search Key** dialog box opens.
2. In the **Search key** box, enter a name for the search key object.
3. In the **Description** box, enter a description (optional).
4. In the **Type** list, select **Application**.
5. Click **Finish**.

The search key opens in the Property Panel.

## Specifying Where to Place an Application Search Key

For most output types, you can place search keys before or after each customer, document, page, file, named page, or bundle. For PDF/VT output, you can place document part metadata (DPM) search keys on each customer, document, page, file, or named page.

For more information about using DPM, see “[Using Document Part Metadata for PDF/VT Output](#)” on page 150.

If you are using application search keys with multiple-ups and you choose to combine customers on a single multiple-up sheet, Exstream does not support search key placement after each customer. If you select to include your search key after each customer in this situation, Exstream divides your output and places each customer on a separate multiple-up sheet.

For AFP output, when you set up for sorting and bundling, TLEs set to be placed before or after each bundle are nested with multiple Begin Named Group (BNG)/End Named Group (ENG) pairs, similar to BNG/ENG nesting for customer and document TLEs.

To specify where to place an application search key:

1. In Design Manager, from the Library, drag an application search key object to the Property Panel.
2. To specify where you want to place the search key in the print stream, select one of the following options in the **Placement** list, depending on the [type of application search key](#) that you are using:

All other output types	PDF/VT metadata output type
<ul style="list-style-type: none"> <li>• Before each customer</li> <li>• After each customer</li> <li>• Before each document</li> <li>• After each document</li> <li>• Before each page</li> <li>• After each page</li> <li>• Before each file</li> <li>• After each file</li> <li>• Before each named page</li> <li>• After each named page</li> <li>• Before each bundle</li> <li>• After each bundle</li> </ul>	<ul style="list-style-type: none"> <li>• On each customer</li> <li>• On each document</li> <li>• On each page</li> <li>• On each file</li> <li>• On named page</li> </ul>

3. If you selected **Before each page**, **After each page**, or **On each page** in the **Placement** list, you can specify to include banner pages in the placement of the search key by selecting the **Include banner pages** check box.
4. If you selected **Before named page**, **After named page**, or **On named page** in the **Placement** list, you can specify further options:

To	Do this
Select the page before or after which you want to place the search key	<p>a. In the <b>Named page</b> box, click  .</p> <p>The <b>Select Page</b> dialog box opens.</p> <p>b. Select the page to use.</p> <p>c. Click <b>OK</b>.</p>
Include the page if it is a banner page	Select the <b>Include banner pages</b> check box.

To	Do this
Specify the pages on which the search key is placed	<p>In the <b>Which pages</b> list, select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>All pages</b>—Applies the search key to all instances of the named page</li> <li>• <b>First page</b>—Applies the search key only to the first instance of the named page</li> <li>• <b>Last page</b>—Design Manager places the search key only if the named page you specified in the <b>Named page</b> box is the last page in the output for a customer.</li> </ul>

5. From the Menu bar, select **Edit > Save**.

## Specifying the Content of an Application Search Key

1. In Design Manager, from the Library, drag an application search key object to the Property Panel.
2. In the **Value** area, in the **Type** list, specify the content (or value) of the search key:

To	Do this
Use a variable	<ol style="list-style-type: none"> <li>Select <b>Variable</b>.</li> <li>In the <b>Variable</b> box, click  to select the variable that sets the value of the key.</li> <li>If you do not want to add a search key if the variable has no value, select the <b>Do not create if variable is empty</b> check box.</li> </ol>
Use any text or combination of characters	<ol style="list-style-type: none"> <li>Select <b>Text</b>.</li> <li>In the <b>Text</b> box, enter the text you want to use.</li> </ol>
Use the name of the object where you place the key (for example, a page name or document name)	<ol style="list-style-type: none"> <li>Select <b>Name of object</b>.</li> <li>In the <b>Name</b> box, enter the name you want to use.</li> </ol>

3. From the Menu bar, select **Edit > Save**.

## Specifying Which Output Drivers Can Use the Search Key

1. In Design Manager, from the Library, drag an application search key object to the Property Panel.
2. In the **Where to include** list, specify which type of output in which you want to use the

search key:

To	Do this
Place the search key regardless of the output driver	Select <b>All outputs</b> .
Place the search key when one specific output driver is used	<p>a. Select <b>Specified PDL</b>.  b. In the list below, select the output where you want to place the search key.</p> <p><b>Tip:</b> When you specify a PDL, the search key will be in that output format, even if you set the <b>Type</b> setting to <b>As is</b>. To disable this feature, use the following engine switch: –DISABLE_SEARCH_KEY_CONVERT</p>
Place the search key on a specific output queue	<p>a. Select <b>Specified output queue</b>.  b. In the box below, click  and select an output queue.</p>

- From the Menu bar, select **Edit > Save**.

## Specifying the Type of Application Search Key to Use

- In Design Manager, from the Library, drag an application search key object to the Property Panel.
- In the **Type** list, select the type of search key to use:
  - Default type**—Uses the default search key setting from the properties of the output object.
  - As is**—Adds a record with no comment or NOP characters. This method can be used to add search data such as special indexing or Table of Contents tags. Use this method to enter custom printer commands and secondary device commands (for example, for staplers).
  - NOP comments**—Prefixes the search key with the appropriate No Operation (NOP) comment value for the driver (such as %). This method is useful for adding information that can be used for indexing, archiving, future machine instructions, or a very large instruction word (VLIW).
  - TLEs**—(AFP output only) Adds Tagged Logical Elements (TLEs). TLEs serve as begin and end records in AFP output. These can be array variables so that each element is included as a separate entry in the print stream.
  - NOP comments and TLEs**—(AFP output only) Allows the engine to write search keys to both TLEs and NOP comments in AFP output.

- **PDF/VT metadata**—(PDF/VT output only) Adds document part metadata (DPM) information to PDF/VT output. This method is useful for adding information that can be used for production (such as sorting and bundling) or information about the recipient (such as the corresponding ZIP code).
3. From the Menu bar, select **Edit > Save**.

## Setting Up an Application Search Key for Sorting

If you are sorting output with the [sorting and bundling](#) capabilities in Exstream, the timing of search keys is important when they are based on variables. Formulas that reference these variables are not automatically processed during post-sort processing. Because variables are based on data, and the type of data available is affected by sorting, the variable data can change if you are using pre-sort or post-sort output queues. If the value of the variable is affected by the values read from the sort index file, then the search key is also affected.

For information about setting up variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

To set up a search key for sorting:

1. In Design Manager, from the Library, drag an application search key object to the Property Panel.
2. If you are creating a search key based on a variable, you can specify when you want the value of the search key to be populated during the engine run in the **Sort timing** list.

Select one of the following options:

- **Before sort**—Populates the search key value during pre-sort processing. You also select this option if you are not using the sorting process.
  - **After sort**—Populates the search key value during post-sort processing.
3. From the Menu bar, select **Edit > Save**.

## Adding an Application Search Key to an Application

1. In Design Manager, from the Library, drag an application object to the Property Panel.
2. Click the **Search Keys** tab.
3. Click  and select one of the following:
  - **Add new search key**—Creates a search key
  - **Add library search key**—Adds a search key from the Library

4. Repeat step 3 as needed to add additional search keys to the application.
5. To place a search key in the order you want, select the search key in the list and click  or  as needed.
6. From the Menu bar, select **Edit > Save**.

## 6.5.2 Using Locator Search Keys

If you want to create a search key for only one application, you can create a locator search key (also known as a location search key or a location NOP) and place it directly in the content of the design. Locator search keys are placed in the print stream within content.

For more information about variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

This section discusses the following topics:

- “[Creating a Locator Search Key in Design Manager](#)” below
- “[Specifying the Content of a Locator Search Key](#)” below
- “[Specifying the Length of a Locator Search Key](#)” on page 334
- “[Adding Locator Search Keys in Designer for AFP or Metacode Output](#)” on page 334

### Creating a Locator Search Key in Design Manager

1. In Design Manager, in the Library, right-click the **Search Keys** heading and select **New Search Key**.  
The **Create a New Search Key** dialog box opens.
2. In the **Search key** box, enter a name for the search key object.
3. In the **Description** box, enter a description (optional).
4. From the **Type** drop-down list, select **Location NOOP**.
5. Click **Finish**.

The locator search key opens in the Property Panel for you to define.

### Specifying the Content of a Locator Search Key

For AFP and Metacode output, you must keep some considerations in mind when you set up locator search keys.

If you give a search key a name, Exstream appends the search key name with ":" (a colon and a space), to the search key name. For AFP output using TLEs, the Exstream engine uses the search key name as the TLE name, without the colon and space. TLEs are the only exception, because TLEs are designed as true searchable keys, whereas NOPs and other comment types are not designed to be used for searching.

The name you use must meet the following format requirements:

- The location search key name must be 8 characters.
- The starting offset of the variable data must be 4 characters.
- The ending offset of the variable data must be 4 characters.

For Metacode output, locator search keys do not contain offset information and are not limited to a specific number of characters. The search keys can contain a name (if supplied) and, if you use a variable or text to specify the content, a colon followed by a space and the excess data from the variable or text.

To specify the content of a locator search key:

1. In Design Manager, from the Library, drag a locator search key object to the Property Panel.
2. In the **Value** area, from the **Type** drop-down list, specify the value of the search key:

To	Do this
Use a variable to set the value of the key	<ol style="list-style-type: none"><li>Select <b>Variable</b>.</li><li>In the <b>Variable</b> box, click  to select the variable that sets the value of the key.</li><li>If you do not want to add a search key if the variable has no value, select the <b>Do not create if variable is empty</b> check box.</li></ol> <p>For AFP output, a colon and space are added, followed by the data from the variable or text.</p>
Use any text or combination of characters as the value of the key	<ol style="list-style-type: none"><li>Select <b>Text</b>.</li><li>In the <b>Text</b> box, enter the text you want to use.</li></ol>
Use the name of the object where you place the key (such as a page name or document name) as the value of the key	<ol style="list-style-type: none"><li>Select <b>Name of object</b>.</li><li>In the <b>Name</b> box, enter the name you want to use.</li></ol>

3. From the Menu bar, select **Edit > Save**.

## Specifying the Length of a Locator Search Key

1. In Design Manager, from the Library, drag a locator search key object to the Property Panel.
2. In the **Length** box, you can enter the number of bytes you want to specify as the length of the search key.
3. From the Menu bar, select **Edit > Save**.

## Adding Locator Search Keys in Designer for AFP or Metacode Output

You must have created a search key in Design Manager before you can add it to a variable in Designer.

To add a locator search key in Designer:

1. In Designer, right-click a variable and select **Variable Properties**.  
The **Variable Use Properties** dialog box opens.
2. From the **Search Key** drop-down list, select the search key you want to use.  
This drop-down list displays all existing search key objects in the Library.
3. Click **OK**.  
The search key is applied to the selected variable.

# Chapter 7: Setting Up Output for Sorting and Bundling

When you license the Output Sorting and Bundling module, Exstream provides the opportunity to set up for sorting data and to place multiple customer communications to the same address into one envelope (also called "householding"). After sorting the data stream with a third-party program, you can then complete the sorting and bundling process in Exstream. This process helps you simplify processes and improve overall production performance.

For more information about licensing options in Exstream, see *Getting Started* in the Exstream Design and Production documentation.

This chapter discusses the following topics:

- “An Overview of the Sorting and Bundling Process” below
- “Setting Up for Pre-Sort Processing” on page 339
- “Setting Up Post-Sort Processing” on page 345
- “Running the Engine for Pre-Sort Processing” on page 349
- “Merging Sort Index Files from Multiple Pre-Sort Engine Runs” on page 350
- “Sorting Files with an External Program” on page 355
- “Running the Engine for Post-Sort Processing” on page 356

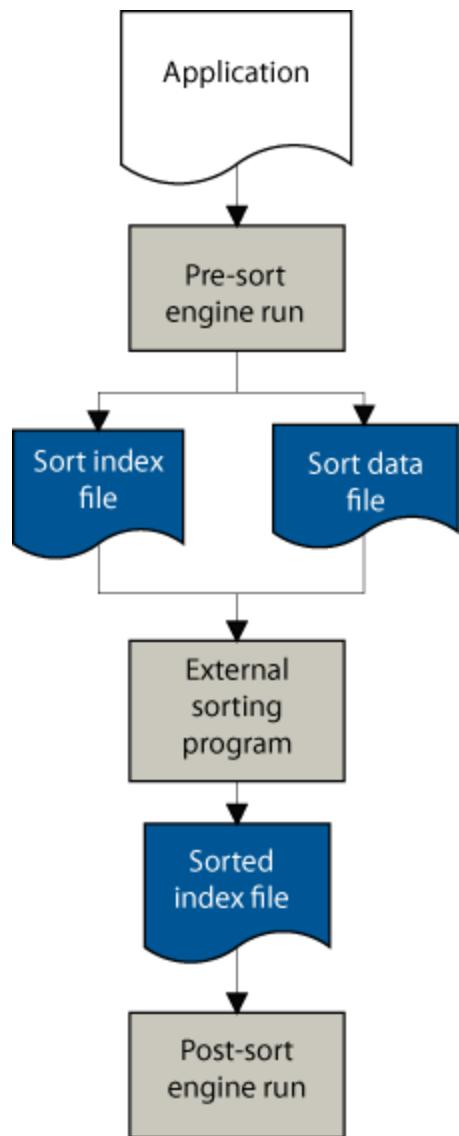
## 7.1 An Overview of the Sorting and Bundling Process

**Note:** The sorting and bundling process is not currently supported in applications that are fulfilled through the OT2 platform using Exstream Document Generation on Demand.

As you design the components of your application, you can optimize settings on design and Library objects to determine how and when the engine uses them for sorting and bundling. After you have set up an application for production, you run the engine twice.

The following graphic provides a basic overview of the sorting and bundling process:

#### Sorting and bundling process



The sorting and bundling process includes the following steps:

1. **Pre-sort processing**—After you have set up your application for production, you can set up to run the engine for the first time for pre-sort processing (also called pre-sort, presort, or Step One). Pre-sort processing produces customer data that can then be manipulated with third-party software to sort, cleanse, and update the files. For pre-sort processing, you can also set up for bundling, which is a convenient way to group customer documents. You can group customer documents in a number of ways, such as a group of customers at one address (also known as "householding") or in a hierarchical arrangement. When you are producing recipient copies, recipient document sets are indexed separately from customer

document sets so that you can sort and bundle recipient document sets appropriately based on the recipient data.

For more information about pre-sort processing, see [“Setting Up for Pre-Sort Processing” on page 339](#).

For more information about recipient copies, see *Designing Customer Communications* in the Exstream Design and Production documentation.

2. **Merging sort index files**—For multiple pre-sort runs, you can use either batch aggregation or the Exstream Application Consolidator, depending on whether you are combining runs from one application or multiple applications, to consolidate the information into a single output file. Using a single output file streamlines the production process, allowing you to use the sorting and bundling capabilities in Exstream for a large volume of documents. You must have licensed the Application Consolidator module to use the Application Consolidator.

For more information about batch aggregation and the Application Consolidator, see [“Merging Sort Index Files from Multiple Pre-Sort Engine Runs” on page 350](#).

3. **Sorting with a third-party program**—From the files you generate from the pre-sort run or runs, you can use a third-party (external) sorting software to reorder customer information for the best postal discounts. You can sort output based on information from customer driver files in many ways (for example, by state and ZIP Code).

For more information about sorting, see [“Sorting Files with an External Program” on page 355](#).

4. **Post-sort processing**—After the sorting process, you run the engine a second time for post-sort processing (also called post-sort, postsort, or Step Two). The engine uses the data in the sort data file(s) to produce final output according to the changes in the sorted index file and other application settings. The engine then produces output based on the sorted information.

For more information about post-sort processing, see [“Setting Up Post-Sort Processing” on page 345](#).

### 7.1.1 Timing Considerations for Variable Data

Before you begin the sorting and bundling process, you must consider how some Exstream objects are affected by timing. As you design your application, keep in mind the following timing considerations:

- **Composition time**—Using output sorting affects your choice of compose time for some design elements. The engine composes pages during the pre-sort processing step, but you might need to defer composing some objects until post-sort processing. Any Library or design object that is dependent on variable data (for example, search keys or text boxes) can be affected. Tables and text boxes that contain variable information affected by user sorting must be composed after that information is complete. For example, an address-cleansing

program can add or remove lines from an address block. To prevent odd spacing in the final output, you must set the composition time for the address block to after document sorting.

Since the engine composes the page before it composes the object, keep in mind the following:

- You must design for the maximum anticipated space in design objects that are dependent on variable data.
- You must set other design objects relative to these objects on design dimensions (not composed dimensions).
- You cannot set these objects to split and flow.

For more information about engine timing, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

For more information about design object properties, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Variable reset**—During post-sort processing, the engine resets variables each time it reads a new customer.

If you want to prevent this behavior, do one of the following:

- Initialize the variable with a post-sort initialization file.
- In the variable properties, set the variable reset time so that it never resets.

For more information about post-sort initialization files, see “[Creating a Post-Sort Initialization File](#)” on page 346.

For more information about variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

- **Search key compose time**—Exstream supports search keys in sorted output queues. If the search key references a system variable that is unavailable in pre-sort processing, then the engine composes the search key automatically in the post-sort run. This applies to all queue system variables such as 'SYS\_DocumentInQueue' and 'SYS\_PageInQueue'. However, if you base a search key on any other type of variable with a value that is not valid until after user sorting, you must set the timing of the search key to **After Sort** on the **Search Keys** tab of the application properties.

For more information about search keys, see “[Adding Non-Printing Data or Production Device Controls with Search Keys](#)” on page 326.

## 7.2 Setting Up for Pre-Sort Processing

You use the same package file for both pre-sort and post-sort processing engine runs, so you must define all objects and add them to your application before you create your package file.

When you run the engine for pre-sort processing, the engine creates the following pre-composed output files for you to use in the third-party sorting program to sort customer data:

- **Sort data file**—This file contains customer information in a format that you cannot change. To sort this data, the engine also generates a sort index file that you can change.
- **Sort index file**—This file points to customer data in the sort data file. If you run the engine multiple times, you generate multiple sort index files that you can combine with batch aggregation or with the Application Consolidator.

For more information about combining sort index files, see “[Merging Sort Index Files from Multiple Pre-Sort Engine Runs](#)” on page 350.

For more information about sorting, see “[Sorting Files with an External Program](#)” on page 355.

To set up for pre-sort processing, you must complete the following tasks:

1. “[Creating a Sample Sort Index File Layout](#)” below
2. “[Creating a Sort Index File Object and Mapping the Data](#)” on the next page
3. “[Setting Up an Output Queue for Pre-Sort Processing](#)” on page 342
4. “[Using Bundling \(Householding\) to Group Customer Information](#)” on page 343

You can also complete the following optional task:

- “[Setting Up Control Files for Pre-Sort Processing](#)” on page 345

For information about running the engine for pre-sort processing, see “[Running the Engine for Pre-Sort Processing](#)” on page 349.

### 7.2.1 Creating a Sample Sort Index File Layout

The sort file layout instructs the engine how to arrange data when it creates the sort index file. To make a sample sort index file layout, you must create a simple .txt file.

To create a sort index file layout:

1. Create a new .txt file in a program such as Notepad.
2. Reserve a 20-character area within the text file for the sort index key information. You can use spaces or any character, such as X. The engine writes the key to the file during the pre-

sort processing engine run and reads the key information during the post-sort processing engine run. You will map this area to the 'SYS\_SortIndex' system variable.

3. Enter columns of information in the specific sequence and fixed lengths you want them to appear in the sort index file. For example, if you want to use a five-digit ZIP Code, reserve five spaces. If you want to use the extended ZIP code, reserve ten spaces (to include the hyphen).
4. Enter data that uniquely identifies the customer and any data areas you plan to sort or change during the sorting step or during application consolidation.
5. Depending on how you want to sort data and which documents you want to include in bundles, you might want to include space in the sample layout text file for specific system variables you can map later (optional).

For more information about system variables commonly used for sorting, see ["System Variables Commonly Used for Sorting" on the next page](#).

6. For sorting or troubleshooting purposes, you might also want to leave room to map a variable that identifies a document with descriptive information (for example, name or publication number) (optional).
7. Save the file.

## 7.2.2 Creating a Sort Index File Object and Mapping the Data

After you have created the sort index file layout, you will map the data for the sort index file based on that layout.

To create a sort index file object and map the data:

1. In Design Manager, in the Library, right-click the **Data Files** heading and select **New Data File**.

The **New Data File** dialog box opens.

2. In the **Name** box, enter a name.
3. In the **Description** box, enter a description (optional).
4. From the **File Type** drop-down list, select **Sort index**.
5. From the **File format** drop-down list, select **Columnar data file**.
6. Click **Finish**.

The new data file appears in the Property Panel for you to define.

7. Click the **Basic** tab.

8. Next to the **Sample layout** box, click  and browse to the sort index file layout you

created.

9. From the Property Panel, drag the sort index file to the Edit Panel.
10. Highlight the 20-byte area you reserved for sort index keys, and map it to the 'SYS\_SortIndex' system variable. You can use spaces or any character, such as X. This variable contains information to find the customer document in the data file and cannot be changed or deleted during the sorting process.
11. Map the remaining data areas with variables used in the customer driver file.
12. Map other system variables, if applicable.
13. Save the sort index file object.

For more information about creating or mapping data files, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

## System Variables Commonly Used for Sorting

The following are some common system variables you might want to use as you map data for your sort index file and a description of when might you want to use them:

- **'SYS\_SortIndexDocument'**—Use the 'SYS\_SortIndexDocument' system variable when you need to exclude certain customer documents during sorting. You must have reserved 15 spaces or characters in your sample layout for this variable. During pre-sort processing, the engine generates a separate record for each document that qualifies to print for each customer. During sorting, you can delete records with the documents you want to exclude from the final output.
- **'SYS\_PagePrintStart' and 'SYS\_PagePrintEnd'**—Use the 'SYS\_PagePrintStart' and 'SYS\_PagePrintEnd' system variables when you need certain pages or documents in customer output. Documents are only affected if you use these variables in combination with the 'SYS\_SortIndexDocument' system variable. For the 'SYS\_PagePrintStart' system variable, the engine automatically inserts a 1 during pre-sort processing. For the 'SYS\_PagePrintEnd' system variable, the engine automatically inserts the value of 'SYS\_PageTotalInDocument' during pre-sort processing. During the sorting step, you can change these values as needed. If you enter a zero value in both variables, the document is not printed.

You must use the 'SYS\_PagePrintStart' and 'SYS\_PagePrintEndExclude' system variables together. If customers have multiple documents and you want to specify page ranges to be printed in each document, you can use the 'SYS\_PagePrintStart' and 'SYS\_PagePrintEndExclude' system variables with the 'SYS\_SortIndexDocument' system variable.

For more information about setting different bundling levels, see [“Using Bundling \(Householding\) to Group Customer Information” on page 343](#).

### 7.2.3 Setting Up an Output Queue for Pre-Sort Processing

When you set up an output queue for pre-sort processing, the engine generates a sort data file and a sort index file. The engine then stops so you can complete sorting tasks.

For more information about output queues, see “[Controlling and Setting Up High Volume Production Options with Output Queues](#)” on page 315.

To set up an output queue for sorting:

1. In Design Manager, from the Library, drag an output queue to the Property Panel.
2. Click the **Basic** tab.
3. From the **Use** drop-down list, select **Normal full production**.
4. Click the **Sorting and Bundling** tab.
5. Select the **Build files to enable sorting, bundling, and reprint** check box.
6. From the **Sort file layout** drop-down list, select the sort index file layout.
7. In the **Sort index file** box, specify a path and file name for the sort index file the engine generates. A sort index file points to customer data in the sort data file. If you are using a control file to run the engine for the pre-sort run, you can enter a symbolic name (for example, `sortindex`).
8. In the **Sort data file** box, specify the path and file name of the sort data file the engine generates. A sort data file is a file that contains customer information. If you are using a control file to run the engine for the pre-sort run, you can enter a symbolic name (for example, `sortdata`).
9. If you want the engine to use convenience breaks on the sort index file and sort data files, select the **Use convenience breaks for these files** check box.

The engine numbers the files sequentially (for example, `sortindex`, `sortindex1`, `sortindex2`, and so on). The engine continues to follow your break specifications during post-sort processing.

For more information about convenience breaks, see “[Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages](#)” on page 318.

10. Save the output queue object.

## 7.2.4 Using Bundling (Householding) to Group Customer Information

Bundling is a convenient way to group customer documents. These documents can be grouped in a number of ways, such as a group of customers at one address (also known as "householding") or in a hierarchical arrangement. You set up for bundling by applying at least one control variable on which to sort the data. If you want to bundle output within bundled information, you can apply multiple control variables. For example, you might use the customer state as the main bundling criteria, the customer city as a second-level variable, and postal codes as the third-level variable.

You can also add cover pages or trailer pages to bundles. Cover pages are inserted at the beginning of each bundle, and trailer pages are inserted after each bundle. Cover and trailer pages, unlike banner pages, are always included in the page counts for the bundle and can be designed with a barcode. For example, you can add a barcode to the trailer page to trigger an envelope on your output device.

For more information about banner pages, see ["Using Banner Pages to Mark Breaks in Output" on page 295](#).

For more information about barcodes, see ["Using Barcodes in Exstream" on page 256](#).

You can use the same page object for cover pages or trailer pages on multiple bundling levels. Remember that if you add design objects to the cover or trailer pages, you must set the objects to be composed after the documents are sorted. You set composition time properties on the **Dynamic Size and Placement** tab of the object properties in Designer.

For more information about design objects and page properties, see *Designing Customer Communications* in the Exstream Design and Production documentation.

Although the engine applies control variables during the post-sort processing engine run, you must set the bundling properties before you run the engine for pre-sort processing.

## Setting Up a Pre-Sort Output Queue for Bundling

1. In Design Manager, from the Library, drag a pre-sort output queue to the Property Panel.
2. Click the **Sorting and Bundling** tab.
3. In the **Control variable** column, click  and from the list of available variables, select a variable to identify each level of bundling. The first control variable is the main bundle, or the largest container. Each subsequent variable is a subcategory (or a different level) within the previous bundle. You can use any combination of system variables and user-defined variables. You can specify up to five levels of control variables.

For information about the system variables you can use with bundling, see ["System Variables You Use with Bundling" on the next page](#).

4. In the **Cover page** and **Trailer page** columns, click  to specify a cover page, trailer page, or both, for each set of documents within the bundle. You can selectively include cover and trailer pages by using a rule on the **Targeting** tab of the page properties. Cover pages and trailer pages are optional.

For more information about page properties, see *Designing Customer Communications* in the Exstream Design and Production documentation.

**Tip:** Since you can place barcodes on cover and trailer pages, you can use the system variable 'SYS\_BundleCustomer' to control an inserter for each bundle.

5. Save the output queue.

## System Variables You Use with Bundling

When you select a different control variable for each level, you can apply variables that apply to bundling to the design pages. You can use system variables to count customers, pages, printed sheets, or bundles. Each time a control variable is reset, the corresponding counter is reset, allowing you to track bundled documents accurately.

For more information about variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

Because you use a different variable for each level of bundling, you can set each level of page or customer formatting differently. So that system variables can help you assign levels of bundling and derive totals for each level, there are five separate versions of each bundling variables, each with a suffix number ranging from 1 to 5.

The following table lists system variables you can use with bundling:

System variables for bundling

Variable	Description
'SYS_BundleCustomer1' to 'SYS_BundleCustomer5'	Defines the current customer number in a given level of bundling
'SYS_BundleCustomerTotal1' to 'SYS_BundleCustomerTotal5'	Defines the total number of customers in a given level of bundling
'SYS_BundleInBreak1' to 'SYS_BundleInBreak5'	Gives the bundle sequence number within the current convenience break
'SYS_BundleInQueue1' to 'SYS_BundleInQueue5'	Provides more control of post-processing equipment by giving the sequence number of the current bundle within the current output queue
'SYS_BundlePage1' to 'SYS_BundlePage5'	Defines the current page number in a given level of bundling
'SYS_BundlePageTotal1' to 'SYS_BundlePageTotal5'	Defines the total pages in a given level of bundling

#### System variables for bundling, continued

Variable	Description
'SYS_BundleSheet1' to 'SYS_BundleSheet5'	Defines the current sheet count in a given level of bundling. Duplex operation counts two pages as a sheet, and simplex operation counts each page as a sheet.
'SYS_BundleSheetTotal1' to 'SYS_BundleSheetTotal5'	Defines the total sheet count in a given level of bundling

## 7.2.5 Setting Up Control Files for Pre-Sort Processing

Use the SORTID engine switch to customize the ID in your sort index key. Each sort index key in your sort index file must be different, or the keys can cause errors during the final engine run. By default, Design Manager creates unique sort index keys using the time the key was created. However, in large, high-speed, automated operations, two sort index keys can be created at the same time, especially during consolidated application runs.

Use the DISABLESORT engine switch when you want to produce your final output without sort index files or sort data files.

For more information about control files and the engine switches you can use, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

## 7.3 Setting Up Post-Sort Processing

The second engine run, known as post-sort processing, produces the final output for the sorted files. After you have set up for pre-sort processing, you must set up for post-sort processing. The engine run uses data stored in the sort data file(s) according to the updated customer data in the sorted index file. When you run the engine for post-sort processing, the engine generates bundles, places barcodes, generates post-sort report files, prints banner pages, and composes design components that have been set up to compose after document sorting.

To set up for post-sort processing, complete the following tasks as needed:

- “[Setting Up an Output Queue for Post-Sort Processing](#)” on the next page
- “[Creating a Post-Sort Initialization File](#)” on the next page
- “[Setting Up a Control File for Post-Sort Processing](#)” on page 347

For information about running the engine for post-sort processing, see “[Running the Engine for Post-Sort Processing](#)” on page 356.

### 7.3.1 Setting Up an Output Queue for Post-Sort Processing

Because of engine timing and how the engine uses output queues to direct output, you cannot use the same output queue for pre-sort processing and for post-sort processing. When you create a post-sort output queue, it can be used during post-sort processing only.

If you are using multiple-up (MUP) objects to place more than one design page on a printed sheet, you can use post-sort output queues to place customers in order from north to south during post-sort processing. You can also use convenience breaks on the post-sort output queue to divide the customers into separate groups of north-to-south ordered customers. The engine places customers on the MUP sheets until it reaches the last customer in the convenience break. The north-to-south ordering restarts at the first customer of the next convenience break.

For more information about convenience breaks, see “[Using Convenience Breaks to Separate Output into Multiple Files or into Groups of Printed Pages](#)” on page 318.

For more information about MUP sheets and customer ordering, see “[Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects](#)” on page 300.

To set up an output queue for post-sort processing:

1. In Design Manager, from the Library, drag an output queue object to the Property Panel.
2. Click the **Basic** tab.
3. From the **Use** drop-down list, select **For postsort processing only**.
4. From the Menu bar, select **File > Save**.

### 7.3.2 Creating a Post-Sort Initialization File

The post-sort initialization file works like any initialization file, but the engine reads the data only at the beginning of the post-sort process. This file lets you make data available in the post-sort engine run that is not required in the pre-sort run. For example, if your external sorting software produces a code that you want to place in the final output, you can add this information in the post-sort initialization file.

For more information about initialization files, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

You can use a post-sort initialization file to do the following:

- Set variable values that are not affected by the pre-sort process.
- Override any variable values set in the pre-sort process.
- Identify the variable values that you do not want to be reset. By default, Design Manager

resets variables each time the engine reads a new customer during the post-sort process. If you do not want variables to be reset, map these variables in a post-sort initialization file.

Make sure that you add all initialization and post-sort initialization files to the application object.

To create a post-sort initialization file:

1. In Design Manager, in the Library, right-click the **Data Files** heading and select **New Data File**.
2. The **New Data File** dialog box opens.
3. In the **Name** box, enter a name.
4. In the **Description** box, enter a description.
5. From the **File type** drop-down list, select **Post-sort initialization file**.
6. From the **File format** drop-down list, specify the data format of the external data source from the following options:

To	Do this
Sort data in columns	Select <b>Columnar data file</b> .
Separate data with a delimiter character	Select <b>Delimited data file</b> .
Use XML format for the data. This option is available only if you have licensed the XML/JSON Input module.	Select <b>XML data file</b> .
Place data in an ODBC database. This option is available only if you have licensed the ODBC Access module.	Select <b>ODBC data source</b> .
Place data in a PDF XFA form. This option is available only if you have licensed the PDF Form Miner or PDF Form Pre-Fill module.	Select <b>PDF Form</b> .

6. Click **Finish**.

The post-sort initialization file opens in the Property Panel for you to define.

For more information about defining data files, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

### 7.3.3 Setting Up a Control File for Post-Sort Processing

To indicate the sort index file and the sort data file you want to use for final production, you must set up a control file.

For more information about control files, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

Use the following engine switches to set up a control file for post-sort processing:

Engine switch	Description
CONTROLFILE	The CONTROLFILE switch enables nested control files that you can either use or comment out as needed for specific runs. Use an asterisk (*) to comment out a control file. Exstream ensures that each control file is referenced only once.
SORTINDEX	The SORTINDEX switch specifies the sort index file. If you include the SORTINDEX switch, the engine uses the control file for the post-sort processing step.  You can have only one sort index file.
SORTDATA	The SORTDATA switch specifies one or more files to use as data files for post-sort processing. If you use the SORTDATA switch, you must also use the SORTINDEX switch. If you include the SORTDATA switch, the engine uses the control file for the post-sort processing step.  You can specify multiple sort data files by listing the SORTDATA engine switch for each file.  <b>Caution:</b> In a Windows environment, you can specify a maximum of 512 sort data files in your control file. If you have more than 512 files open, the operation will fail.
PACKAGEFILE	The PACKAGEFILE switch specifies the same package file you used during the pre-sort process engine run. This switch is required.
POSTSORTQUEUE=ALL	The POSTSORTQUEUE switch, with the ALL parameter, specifies that all post-sort output queues are included. If you do not use the ALL parameter, the output queue name(s) must match exactly and are case-sensitive.  <b>Caution:</b> If you are using north-to-south ordering of customers and you use the POSTSORTQUEUE=ALL switch, be aware of the ordering method for each output queue. The engine cannot preserve the sort index order of the output queues that do not use north-to-south ordering. The engine will use north-to-south ordering for all post-sort output queues.  For more information about north-to-south ordering, see <a href="#">"Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects" on page 300</a> .
APP_PACKAGEFILES	The APP_PACKAGEFILES switch specifies multiple sub-package files for sorting.
NORTHSOUTHINDEX	The NORTHSOUTHINDEX switch specifies the location in which to write and read the sort index file. By default, when ordering customers north to south, the engine deletes the temporary sort index file from memory. When you use this switch, the engine writes the file to location specified in the switch. Then, the engine reads the sort index file from that location.  For more information about sorting customers during post-sort processing, see <a href="#">"Printing Multiple Design Pages on One Printed Page with Multiple-Up Objects" on page 300</a> .

For more information about engine switches, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

If you used either batch aggregation or the Application Consolidator to combine multiple pre-sort processing runs, you must do the following in your post-sort control file:

- Use only one SORTINDEX engine switch to point to the consolidated sort index file.
- List all sort data files referenced by the consolidated sort index file. Use a SORTDATA

engine switch for each sort data file you include.

If you used the Application Consolidator, you must also do the following:

- Identify each package file with a PACKAGEFILE engine switch.
- If you want to include sub-package files in the control file for post-sort processing, you must list each sub-package file in a separate APP\_PACKAGEFILES engine switch per application. This switch points to an external text file that lists the package files derived from the same application. The application package file must be at the top of the list and then the document or campaign sub-package files follow in any order.

## 7.4 Running the Engine for Pre-Sort Processing

For all platforms, use your normal method of running the engine.

Unless you also create an output queue for regular output, no customer output is produced during the packaging run. Since the sort data file and the sort index file are external files, neither appears in the Library.

When processing applications for production, the engine automatically breaks large sort data files into 2GB files. The engine then appends a suffix number to the names of the separate sort data files. For example, if sortdata.dat is too large because of memory limitations, the engine creates sortdata1.dat, sortdata2.dat, sortdata3.dat, and so on. This is true even if you select a variable from the **Variable for File Naming** box on the **Basic** tab of the output queue properties. The engine creates a maximum of 62 sort data files.

Using multiple sort data files does not require the Application Consolidator module or batch aggregation. These tools are necessary only when you are consolidating multiple sort index files from multiple pre-sort processing engine runs.

For more information about batch aggregation and the Application Consolidator, see “[Merging Sort Index Files from Multiple Pre-Sort Engine Runs](#)” on the next page.

The sort index file contains either one record for each customer or, with the 'SYS\_SortIndexDocument' variable, one record per document. The third-party sorting program you use for sorting uses the information in the sort index file to arrange your customer data. The index file contains a 20-byte key that points to the location in the sort data file for each customer's data.

After you run the engine, you use the external sorting program to sort the data.

For more information about sorting, see “[Sorting Files with an External Program](#)” on page 355.

## 7.4.1 Running the Engine for Pre-Sort Processing on the z/OS Platform

On the z/OS platform, sort data files must be allocated as Entry Sequence VSAM files. The files break if the file size exceeds the allocated dataset size, or when the file size exceeds 2GB, whichever comes first. You must specify the RECORG=ES parameter in the allocation parameters.

For example:

```
//SORTDAT DD DSN=HLVEL1.HLLEV2.SORTDATA,  
//           DISP=(NEW,CATLG,DELETE),RECORG=ES,  
//           DCB=(LRECL=32756,BLKSIZE=32760),  
// SPACE=(???,(?,?),RLSE),AVGREC=U
```

When the files break, the engine automatically calls for the next Data Definition (DD) name in the sequence. In the previous example, the DD name is SORTDAT. If the files break in this example, the DD names for the subsequent breaks ( DD:SORTDAT2, DDSORTDAT3, and so on) must be included in the JCL file.

## 7.5 Merging Sort Index Files from Multiple Pre-Sort Engine Runs

With each pre-sort processing engine run, you create one sort index file. If you want to merge the data from multiple-pre-sort engine runs so that they are all sorted together with the third-party program, you can use one of the following tools:

- **Batch aggregation**—You use batch aggregation to merge sort index files created from multiple runs of the same application using the same package file. This tool is available with the Output Sorting and Bundling module.
- **Application Consolidator**—You use the Application Consolidator to merge sort index files created from runs of different applications using different package files. This tool is available if you have licensed the Application Consolidator module.

For more information about licensing options, see *Getting Started* in the Exstream Design and Production documentation.

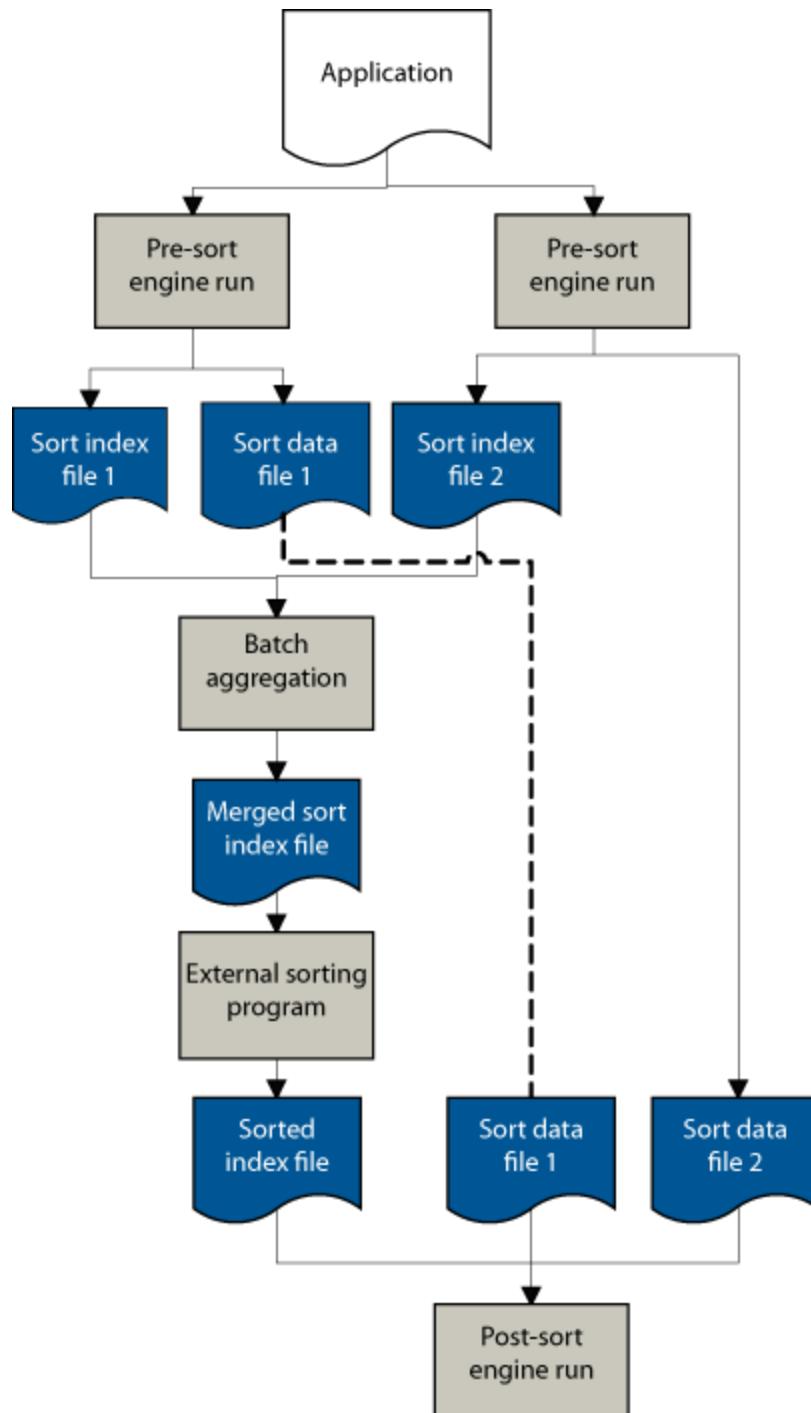
## 7.5.1 Using Batch Aggregation to Merge Sort Index Files from Multiple Runs of One Application

**Note:** Batch aggregation is not currently supported in applications that are fulfilled through the OT2 platform using Exstream Document Generation on Demand.

Batch aggregation lets you combine multiple pre-sort engine runs of one Exstream application into a single output file. For example, suppose you run a production output file every Monday through Thursday night. The actual publishing takes place on Friday, but you want to sort all of the files together for the best postal rates. You can do a pre-sort run each day, and then use batch aggregation to combine the files.

Batch aggregation is also helpful when you must create output for a specific customer but you are not ready for the finished production. When you are ready to package, you can generate output with a normal output queue and also generate pre-sort files with a pre-sort queue during the same package run.

The following graphic provides an overview of the sorting process when you use batch aggregation:

**Sorting and bundling process, with batch aggregation**

To use batch aggregation, you must use the same package file for each engine run. When you set up for batch aggregation, you must make sure that you have included all sort data files in your post-sort control file.

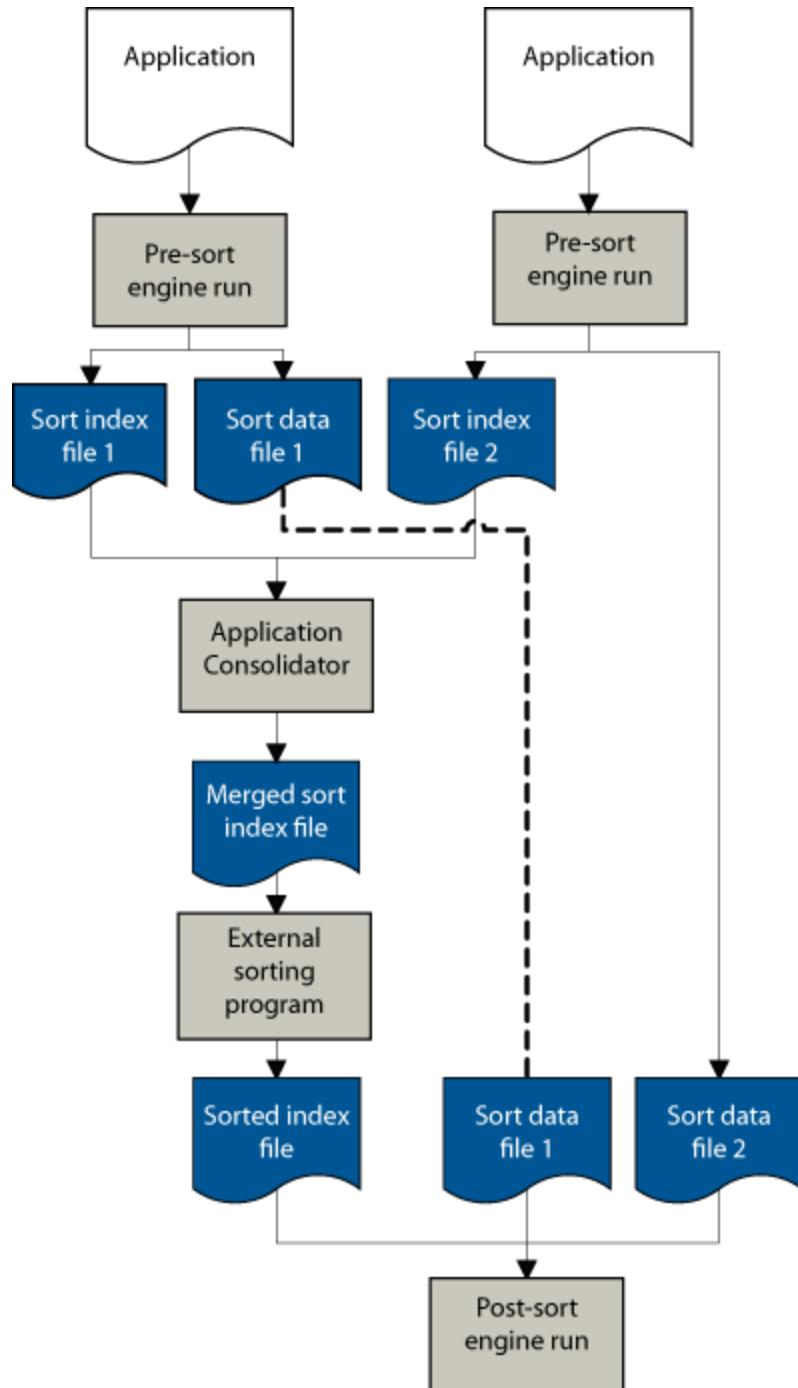
## 7.5.2 Using the Application Consolidator to Merge Sort Index Files from Different Applications

**Note:** Application consolidation is not currently supported in applications that are fulfilled through the OT2 platform using Exstream Document Generation on Demand.

If you want to combine and sort output from different applications, you can license the Application Consolidator module. The Application Consolidator module is licensed separately from the Output Sorting and Bundling module.

One reason you might need to set up different applications is to accommodate different output driver requirements. For example, suppose you are creating output for an event that requires sending customers a letter, event tickets, and parking passes. Because of the design of the tickets and parking passes, as well as the output device you must use, you have decided to set up different applications. You can package these applications separately for multiple pre-sort runs, and then use the Application Consolidator to combine and sort all of these design components for easy bundling and mailing.

The following graphic provides an overview of the sorting process when you use the Application Consolidator:

**Sorting and bundling process, with the Application Consolidator**

You can manually combine only sort index files. Sort data files and package files must remain as output from the Exstream engine. You list each sort data file and each package file separately in the post-sort control file.

For each engine run in the pre-sort processing step, keep in mind the following guidelines:

- All applications must use the same output queue configuration. This supports a common sort index file among different package files. The engine stops and generates an error if it detects differences in the output queues.
- You must use applications from the same database. The Application Consolidator module does not support running two applications from two different databases.
- Make sure that all objects with the same name that are used in multiple runs do not differ between runs. For example, if a variable type differs among package files, the engine stops and issues an error. To reduce conflict with different properties on objects with the same name, the Exstream engine identifies the first specified package file as the main package file. The engine keeps the properties for objects in the main package file in the memory. If the engine encounters different versions of these objects in other package files, the engine overwrites these properties with those from the main package file.
- When you use the Application Consolidator, all common objects must have the same version number. Because effectivity can trigger changes in versioning, applications packaged with different effective dates will not work with the Application Consolidator. If common objects use different effective dates, each object will reflect its version number on a specified date or range of dates. If versioning has changed from one date to the next, then the engine stops and issues an error.
- When you use the Application Consolidator, all output-related objects must have the same version number, and must have been modified on the same date. The engine stops and generates an error if it detects an output-related object that does not have the same version number or modification date in all package files.
- All package files must use the same method for selecting effectivity dates. For example, all package files can use the **As of Now** option or the **Date Range** option, but you cannot select the **As of Now** option for one package file, and the **Date Range** option for another package file. Because effectivity can trigger changes in versioning, applications packaged with different effective dates will not work together with Application Consolidator. If the method for selecting effectivity dates is not the same in all package files, then the engine stops and issues an error.

## 7.6 Sorting Files with an External Program

After you have completed the pre-sort processing step, you can use one or more third-party programs to alter the sort index file so that you can sort and bundle the customer information as needed. Third-party programs are available to help you to cleanse addresses, optimize for mailing, reprint copies, bundle customers, insert last-minute changes, and other options.

When you sort the files with an external program, you produce a sorted index file (as opposed to the sort index file), which you use in the post-processing step for final output.

For more information about post-sort processing, see “[Setting Up Post-Sort Processing](#)” on [page 345](#).

## 7.6.1 Sorting Considerations

As you sort with a third-party program, keep in mind that you must not change or move the 20-character sort index key variable values in the sort index files created from pre-sort processing. You can sort them, but do not change any information.

### Understanding How Data Can Change

If you included the 'SYS\_SortIndexDocument' or the 'SYS\_PagePrintStart' and 'SYS\_PagePrintEnd' system variables in your sort index file, entries in the engine-generated sort index file can appear to be different.

If you used the 'SYS\_SortIndexDocument' system variable to map data, the engine generates a separate record for each document that is available to print for each customer. The engine uses encoded information to point to the document information in the sort data file. Do not change or move this value. You can delete records for documents you want to exclude from the final output by referencing another mapped variable that identifies a document with descriptive information (for example, name or publication number).

If you mapped the 'SYS\_PagePrintStart' and 'SYS\_PagePrintEnd' system variables, the engine inserts a 1 for the 'SYS\_PagePrintStart' variable and the value of 'SYS\_PageTotalInDocument' for the 'SYS\_PagePrintEnd' variable. You can change these values as needed. If you do not want to print the document, enter a zero value in both variables.

For more information about these system variables, see “[System Variables Commonly Used for Sorting](#)” on [page 341](#).

## 7.7 Running the Engine for Post-Sort Processing

For all platforms, use your normal method of running the engine.

The engine uses the data in the sort data file(s) to produce final output according to your changes in the sort index file and other application settings.

For more information about running the engine, see *Preparing Applications for Production* in the Exstream Design and Production documentation.