opentext™

OpenText™ Exstream™

Importing Designs

Design and Production Documentation

Release 16.6.0

OpenText™ Exstream
Importing Designs
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

Contents

Chapter 1: About importing designs

If you maintain legacy applications or designs outside of Design and Production, you can use conversion tools to convert those designs into an Exstream Design and Production-compatible format. Converted designs can be stored as page objects in the design database, just like pages you create using Designer. For example, suppose you created and maintained legacy mailings using Adobe InDesign. You can use the InDesign Converter to convert those files and import them into the Design Manager Library. You can then interact with the designs in Design Manager and Designer just as you do with other types of objects.

Design and Production provides you with multiple conversion options for legacy designs:

- If you want to import xPression designs, you can use the built-in xPression import tool in Design Manager to convert your xPression designs into Design and Production designs.
- If you want to import static legacy designs, that is, designs that do not use dynamic data, you can use converter tools to convert your designs into the Exstream Exchange Fomat (DXF) and then import them into Design and Production. The following converters are available for importing static legacy designs into the Design and Production environment:
  - InDesign Converter
  - Metacode Converter
  - OGL Converter
  - PDF Converter
  - Quark Converter

  Each converter is available as a module that can be enabled on your license key.

If you have legacy content that you do not need to edit in Designer for use in future applications, you might consider using another Design and Production tool to meet your needs. For example, you can use a third-party tool to convert content into a format that you can use with the Dynamic Content Import module, such as TIFF or PDF. In this scenario, you can import legacy content into applications at engine run time, and you do not store them in the design database.

For more information about importing external content into your applications, see *Importing External Content* in the Exstream Design and Production documentation.

1.1 Overview of the DXF conversion process

The basic conversion process to convert a static design into DXF is essentially the same, regardless of the converter program you use.

1. Converters analyze the legacy design to identify the types of objects used in it, and whether the objects are supported by the converter. Objects not supported by the converter or by the Design and Production environment are converted differently, depending on the converter you are using.
2. After the file has been analyzed, the converter converts the design into an intermediate format called DXF. DXF is an XML-based format that is fully recognized and supported in the Design and Production environment. One DXF file is created per legacy file, even if the legacy design has multiple pages. The DXF file maintains information about each page so that when you import the DXF file, each page from the legacy design becomes a page object in the design database.
3. Using Design Manager, you import the DXF file into the Library. When you carry out this step, the new, Design and Production-compatible design is added to the design database as page objects, and you can interact with the page just as you can interact with pages designed in Designer.

Basic conversion process

Some converters handle specific objects in such a way that they do not appear exactly the same as the original after they have been converted and imported. You can make minor adjustments to these objects in Designer so that they appear as they did before the conversion. Information about how to clean up some types of objects is provided with the respective converter information.

1.2 Understanding the DXF format

The DXF format is a Design and Production file type that is used by converter tools and in other Design and Production design processes. DXF is an XML file type, based on the XSL

Formatting Objects (XSL-FO) standard. When using a converter to import designs, you do not need to interact with the DXF files created by the converter. However, if you want to use the XML representation of the design in other processes, you can learn more about the DXF format by referencing other Exstream Design and Production documentation. You can also create a sample DXF file from existing Exstream objects to help you better understand the DXF format and the representation of the objects you want to import.

For more information about DXF, see the *DXF Reference* in the Exstream Design and Production documentation.

1.2.1 Creating a sample DXF file

In working with DXF, it can be helpful to understand the XML structure for existing Exstream design objects. You can export a DXF file from the Library to see the structure for existing objects that are similar to objects you want to import. Because DXF is XML, you can open the file in any XML or text editor.

When exporting DXF, keep in mind the following considerations:

- Although most design objects are supported in DXF, certain objects with limited support might not be exported fully.
- Some objects that are not part of the physical design, such as design groups, are not supported in DXF and are therefore not exported.
- If you plan to import the sample DXF back into the design environment, you should first make sure that the DXF contains all of the expected objects and properties before overwriting existing objects.
- If the exported DXF contains attributes that reference objects in the Library, you must import the DXF into the same design database to keep those references intact. Keep in mind that DXF that contains such references cannot be imported into a different database at run time. Additionally, if you import DXF that contains such references into a different database at design time, the import process might create new objects to replace the missing referenced objects, and the objects that are imported from the DXF might need modification to function as expected.

To create a DXF file from an existing design:

1. In Design Manager, in the Library, right-click the Library object that you want to export to DXF.
2. Click **Export to DXF**.

Chapter 2: Importing xPression designs

The built-in import utility in Design Manager lets you import your xPression designs and convert those designs into a Design and Production-compatible format. Converted designs are stored as objects in the design database, just like objects that are created in Design Manager and Designer. You can then interact with the designs in Design Manager and Designer and produce output using the Exstream engine.

> **Note:** You can use the import utility to import only xDesign documents; xPresso documents and CompuSet applications are not supported.

The xPression import utility can convert most xPression features into equivalent Design and Production objects. However, some of the features used in xPression designs are not supported as Design and Production objects, and you might need to modify the imported design in Design Manager or Designer so that your output appears as expected.

This section discusses the following topics:

2.1 xPression document feature support

You can import an xPression document PDPX file into Design and Production to import your xPression designs. The import utility in Design Manager can convert most xPression document features into equivalent Design and Production objects. However, some features are not supported or might not be imported correctly. Before importing a PDPX file, OpenText recommends that you familiarize yourself with how objects in an xPression document will appear after the import.

The following information describes the import behavior for xPression document features in Design and Production. For information about output profile features, see "xPression output support" on page 19.

2.1.1 Content types

Content type support

Feature	Import behavior
Charts	Charts are not supported and will not be imported. However, you can use the traditional charts or advanced charts functionality in Design and Production to recreate the charts in your designs.
Content groups	Content groups are converted into sections in Design Manager. • Content items that are contained in these groups are converted into paragraph objects. Each shared content item is converted into a separate paragraph object in the Shared Contents And Rules folder in the Library, and references to that paragraph object are added to the sections that were created for each content group that contained that shared content item. • Criteria on the content groups are converted into Library rules.
Images	Images that are referenced in xDesign documents are imported, and retain their original dimensions when you open your designs in Designer. Embedded images are imported as embedded images in Design and Production. Linked images are imported as image placeholders. You might need to validate that the correct images are referenced in these placeholders. Alternate text on images is also imported.
Signatures	Signatures and signature placeholders are not supported.
Subdocuments	**xDesign subdocuments** xDesign-based subdocuments are supported. During the import process, a separate folder is created for each category that is referenced in the PDPX file, and the subdocument is imported into the category folder to which it belongs. Each subdocument is imported as a complete application with its own driver file. If more than one data source exists in the imported data source group, the subdocument driver file uses the primary data source in the group. If your subdocument is mapped by value, any table mappings that are contained in this subdocument will not be imported. Additionally, nested subdocuments are not fully supported. Any structure, rules, or mapping in the embedded subdocuments is lost during the import process. If your subdocument uses the CompuSet publishing engine, any custom CompuSet commands that are used in the content are ignored during the import process. **xPresso subdocuments** xPresso-based subdocuments are not supported.

Content type support, continued

Feature	Import behavior
Universal content	**Static universal content** The following static universal content types are imported as placeholder variables in Design Manager: • DOCX files • GIF, JPEG, and PNG image files • SVG image files (HTML output only) • PDF files • TIFF files The following content types are not supported: • Multi-page TIFF files • BMP and EPS image files • PDF and TIFF as single-page image files **Dynamic universal content** If your design uses a literal name to reference xDesign-based universal content, and if both the design document and the referenced document are in the same PDPX file, then the referenced content is imported as a new section into Design Manager. The reference to the subdocument is imported as a placeholder variable. If your design uses a field or a variable to reference universal content, the import utility cannot process that reference, and the referenced content is not imported. xPresso-based universal content is not supported.
Variables	xDesign variables are converted into Design and Production variables. • Data-related variables are mapped to variables in the driver file that is created by the import utility. • User-defined variables are imported but not mapped to the driver file. • User exits are supported. Keep in mind that the formatting for imported variables might not exactly match the original formatting of the variables in xDesign and your output might contain some visual differences.

2.1.2 Data sources

Data source support

Feature	Import behavior
Data source references	References to the data sources in your designs are retained during the import process. A separate folder is created in the Library in Design Manager for each data source group that is contained in the PDPX file.

Data source support, continued

Feature	Import behavior
Data source formats	The import utility creates an XML-formatted driver file object that points to the data source file that is referenced in the PDPX file. During the import, if the data source file is located in the same directory as the PDPX file, then that data source is used as the mapping and test data source in the driver file in Design Manager. By default, the import utility maps the customer driver file as if it were an XML data source. If you are using a data source that is in a format other than XML, you must do one of the following: • Before you create the PDPX file, export your data to an XML file and specify the exported XML file as your data source; or • After the import process is complete, change the format of the customer driver file to match your data source. If you do this, you must re-map your data file before you can package your application. For more information about defining the customer driver file format and mapping a data file, see *Using Data to Drive an Application* in the Exstream Design and Production documentation

2.1.3 xDesign rules

Rules in the xDesign tree structure appear as sections in Design Manager. Library rules are assigned to these objects to provide the same behavior as the criteria in xDesign. If your design uses shared rules, each shared rule is converted into a section object in the Shared Contents And Rules folder in the Library, and references to this section object are added wherever the shared rule appears in the imported application. Shared criteria are also converted into Library rules, and these rules are then assigned to any objects that used the shared criteria.

xDesign rule support

Feature	Import behavior
Chart rules	Chart rules are not supported.
Content rules	Content rules are converted into sections in Design Manager. Content items are converted into paragraph objects. If your design uses conditional rules with selection criteria, the criteria are converted into usage rules on the sections that correspond to the content groups that are contained within a content rule. Overriding attributes on content rules are not supported.

xDesign rule support, continued

Feature	Import behavior

GoTo rules	Supported GoTo rules are converted into data sections in Design Manager. Most GoTo implementations are supported, including nested GoTo rules such as:
	`Label A -> Label B -> GoTo B -> GoTo A`
	However, interlaced (non-nested) GoTo rules are not supported. For example, the import utility cannot process the following GoTo implementation:
	`Label A -> GoTo B -> GoTo A -> Label B`
	Supported GoTo implementations are imported in the following manner:
	• **Summing rules**—A rule that is used to sum up the values that are obtained from a subtable is converted into an array variable and a sum variable that returns the sum of the array. You must map the array variable to the array in the driver file.
	• **Repeating content items**—A rule that is used to repeat content based on a subtable is converted into a section that contains the repeating paragraphs, which are driven by a data section. If the rule is used to repeat content based on a set count, or on a condition that is set in a variable rule, the import utility will attempt to process the repeat count and insert the repeating content into a section that is set to repeat as many times as the original content.
	• **Repeating tables**—A rule that is used to repeat table rows based on a subtable is converted into a table that contains a data section on the repeating row. This new table is added to the Automated Table folder in the Design Manager library.
	• **Paragraph list items**—A rule that is used to list items in a paragraph (for example, "KY, IN, and OH") is converted into a paragraph object that contains an array variable. The array variable properties depend on the merge property of the original paragraph and the presence of delimiters.
	• **Skip rules**—A rule that is used to skip content in the xDesign tree is converted into a section with a usage rule on it. The section is excluded if the conditions of the usage rule are met.

Read and Read Next rules	Read rules are supported only if they appear in a GoTo loop. All other read rules are not supported. When read rules are encountered during the import process, the import utility cannot process custom `WHERE` clauses in specific queries. Read rules are not converted into corresponding objects; however, the content items that are associated with the read rules are imported. This can impact variable content and prevent business logic from being processed correctly when generating output, and you might need to review and redesign your application to include this content. For example, if your design contains a read rule with a secondary repeating node for transactions, that repeating node is converted into a data section. If that rule in your design contained a `WHERE` clause that selected only the transactions with a negative amount, then you can apply a usage rule on the data section to select only those transactions that meet the condition `IF TRANSACTIONS < 0 THEN INCLUDE ENDIF`.
Section rules	Section rules are converted into sections in Design Manager.
Subdocument rules	A subdocument rule is imported as a section, and has the same name as the subdocument rule in the xPression master document. If your subdocument rule is mapped by value, any table mappings that are contained in this subdocument rule will not be imported.

xDesign rule support, continued

Feature	Import behavior

Table rules	When a table rule is encountered during the import process, the import utility creates a paragraph object that contains a table with the content that is referenced by the table rule.
	
	Table rule settings are imported in the following manner:
	
	• **Data source group and table**: This setting corresponds to a data section in Design and Production. All of the rows in the table that is created by the import utility will be set to repeat on a data section that matches the data source table name for the table rule. Keep the following table design considerations in mind:
	  ◦ The last table in the content item that uses the specified data source group and table will be set to repeat in the imported design. All rows from that table will also be set to repeat, even if some of the rows do not use any data.
	  ◦ If multiple tables access the same data, only the last table will be set to repeat. Tables that do not access the data will not be set to repeat.
	  ◦ If the content item does not contain any tables, then the entire imported paragraph will be set to repeat on the data section.
	  ◦ Any text content that appears before or after the table will appear as-is.
	• **Read criteria**: Read criteria that are set on the table rule are converted into usage rules on the appropriate rows in the table generated in Design and Production.
	• **Read order**: This setting is not supported for automatic sorting. However, you can map your data into arrays and then use the built-in sort function on the array before you set your table rows to repeat based on the number of elements in the array.
	• **Criteria**: This setting corresponds to the usage rule for the table.

Universal content definition	Universal content definition rules are supported. These rules, however, are not converted into equivalent objects in Design Manager. Instead, information from the rules is used to correctly import referenced universal content.
Variable rules	Variable rules are converted into empty sections in Design Manager, and the rule logic is converted into usage rules on those sections. User exits are supported; If the variable rules in your PDPX document contain user exits, those user exits are imported into Design Manager as library functions. A separate function is created for each user exit definition that is contained in your PDPX file. To build output from imported applications that contain user exits, you must configure your environment to reference the correct user exit JAR files. After the import is complete, review the timing of the variables to make sure that your output appears as expected.

2.1.4 Versioning and workflow

Versioning and workflow support

Feature	Import behavior
Content item versions	Versioning for content items is supported. When you import a PDPX file, you have the option to import only the latest version or all versions of the content items included in the xDesign documents. This is controlled using the options on the **Content item versions** list on the **Import xPression design** dialog box. • **Show only the latest version in the content group**—This is the default option. The import utility creates a design object for the latest version of each content items in a content group, with the version number appended as a suffix in the object name. To see previous versions of the object, in the Design Manager Library, right-click the object and select **History**. • **Show all versions**—The import utility creates a separate design object for each version of the content items in a content group. You might need to re-order objects in the Design Manager Library to create the correct output.

Versioning and workflow support, continued

Feature	Import behavior

Approval workflow	xPression workflow states are supported. Documentum workflow states are not supported.
	
	Before you create the PDPX file, review the approval states of content items to determine whether they should be marked as **Approved** in xDesign, based on the following import behavior:
	
	• Content items that are in an **Approved** state in xDesign are imported with the **Approved** state applied in Design Manager.
	• Content items that are in the **Withdrawn** state are not imported and a message is issued in the log file.
	• Content items in **Pending**, **Submitted**, **Rejected**, or any custom state in xDesign are imported with the **Work in Progress** state applied in Design Manager..
	
	Shared content
	
	If your document contains shared content items, the versions that you choose to import will affect the approval state of the design objects that are created in Design Manager.
	
	For shared content items, the **Approved** state is applied only to the major version of each shared content item in its content group (for example, version 2.0). If the latest version of the shared content item is a minor version (for example, version 2.01), the corresponding imported design object has the **Work in Progress** state applied in Design Manager.
	
	For example, suppose that your xDesign document contains two content groups with multiple versions of a shared content item:
	
	• Content Group 1 has two approved major versions of the content item (1.0 and 2.0).
	• Content Group 2 has two major versions that are approved and one minor version that is not approved (1.0, 2.0, and 2.01).
	
	When you import the PDPX file, depending on the versions that you choose to import, the following versions of each shared content item will be imported:
	
	• **Show only the latest version in the content group**—The latest version of each shared content item is imported as a single design object in the Design Manager Library, with previous versions available from the history.
	  • For Content Group 1, version 2.0 will be imported with the **Approved** state applied.
	  • For Content Group 2, version 2.01 will be imported with the **Work in Progress** state applied.
	• **Show all versions**—Each version of each shared item is imported as a separate design object in the Design Manager Library.
	  • For Content Group 1, versions 1.0 and 2.0 will be imported with the **Approved** state applied.
	  • For Content Group 2, versions 1.0 and 2.0 will be imported with the **Approved** state applied, and version 2.01 will be imported with the **Work in Progress** state applied.

Jurisdictions	Jurisdiction attributes are supported. You might need to validate that the imported application has the correct jurisdiction variable mapped from the driver file, or that the system jurisdiction variable value is set correctly.
Effective dates	Effective date attributes are supported. You might need to validate that the effective date variable is mapped correctly in the driver file, or that the system effective date variable value is set correctly.

2.1.5 Microsoft Word templates

Template style support

Feature	Import behavior
Page setup	**Page, section, and column breaks** Page breaks and column breaks are supported. Section breaks are imported as page breaks, but they might not always be imported correctly. **Columns** Multi-column content is supported. Content in this format appears in a text box in Designer. **Headers and footers** Headers and footers are not fully supported. The import utility recognizes headers and footers in xPression designs, and imports them as separate paragraph items into a new folder named **Headers and Footers** in the Design Manager Library structure that is created for your imported designs. You can then use Designer to manually reincorporate these paragraph items into your design. Odd and even page settings and different first page settings for headers and footers must be implemented manually using flow page options in Design and Production. **Page numbering** Page numbering fields are imported as system variables in Design and Production: • The **Page** field is imported as the SYS_PageInDocument system variable. • The **NumPages** field is imported as the SYS_PageTotalInDocument variable. The system variables are automatically inserted in the paragraphs that contain the **Page** field and the **NumPages** field. To implement page numbering correctly, you must also insert these variables manually to the flow pages in your design. **Page borders** Page border settings are not supported.

Template style support, continued

Feature	Import behavior
Paragraphs	**Borders and shading** Border and shading settings are not fully supported. The import utility recognizes border and shading settings; however, Design and Production does not support left or right borders, so those settings will be ignored. In addition, top and bottom border settings cannot be set separately in Design and Production; the last setting encountered during the import will be used as the value for top and bottom borders. **Hyphenation** Hyphenation is not supported. **Lists** Lists are supported. However, numbered list formatting might not be imported correctly, and numbered lists might appear as bulleted lists in your output. **Tab stops** Design and Production supports only a single tab stop setting. If your design contains multiple tab stops, only the first tab stop value is used. **Widow and orphan settings** Design and Production does not support separately specifying the settings for widow and orphan rows. If your design contains multiple paragraphs with different settings for widow rows and orphan rows, then the larger value is used as the setting for both widow and orphan rows.
Fonts	**Font options** Fonts that are used in your xPression design are added to the design database during the import. Character-level formatting is imported as styles in Exstream. Your output might contain some visual differences. **Special characters** Special characters are supported. If any characters are missing in your output, you might need to adjust your application settings to include additional character sets.
Tables	Most table settings are supported and converted into equivalent Design and Production settings. Keep the following considerations in mind: • If split and flow settings are not explicitly specified in the xDesign template, imported tables will be automatically set to split and flow in Design and Production. • If several content items in the xPression design contain tables with the same number of columns and the same column widths, the import process creates a new table in Design and Production that contains all of these tables as they would appear in xPression output. The new table is placed in the Automated Tables folder. • Rowspan attributes are not supported in Design and Production. Rowspan attributes in your tables are expanded to multiple cells in the imported design, and the content in the rowspan is placed in the first row.

Template style support, continued

Feature	Import behavior
Special features	**Color** Color settings are supported. **Date/Time functionality** Date/Time functionality is supported. The Date/Time field is imported as a Design and Production variable. **Hyperlinks** Hyperlinks are supported. **Interactive controls** Content from Microsoft Word legacy form fields in your xPression designs is imported, but the interactive properties are disabled in the imported designs. You can use the interactive functionality that is available in Exstream to add interactive features to your output. **Paragraph merging** Paragraph merging is not supported. Design and Production does not retain the **Mark Paragraph for Merge** setting on imported content items. If your design included paragraphs that were set to merge, they might appear as separate paragraphs when you generate output using the Exstream engine. **Other features** The following Microsoft Word features are not supported: • Cross-references • Document protection • Fields • Frames and text boxes • Indexes • Tables of content

2.2 xPression output support

You can import an output profile PDPX file into Design and Production to import your xPression output information. The import utility can convert most xPression output definition settings into equivalent Design and Production output objects. However, some output settings are not supported or might not be imported correctly. Before importing a PDPX file, OpenText recommends that you familiarize yourself with how your xPression output information appears after an import.

The following information describes the supported output definition settings and the equivalent Design and Production settings on the output object properties for each output format. For information about xPression document features, see "xPression document feature support" on page 9.

Note: PCL and text output format definitions are not imported into Design and Production.

Although, the PCL output format is supported in SBCS applications in Design and Production, any PCL output definitions in your output profile PDPX file will not be imported because the import utility imports xPression documents as DBCS applications. You can separately create PCL output objects in Design Manager. To do so, you must set the application mode for your database to **SBCS/DBCS** on the **Workflow** tab in **System Settings**.

2.2.1 AFP output

AFP output settings

xPression setting	Design and Production setting
Resample Images to Output Resolution	This setting corresponds to **Resource Management** tab > **Resolution method** list. If the xPression setting is selected, **Resolution method** is set to **Printer resolution**. If the xPression setting is not selected, **Resolution method** is set to **Automatic**.
Select a printer for the output	Not supported
Convert all images to black and white	This setting corresponds to the following settings: • **Basic** tab > **Color mode** list • **Resource Management** tab > **Image management** area > **Format** list If the xPression setting is not selected, the following settings are applied: • **Color mode** is set to **Full color** • **Format** is set to **JPEG** If the xPression setting is selected, then for both **Diffusion Dither** and **50% Threshold** settings, the following settings are applied: • **Color mode** is set to **Black and white** • **Format** is set to **IOCA B&W uncompressed**

AFP output settings, continued

xPression setting	Design and Production setting
Fonts > **TrueType to AFP Outline Font**	Not supported By default, on the **Resource Management** tab, the following settings are applied: • **Font usage** is set to **Reference all fonts** • **Font type** is set to **Outline**
Fonts > **TrueType to AFP Subset Font**	For both of the **Bitmap** formats, the following settings are applied: • **Font usage** is set to **Embed all fonts** • **Font type** is set to **Bitmap** The **Outline** format setting is not supported. Be default, on the **Resource Management** tab, the following settings are applied: • **Font usage** is set to **Reference all fonts** • **Font type** is set to **Outline**
Fonts > **Embedded TrueType**	On the **Resource Management** tab, the following settings are applied: • **Font usage** is set to **Embed all fonts** • **Font type** is set to **Bitmap**
Use Dither Pattern Shading	Not supported
RGB Black&White Threshold	Not supported
CMYK Black&White Threshold	Not supported
Do Not Embed Font List	Not supported
TLE Items	Not supported

2.2.2 HTML output

HTML output settings

xPression setting	Design and Production setting
Image Storage Path	Not supported

HTML output settings, continued

xPression setting	Design and Production setting
Image URL Root	This setting corresponds to **Resource Management** tab > **Image management** area > **Location path** box. If **Image URL Root** box contains any content, **Image path** is set to **Use a static value to define a default path**, and the content in the **Image URL Root** box is inserted into the **Location path** box.
Use Style Classes	Not supported

2.2.3 PDF output

PDF output format definitions are converted into PDF or PDF/A output objects, depending on the **PDF Type** setting in xAdmin.

PDF output settings

xPression setting	Design and Production setting
Resample Images to Output Resolution	This setting corresponds to **Resource Management** tab > **Resolution method** list. • If the xPression setting is selected, **Resolution method** is set to **Printer resolution**. • If the xPression setting is not selected, **Resolution method** is set to **Automatic**.
Select a printer for the output	Not supported

PDF Type	This setting corresponds to **Basic** tab > **Driver** list.
	The setting in the **Driver** list indicates the format of the output object. For all settings except **PDF/A**, the import utility creates a PDF output object in Design Manager. The created output object uses the default PDF settings.
	• For **PDF/A-1a** and **PDF/A-1b**, the import utility creates a PDF/A output object and sets the level to **Level 1a** or **Level 1b**.
	• For **PDF/UA**, the following additional settings are applied:
	  • On the **Basic** tab, **Accessibility standard** is set to **PDF/UA**
	  • On the **Resource Management** tab, **Overlay processing** is set to **Do not create overlays**.
	Keep in mind that **PDF/X-1a** and **PDF/X-1b** are not supported. The import utility defaults to creating a PDF output object for PDF/X output definitions.

ICC Profile (PDF/A only)	This setting corresponds to **Basic** tab > **PDF/A controls** area > **Output Condition Identifier** box. Depending on the value specified in xAdmin, **Output Condition Identifier** is set to **sRGB** or **CGATS TR 001**.

PDF output settings, continued

xPression setting	Design and Production setting
Page Scaling	Not supported
Compress PDF	This setting corresponds to **Basic** tab > **PDF controls** area > **Compress** check box. If the xPression setting is set to **True**, the **Compression level** is set to 6.
Fillable PDF	Not supported
Enable Bookmark	This setting corresponds to **Basic** tab > **PDF bookmarks** area. • If the xPression setting is selected, the **Documents** check box and the **Customers** check box are selected. • If the xPression setting is not selected, all of the check boxes in this area are cleared. The following xPression bookmark settings are not supported: • Transfer External Bookmark • Package Name • Document Name • Levels in Document
Enable Digital Signature	Not supported However, Design and Production does not provide the option to disable digital signatures for PDF output. If your design contained digital signatures, those signatures will be included in the output.

2.2.4 PostScript output

PostScript output settings

xPression setting	Design and Production setting
Resample Images to Output Resolution	**Resource Management** tab > **Resolution method** list • If the xPression setting is selected, **Resolution method** is set to **Printer resolution**. • If the xPression setting is not selected, **Resolution method** is set to **Automatic**.
Select a printer for the output	Not supported

2.2.5 PPML output

PPML output settings

xPression setting	Design and Production setting
Resample Images to Output Resolution	**Resource Management** tab > **Resolution method** list • If the xPression setting is selected, **Resolution method** is set to **Printer resolution**. • If the xPression setting is not selected, **Resolution method** is set to **Automatic**.
Select a printer for the output	Not supported
Page Scaling	Not supported
Compress PDF	Not supported
Do Not Embed Font List	Not supported

2.2.6 TIFF output

TIFF output format definitions are converted into TIFF output objects. However, there are no equivalent Design and Production settings that correspond to the xPression settings for TIFF output.

2.3 Importing PDPX files

You can use xPression PDPX files to export your designs and then import it into Design and Production. You can import the following types of PDPX files:

- **Document PDPX files**—Use xPression document PDPX files to import your xPression designs into Design and Production. Before you import, make sure that you have reviewed "xPression document feature support" on page 9.
- **Output profile PDPX files**—Use output profile PDPX files to import your output information and convert output format definitions into equivalent output objects in Design and Production. You can then use the output objects in Design Manager to create output queues for your application. Before you import, make sure that you have reviewed "xPression output support" on page 19

You can import PDPX files into Design and Production from the command prompt or from within Design Manager. If you use a batch file to import multiple PDPX files from the command prompt, keep in mind that you must invoke Design Manager separately for each file that you

want to import. Before you begin, make sure that you have reviewed the best practices for creating a PDPX file.

2.3.1 Best practices for creating PDPX files

Before you begin importing your xPression designs into Design and Production, you must export your designs into a PDPX file using xAdmin.

For an optimal import experience, use the following best practices when you create a PDPX file:

- **Review the approval state of your content items**— When you import an xPression document PDPX file, the workflow state of the imported content items depends on the workflow state that was applied in xDesign. Before you create the PDPX file, review the approval states of content items to determine whether they should be marked as **Approved** in xDesign, based on the following import behavior:
  - Content items that are in the **Approved** state in xDesign are imported with the **Approved** state applied in Design Manager.
  - Content items that are in the **Withdrawn** state are not imported.
  - Content items that are in any other workflow state are imported with the **Work in Progress** state applied.

  Additionally, for shared content items, only the latest major version of the content item is imported with the **Approved** state applied in Design Manager. All minor versions of a shared content item are imported with the **Work in Progress** state applied.
- **Determine the versions needed for your content items**—When you import an xPression document PDPX file, the choice that you make about the versions to import affects how content items are created in Design Manager. You have two options on the **Import xPression design** dialog box:
  - If you select **Show only latest version** from the **Content item versions** list, Exstream creates a design object for the latest version of the content items in the content group. Previous versions are available from the object's history.
  - If you select **Show all versions**from the **Content item versions** list, Exstream creates a separate design object for each version of the content items in the content group. You might need to re-order objects in the Design Manager Library to create the correct output.
- **Verify your data source format**—By default, the import utility maps the customer driver file as if it were an XML data source. If you are using a data source that is in a format other than XML, you must do one of the following:
  - Before you create the xPression document PDPX file, export your data to an XML file and specify the exported XML file as your data source; or
  - After the import process is complete, change the format of the customer driver file to

match your data source. If you do this, you must re-map your data file before you can package your application. For more information about defining the customer driver file format and mapping a data file, see *Using Data to Drive an Application* in the Exstream Design and Production documentation

- **Add subdocuments and master documents to the same document PDPX file**—In order for a subdocument to convert correctly, the master document and the subdocument must be in the same PDPX file. If multiple documents use the same subdocument, OpenText recommends that you include as many of the documents as possible in a single PDPX file to avoid duplication of objects.

  Keep in mind that if your PDPX file includes xDesign documents that use the CompuSet publishing engine, any custom CompuSet commands used in the content will be ignored during the import process.

- **Create a single document PDPX file to export documents that use the same Microsoft Word templates**—During the import process, Exstream creates style sheets to import Microsoft Word template styles from your xDesign documents. The import utility reads the `DocXML_STYLE` file that is contained in the PDPX file and creates a style sheet that corresponds to the template, and the styles that are contained in the template are added to the style sheet. Some settings, such as page type, are imported directly as Design Manager settings.

  If your PDPX file contains multiple xDesign documents that use different templates, the import utility will create a separate style sheet for each unique template, and assign styles to imported design objects based on the template that was used by the xDesign document. Keep in mind that if you import a PDPX file that contains a particular template, and then import another PDPX file that contains a newer version of that template, the import utility does not recognize that you are importing multiple versions of the same template.

  For example, if you import a PDPX file that contains template A with styles X and Y, the import utility creates a corresponding style sheet A and adds those styles to it. If you then import a different PDPX file with a newer version of template A that has updated styles X and Y, and a new style Z, then the import utility will not create a new style sheet or update the older styles, but it will add the new style to the existing style sheet A. However, if you import a single PDPX file with both the older and newer versions of template A, the import utility identifies the newest version of the template and imports the latest styles into Design Manager.

- **Include fonts in the document PDPX file**— The import utility automatically adds the font family, font size, and font style to the design database for any fonts that it encounters when importing content items. If you select the **Include fonts** option in xAdmin, the import utility places any fonts that are not installed on your system in a fonts directory, and you receive a message during the import process prompting you to manually install the missing fonts. OpenText strongly recommends that you install these fonts before you click OK to continue the import.

- **Include category information in the document PDPX file**—To include the information necessary for a successful import, you must select the **Category Info** option when you create the PDPX file in xAdmin.

2.3.2 Importing a PDPX file from the command prompt

1. Open a command window in the directory where you installed Design and Production.
2. At the command prompt, enter `DesignManager` followed by a space, and then enter the following switches as required, adding a space after each switch:

   - `-XPRESS_IMPORT=<fileName>`

     Specifies the file name of the PDPX file that you want to import.

   - `-EXSTREAMUSER=<userName>`

     Specifies a Design and Production user name.

     Important: You must specify a user with superuser privileges.

   - `-EXSTREAMPASSWORD=<password>`

     Specifies the password for the Design and Production user.

   - `-DSN=<dataSourceName>`

     Specifies the data source name of the design database that you want to use to store your design objects.

   - `-XPRESS_LOGFILE=<fileName>`

     (Optional) Specifies a custom file name and location for the log file that is generated during the import. If you do not use this switch, the system saves the log file in the same directory as the PDPX file that you are importing, and names it `<importFile>.pdpx.log.txt`, where `importFile` is the name of your PDPX file.

   - `-DBUSER=<databaseUserName>`

     (Optional) Specifies the user name of the database user that you are using to sign in to a password-protected database. This switch is required only if the design database requires you to sign in separately.

   - `-DBPASSWORD=<databasePassword>`

     (Optional) Specifies the password for the database user that you are using to sign in to a password-protected database. This switch is required only if the design database requires you to sign in separately.

   For example:

```
DesignManager -XPRESS_IMPORT=C:\xPressTest\AutomaticPaymentLetter.pdpx
-XPRESS_LOGFILE=C:\xPressTest\AutopayTest.log -EXSTREAMUSER=admin
```

```
-EXSTREAMPASSWORD=xxx -DSN=xPressTestDB
```

For more information about the Design and Production switches that are used with xPression imports, see "Switches for xPression imports" on page 40.

3. To run the command and begin the import process, press **ENTER**.

   Design Manager opens, and the **xPression Import Status** dialog box displays the status of your import. After the import is complete, review the import log and the report file for information about the design objects that were created during the import.

2.3.3 Importing a PDPX file in Design Manager

1. Open Design Manager, and sign in as a user with superuser privileges.
2. In Design Manager, from the Menu bar, select **Tools > Import xPression PDPX...**.
3. In the **Import xPression designs** dialog box, select the PDPX file that you want to import.
4. In the **Page Type** list, select one of the following options:
   - **Auto-detect Paper Type**—Exstream creates a paper type for the imported application based on the size of the first content piece that is encountered in the xPression design that you are importing. This option is selected by default.
   - **Use specified Paper Type**—Lets you specify a paper type object from the Library to use for the imported xPression application. In the **Paper Type** list, select the paper type that you want to use.
5. In the **Content item versions** list, select the content items versions that you want to import. The option that you select affects how content items are created in Design Manager. You can select one of the following options:
   - **Show only the latest version in the content group**—Exstream creates a design object for the latest version of each content item in a content group. Previous versions of each content item are available in the history for the imported object.
   - **Show all versions**—Exstream creates a separate design object for each version of a content item in a content group.
6. Click **Open**.

   The **xPression Import Status** dialog box displays the status of your import. After the import is complete, review the import log and the report file for information about the design objects that were created during the import.

2.4 Reviewing import information

During the import process, Design and Production generates an import log and a report file that contain information about your imported xPression design. OpenText recommends that you review these files carefully to make sure that your design behaves as expected after the import. After you have reviewed this information, modify your designs as required and run the engine to test your output.

2.4.1 Import log

The log file contains the informational and action messages that are generated during the import process when you import a PDPX file into Exstream Design and Production. The messages in the log file provide information about the Design Manager objects that are created during the import, and can help you identify and troubleshoot any potential design updates that you might need to make after the import is complete.

Messages in the log file are identified with a number and a severity level, such as `EX001800I`. You can use the message number that follows the leading `EX00` prefix to search the Message Dictionary for more information. The alphabet character refers to the type of message.

- Informational messages are denoted by the letter `I`. These messages are typically used to provide information that does not require you to take any further action.
- Action messages are denoted by the letter `A`. These messages include both warnings and error messages that were encountered during the import. Depending on the severity of the message, you might need to review and modify your imported design in Exstream Design and Production, or you might need to re-export your PDPX file in xAdmin and initiate a new import in Design Manager.

2.4.2 Report file

The report file contains a mapping report for the xPression design that you imported into Design and Production. It provides a textual record of the one-to-one relationship between each item in the document tree of the xDesign document that you are importing and the corresponding Library object that is created in Design Manager after the import.

For xDesign items that were converted into an equivalent Design Manager library object, the name of the new object is recorded in the report file. For xDesign items that do not have an Exstream equivalent, the report file contains information about any substitutions that were made to convert the xDesign item during the import.

The report file is a useful tool for identifying the objects that make up your new Design and Production application. Use the information in the report file to understand the correlation between your xDesign document tree and the corresponding Design Manager library structure.

2.5 Reviewing imported designs

After the import process is complete and you have reviewed the messages that are generated during the import, you might need to review and modify some aspects of the imported designs. Some design changes are required before you can produce output; if the generated output does not behave as expected, additional changes might be required to adjust your design.

- "Reviewing variable formatting in Design Manager" below
- "Modifying designs before producing output" on page 32
- "Reviewing output for potential design changes" on page 35

2.5.1 Reviewing variable formatting in Design Manager

xDesign variables are imported from xPression designs into Design and Production during the import process. However, some formatting options in xDesign do not exactly match the formatting options in Design and Production.

The following xDesign variable formatting options are not supported in Design and Production:

- `S` (millisecond)
- `F` (day of week in month)
- `W` (week in month)
- `z` (time zone)
- `G` (era)

The following xDesign variable formatting options are supported and the import utility maps them to one of the corresponding options in Design and Production:

xDesign format	Exstream format	Visual difference
`M/d/yyyy`	`m/d/yyyy`	None
`MM/dd/yy`	`mm/dd/yy`	None

xDesign format	Exstream format	Visual difference
`MM/dd/yyyy`	`mm/dd/yyyy`	None
`MMMM d, yyyy`	`MMM d, yyyy`	None
`M/d/yy H:mm`	`m/d/yy hh:nn`	Time format 0:00 in xDesign is represented as 12:00 in Design and Production
`H:mm`	`hh:nn`	Time format 0:00 in xDesign is represented as 12:00 in Design and Production
`H:mm:ss`	`hh:nn:ss`	Time format 0:00 in xDesign is represented as 12:00 in Design and Production
`y`	`y`	None
`M`	`m`	None
`d`	`d`	None
`h`	`h`	None This formatting option is driven by the locale.
`H`	`h`	Hour format 0 to 23 in xDesign is represented as 1 to 12 in Design and Production. This formatting option is driven by the locale.
`m`	`n`	None
`s`	`s`	None
`E`	`ddd` or `dddd`	None
`D`	`y`	None
`W`	`W`	None
`a`	`a`	None
`k`	`k`	Hour format 0 to 23 in xDesign is represented as 1 to 12 in Design and Production. This formatting option is driven by the locale.
`K`	`K`	Hour format 0 to 23 in xDesign is represented as 1 to 12 in Design and Production. This formatting option is driven by the locale.

xDesign format	Exstream format	Visual difference
' (single quote)	" (double quote)	None
' ' (two single quotes)	\ '	None
All other options	All other options	Variables that use these formatting options might not appear in the output.

2.5.2 Modifying designs before producing output

Before you run the Exstream engine to produce output from your imported designs, you might need to modify your designs in Design Manager or Designer so that your output behaves as expected. In addition to the action messages in the import log, OpenText strongly recommends that you review and consider at least the following potential changes to your designs:

1. **Modify your design to include design objects that were created during the import process.**
   - Incorporate the imported header and footer paragraph items in your design.
   - If your design used shared content, you will need to restructure your applications in Design Manager to use the imported shared content. Each instance of a shared content item in the original design is imported as a separate object in the Library. If you want to reuse an imported object in multiple imported applications, you must add references to that object to those applications.
   - If several content items in the xPression design contain tables with the same number of columns and the same column widths, the import process creates a new automated table in Design and Production that contains all of these tables as they would appear in xPression output. The new table is placed in the **Automated Tables** folder in the Library; however, it is not included in the imported design by default. To streamline your design, you can use this optimized table and discard the individual content items.

   For more information about modifying your design, see *Designing Customer Communications* in the Exstream Design and Production documentation.
2. **Review variable content and business logic in the imported application.**
   - If your design contained user exits, the import utility uses the built-in dynamic data access functionality in Design and Production to import those user exits and configure your application. The import utility creates the following items during the import process:
     - Separate functions for each user exit definition that is contained in your PDPX file. These functions are placed in the `UserExits` folder in the Design Manager library.
     - A connector object called UEConnector in the Design Manager library in the `UserExits > Connectors` folder for your imported application. The connector

object references the Java Enabler and the INI file that was created during the import process.

- An INI file that contains the fully qualified path names for the user exit connector that is included in the Design and Production environment. This INI file is created in the same directory as the PDPX file that you are importing.
- Sample JSON-formatted auxiliary and reference data files that are used by Design Manager to visualize the layouts of those data files. These JSON data files are created in the same directory as the PDPX file that you are importing.

To build output that correctly includes information that depends on these user exits, you must complete the following steps:

i. Update the INI file to include the fully qualified path names for all of the JAR files that contain your user exits. By default, the INI file that is created during the import contains a CLASSPATH parameter that references the required UserExitConnector.jar file, and a sample dummy user exit JAR file. You must remove this sample path and replace it with the correct paths for your user exit JAR files.

ii. Check the settings in the UEConnector object that is created in the Connectors folder within your application folder in the Library. . Make sure that the **Open parameters** box in the connector properties contains the path to the INI file.

Note: Keep the following considerations in mind:

- The UEConnector and the associated INI file that are created by the import utility are usable only in the design environment. For a production environment, you must create a new INI file that contains the path to the UserExitConnector JAR file, and create a separate connector object that references this new INI file.
- If you are using the Solaris platform, you must make sure that your user exit JAR file is created using Java version 7 or earlier.
- The Java file (JVM.dll) must be present in the system path.

For more information about configuring user-written dynamic data access routines in Design and Production, see *Configuring Connectors* in the Exstream Design and Production documentation.

- If you assigned jurisdictions or effective dates to objects in your design, validate that the jurisdiction and effective date variables are mapped correctly in the driver file, or that the related system variable values ('SYS_CustomerJurisdiction' and 'SYS_CustomerEffectiveDate') are set correctly. You must also make sure that the correct jurisdiction variable is selected in the **Customer ID for regulatory** box on the **Basic**

tab of the application properties.

For information about using effective dates and jurisdictions, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- If your design contained interlaced (non-nested) GoTo rules or data read rules, you might need to redesign your application to replicate the expected behavior.

  For more information about updating rules, formulas, and functions, see *Using Logic to Drive an Application* in the Exstream Design and Production documentation.

- If the expected value of an xDesign variable is a null value or a predefined default value, review the corresponding variable in Design Manager to see the value that was assigned to that variable during the import. The import utility creates null variables to represent null values, and assigns an initial value to these variables. For example, null string variables are assigned an initial value of `_NULL_`. If a rule in your design contains a comparison to a null value, the import utility replaces the null value with the auto-generated null variable. If required, specify new variable values to make sure that the assigned initial values do not appear in customer data.

  For more information about assigning initial values to variables, see *Using Data to Drive an Application* in the Exstream Design and Production documentation.

3. **Set up your output objects and output queues.**

   The import utility creates a PDF output queue for the imported application. If you want to produce output in a different format, you must set up additional output objects and output queues in Design Manager.

   For more information about setting up your output options, see *Creating Output* in the Exstream Design and Production documentation.

4. **Add any missing functionality.**

   - If your design contained charts, consider redesigning your custom charts, using the charts functionality that is available in Design and Production.

     For more information about adding charts to your designs, see *Designing Customer Communications* in the Exstream Design and Production documentation.

   - If your design contained interactive controls, enable interactive capabilities for your application by using Exstream interactive functionality to design Exstream Live or Exstream Empower applications.

     For more information about creating Exstream Live applications, see *Designing for LiveEditor* in the Exstream Design and Production documentation.

     For more information about creating Exstream Empower applications, see *Designing for Exstream Empower Editor* in the Exstream Design and Production documentation.

2.5.3 Reviewing output for potential design changes

After you run the Exstream engine to produce output from your imported designs, you might need to make the additional changes to your designs:

1. **Review pagination in the generated output.**

   - Design and Production does not support separately specifying the settings for widow and orphan rows in tables or paragraphs. If your design contains multiple items with different settings for widow rows and orphan rows, the larger value is used as the setting for both widow and orphan rows. If your design contains embedded items, the largest widow or orphan setting is applied to the parent object. Review the output and, if required, modify the widow and orphan settings on your objects.
   - Design and Production does not retain the **Mark Paragraph for Merge** setting on imported content items. If your design included paragraphs that were set to merge, they might appear as separate paragraphs when you generate output using the Exstream engine. You can modify this behavior by manually merging paragraph items.

2. **Review the appearance of special characters in the generated output.**

   If your design contains any missing special characters, you might need to include additional character sets in your application. In Design Manager, in the application properties, click the **Font Resources** tab, and then use the **Additional font character ranges to package** selection area to add any required character sets.

3. **Review the appearance of tables in the generated output.**

   - If your tables contained rowspan attributes, those rowspans are expanded to multiple cells in the imported design, and the content in the rowspan is placed in the first row. Review the tables in your output, and if required, consider using embedded tables in Designer to achieve the expected appearance.
   - If you use the auto-generated tables that were created by the import utility in the **Automated Tables** folder, review your output to make sure that the tables appear correctly in your output.

4. **Verify that the correct version of your content items appears in the generated output.**

   If your xPression design contained multiple versions of a content item, the import process creates a corresponding paragraph object for each version of the item in a section. When you generate output, only one of the paragraph objects is included; however, the version that is included might not be the latest version of that paragraph. Review the generated output and, if required, reorder the paragraph objects so that the first object that meets all of the rules and the jurisdiction conditions appears in the output.

5. **Verify that the correct variable values appear in the generated output.**

   If your design contains variables values that are calculated and inserted in the output multiple times for each customer, or if your design includes subdocuments, the variable timing in Design and Production might not match the variable timing in xPression because of the way data is processed by the Exstream engine. Review your output to make sure that your output contains the correct variable values, and redesign your application if required.

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

2.6 Building output from imported designs

After you import your xPression designs into Design and Production, OpenText recommends that you run the Exstream design engine to review any differences in your output.

When you run the engine, you can use a control file to specify engine switches that define how the engine processes your application. To provide a convenient way to test your designs, Design and Production generates a sample control file during the import process. The sample control file is an OPT file that is always generated in the directory that contains your PDPX file, and that uses the same name as your PDPX file with the `*.opt` extension. OpenText strongly recommends that you use the sample control file when you run the engine for the first time.

For more information about additional options for testing your applications, see *Preparing Applications for Production* in the Exstream Design and Production documentation.

To generate output, you must specify the data source for your application, package it, and then run the engine to produce output.

2.6.1 Specifying the data source

The import utility reads the data source mappings that exist in the PDPX file and attempts to find the data source in the same directory as the PDPX file, or in the original location of the data source as recorded in the PDPX file. Before you can package the application, you must make sure that the data source for your imported xPression design is present, and that it is correctly specified in Design Manager.

To specify the data mapping source in Design Manager:

1. In the Library, expand the imported xPression design folder.
2. In the design folder, go to **DataSources > Data Files > DBCS Data Files**, and open the data source object for your application in the Property Panel.

3. On the **Basic** tab, in the **Data mapping source** box, enter the file path to your data source file.

   > **Tip:** If your PDPX file referenced a non-XML data source, you must change the format of the driver file in Design Manager.
   >
   > In the **File format** list, change the selection from **XML** to the format that you want to use. If you do this, you must re-map your data file before you can package your application.
   >
   > For more information about defining the customer driver file format and mapping a data file, see *Using Data to Drive an Application* in the Exstream Design and Production documentation

4. On the **Test Data Source** tab, in the **File to use in production** box, enter the file path to your data source file.

2.6.2 Creating output queues

Before you can package the application, you must also specify the output queue that you want to use for packaging. You can select an output queue using one of the following options:

- **Use the default PDF output queue**—The import utility creates a default PDF output queue for each imported application during the import process.
- **Create output queues based on imported output objects**—You can use output profile PDPX files to import your output definitions from xPression and convert them into equivalent output objects in Design Manager. You can then use the output objects to create output queues for your application. Before you import an output profile PDPX file, make sure that you have reviewed "xPression output support" on page 19.

For more information about creating output and output queue objects, see *Creating Output* in the Exstream Design and Production documentation.

2.6.3 Packaging the application

Before you can run the engine, you must first package the imported application. You can package your application from the command prompt or from within Design Manager. After packaging is complete, run the engine to produce output.

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

The following table describes the steps to package an application:

Packaging an application in Design and Production

To	Do this
Package an application from the command prompt	1. Open a command window in the directory where you installed Design and Production. 2. At the command prompt, enter the following command: `Packager -APPLICATION=<applicationName> -EXSTREAMUSER=<userName> -EXSTREAMPASSWORD=<password> -DSN=<database>` where `<applicationName>` specifies the name of the application that you want to package, as it appears in the Design Manager Library. If the application name has spaces, you must enclose it in double quotation marks or the name is truncated. You can also add optional packaging switches to this command based on your requirements. 3. To run the command and package your application, press **ENTER**.
Package an application in Design Manager	1. In Design Manager, from the Menu bar, select **Tools > Package...**. Alternatively, you can right-click your application in the Library and select **Package...**. 2. On the **Build Package** dialog box, specify the packaging options for your application. **Tip:** If you want to package your application and then run the engine immediately after packaging, select the **Run Engine when complete** check box. You can then specify the engine run options in the **Run the Engine** dialog box. 3. In the **Package File** box, enter the path and name of the package file that you want to create. This must match the path and name of the package file that is specified in the sample control file. By default, in the sample control file, your PDPX file path and the name of the PDPX file are used as the name of the package file. That is, `<importFilePath>/<importFile>.pkg`, where `importFilePath` is the file path of the PDPX file that you imported, and `importFile` is the name of that PDPX file. 4. Select **Create for DBCS output queue device(s)** to use the output queues that are defined in your application. 5. Click **OK** to begin packaging. The progress bar on the **Build Production Package File** dialog box indicates the status of your package file.

2.6.4 Running the engine to produce output

Use the design engine to produce output from your imported application. Running the design engine lets you test your application in the design environment using all of the same options that you would use for a production engine run, with the added advantage of being able to access comprehensive testing tools.

You can run the design engine from the command prompt or from within Design Manager. When the engine run is complete, review the message file and the generated output for any design changes that you might need to make.

The following table describes the steps to produce output:

Producing output in Design and Production

To	Do this
Produce output from the command prompt	1. Open a command window in the directory where you installed Design and Production. 2. At the command prompt, enter the following command: `Engine_DBCS -CONTROLFILEENCODING=UTF16LE -CONTROLFILE=<sampleControlFile>` where `<sampleControlFile>` specifies the path and name of the sample control file. 3. To run the command and generate output, press **ENTER**.
Produce output from Design Manager	1. In Design Manager, from the Menu bar, select **Tools > Run Engine...**. 2. In the **Run the Engine** dialog box, update the following settings: • Select the **Control file** check box, and then click  to browse for the sample control file that is generated during the import process. • Select the **Package file** check box and enter the fully qualified file name of the package file that you want to create. This must match the path and name of the package file that is specified in the sample control file. 3. Click **Run** to initiate an engine run.

2.7 Switches for xPression imports

When you use Design Manager to import an xPression design, or when you run the Exstream engine to build output from an imported xPression application, you can use Design and Production switches to specify import utility or engine processing behavior.

You must begin your switch command with the name of the program for which you want to use the switches. Enter the program name followed by a space, and then enter the switches as required, adding a space after each switch. Switches always begin with a hyphen (-) when they appear in a command.

Use the following syntax for a switch command:

```
<programName> -<switch> -<switch> ... -<switch>
```

where,

- `<programName>` specifies the program that you want to use.
  - To use the import utility switches, use `DesignManager` as the program name.
  - To use engine switches when you produce output, use `Engine_dbcs` if you are using the design engine, or use `ProdEngine_dbcs` if you are using the production engine.
- `<switch>` indicates a switch that you are using with the program, including any applicable arguments

For more information about using Design and Production switches, and for a comprehensive list of available switches, see *Switch Reference* in the Exstream Design and Production documentation.

2.7.1 Import utility switches

When you import xPression designs from the command prompt, use the following switches to provide information to the import utility in Design Manager.

- XPRESS_IMPORT, to specify the PDPX file that you want to import
- XPRESS_LOGFILE, to specify the name of the import log file
- EXSTREAMUSER, to specify the Design and Production user
- EXSTREAMPASSWORD, to specify the password for the Design and Production user
- DSN, to specify the data source name for your design database
- DBSCHEMA, to specify the schema for your design database (if the database requires a schema)

- DBUSER, to specify the user for your design database (if the database requires you to sign in separately)
- DBPASSWORD, to specify the password for your database user (if the design database requires you to sign in separately)

2.7.2 Engine switches

Engine switches define how the engine processes your application. Depending on your application, you might require several engine switches to produce the desired output. When you run the Exstream engine to produce output from an imported xPression design, OpenText recommends that you use at least the following switches in your control file:

Note: The sample control file generated during the import process contains all of the recommended switches.

- DO_NOT_DEFAULT_REF_VARS, to prevent variables from being reset to their default values before they are read from a lookup file
- FILEMAP, to specify a valid file name for a symbolic file name that is specified in your data file properties, such as for a driver file or an output file
- PRECOMPOSE_PARAS_FOR_NONDATA_DOCS, to compose any data-driven content that appears in documents that are not driven by data files, before composing those documents
- PRESERVE_SECTION_DESIGN_ORDER, to add data-driven sections to the output in the same order in which they appear in the application design
- RETAIN_VARIABLE_RESET_TIME, to change the engine timing of an array variable that is used in a reference data file
- TRY_ALL_IMAGE_PLACEHOLDER_TYPES, to include a mismatched external image file type that is referenced by a placeholder variable
- MESSAGEFILE, to specify the location of the message file that is generated during the engine run
- MSGRESOURCE, to specify the location of the message resource file to use during the engine run
- PACKAGEFILE, to specify the name of the package file to create

2.8 Understanding Exstream terminology

As you work with your imported xPression designs in Design Manager and Designer, it can be useful to understand how Exstream terminology relates to the objects created by the import utility.

The following table describes the usage and definition of Exstream terms that are commonly used in an xPression import setting:

Exsteam term	Usage and definition
Application	A parent object for all of the design objects, logic resources, and settings that references your data repository and delivery channels to produce personalized documents for your customers. Each xPression document and subdocument is imported into Design Manager as a separate application that you can see in the Library under the "Applications" heading.
Application object	Library objects that can be referenced in an application. The contents of an xPression document, such as content items and rules, are imported as application objects, which include documents, pages, sections, paragraphs, output queues, and variables.
Approval state	User-defined steps that an object must go through before being approved. The workflow states of imported xPression content items are mapped to equivalent workflow states in Design Manager.
Array variable	A variable that contains multiple values. xPression array variables are converted into Design and Production array variables.
Child object	A lower-level object that is placed in a higher-level (or parent) object. The child object is usually necessary for the parent object to work correctly.
Component	A design object, such as an address block or company logo, that you create in Designer and add to the Library for reuse in multiple designs. Component objects are stored in the Library under the "Components" heading.
Connector	An object that provides information about the location and operating parameters of external user-written routines (DLLs) for use with Dynamic Data Access (DDA). Connector objects are stored in the Library under the "Connectors" heading. Alternatively, a connector can also refer to a data interface that allows the engine to access enterprise systems for real-time data access.

Exsteam term	Usage and definition
Control file	A reusable format text file, created outside of the design environment, that lets you control the behavior of the engine by specifying required and optional engine switches in a centralized location. The format in which you write a control file varies depending on the engine platform that you use. Any engine switch can be placed in a control file. The import utility generates a sample control file when you import a PDPX file. A list of recommended engine switches for use with an imported xPression application is included in that sample control file.
Customer driver file	A data file that contains customer data and is required to run an application. By default, the import utility maps the customer driver file that it creates as if it were an XML data source.
Data file	A Library object that contains the information required for the engine to locate, read, write, and use data from an external source within an application. There are multiple types of data files. At least one customer driver file (a specific type of data file) is required for each application to provide information about the recipients of the application. Data files are stored in the Library under the "Data Files" heading. When you import an xPression document, Design and Production creates an XML-formatted driver file object that points to the data source file that is referenced in the PDPX file.
Data mapping	The process of assigning areas of a data file to one or more variables. When the engine runs, it reads, writes, or updates data in data areas according to the properties of the variables. If your PDPX file includes an XML data source, that data source is used as the mapping and test data source in the driver file in Design Manager.
Design database	The database that stores all design information. This includes the data for all the objects in the Library as well as System Settings for user access, display preferences, and feature settings.
Design environment	A local workstation where application development and testing occurs. The design environment can also mean the collection of programs that you use to design and test applications. The Exstream design environment consists of a design interface and design database. The design interface allows users to create design components and add personalization to customer communications. Exstream can also be configured so that different users can manage different aspects of the design process (for example, formatting, versioning, or workflow controls). The design database allows users to store the design objects and their settings.
Design object	An object that is used to customize a design. Examples of a design object include barcodes, buttons, charts, check boxes, images, message frames, radio buttons, shapes, signature buttons and fields, tables, and text boxes.
Design time	The period of time when a user is actively creating customer content in Exstream. If you are going to be importing external content, doing so at design time can be useful for content that rarely changes or needs to be formatted after it's been imported.
Document	A Library object that contains all the objects and settings required for the design of a personalized customer communication. A document must contain at least one page. An application must contain at least one document object. Document objects are stored in the Library under the "Documents" heading.
Dynamic	Indicates that an object can be changed or personalized for each customer. Dynamic content is updated at engine run time.

Exsteam term	Usage and definition
Effectivity	A feature that lets you control the content that is included or excluded from documents based on a range of dates. For example, you can use effectivity in a customer account statement to include content based on federal regulations that take effect on a specific date. This feature is analogous to the Effective Date feature in xPression.
Embedded object	An object that has been inserted in a table cell or text box. You can specify that embedded objects be inline with, or linked to, text. These objects can grow and move dynamically with the anchoring text.
Engine	The Exstream program that produces customer documents from the application package file. There are two engines—the design engine, which produces sample output from the design environment, and the production engine, which produces production-ready output from the production environment.
Engine switch	An instruction that is used to control engine processing from the command prompt or included within a control file.
Flow	The process in which an object continues onto a previously specified overflow area, usually a flow frame.
Formula	Logic that you can create in a formula variable to calculate new values based on available customer data.
Formula variable	A type of variable that holds the place for dynamic content when the value of the variable is based on logic using customer data.
Function	An object with logic you create that manipulates data and returns a value. Function objects are stored in the Library under the "Functions" heading.
Java enabler	A program that enables the engine to execute user-written Java code for custom data pre-processing, post-processing, and retrieval. The Java enabler is required to process xPression user exits.
Jurisdiction	A feature that lets you control the content that is included or excluded from documents based on customer locations. For example, you can use jurisdictions in a customer account statement to include legal notices that apply only to certain areas in which the business operates. This feature is analogous to the Jurisdiction feature in xPression.
Key	A file that contains an alphanumeric string that enables Exstream features based on the modules purchased.
Key variables	One or more variables in the customer driver file that notify the engine to read a reference data file.
Language	An environment object in Design Manager that lets you define dictionaries and formatting for numbers, months of the year, and days of the week for each language. You set up a language object for each language that you support, and link each one to a locale. Languages are stored in the Library under **Environment > System > Languages**.

Exsteam term	Usage and definition
Language layer	A virtual layer that lets you create content for multiple languages within a single message, page, or paragraph object. The engine determines which layer to include in the output for each customer based on customer variable data and logic.
Library	A panel of the Design Manager interface that is used to organize and list objects in the design database.
Library function	Special pieces of code that you can create to cause specific actions based on customer data (for example, including an object based on a customer location) or to analyze data and return findings (for example, calculating the total of the values of a series of variables). Library functions can be applied to any design object in an application on which you can place rules or formula variables.
Library rule	A logic rule (or usage rule) that is saved in the Library for reuse. Rules that are placed only on a specific object are called unnamed rules.
Local engine	The Exstream Design and Production program that produces sample output, which lets you verify your design on a Windows workstation. This engine is also called the design engine.
Locale	An environment object that contains the language dictionary and data and currency formats for a geographic location. Locales are used for formatting localized content when designs include content for multiple languages. Locales are stored in the Library under **Environment > System > Locales**.
Mapping	The act of assigning a variable to specific types of information contained in the customer data.
Marketing message	A communication that can be placed in a frame according to available space. Marketing messages are often date-sensitive, locale-sensitive, or cost-contained.
Message	A Library object that contains all the design, setup, and targeting controls for an informational or marketing message intended for placement in a larger customer communication. It can also represent pre-printed materials inserted at print time. Message objects are stored in the Library under the "Messages" heading.
Metadata	Data [information] that provides information about other data. Metadata consists of user-defined tags that you can add to an object to enable more efficient searches in the design database. You can also add metadata to your content to facilitate communications with downstream business processes in your organization.
Module	An optional component of Exstream Design and Production that enables additional function in the software.
Object	Any item that you create in Exstream.
Object status	An object's current status (such as "Work in Progress" or "Approved"), which is automatically assigned to each object as it advances through the various stages of the design workflow. The object's status is indicated by an icon to the left of the object in the Library. For design objects created by the import utility, the status depends on the approval state of the object before import, whether it is a shared object or not, ,and whether you selected to import all version of the object or the latest version only.

Exsteam term	Usage and definition
Output	An environment object that you use to set up the format of the final output, which can be print or electronic. The output object settings control the print stream that is sent to a device for printed output or the viewing software for electronic output. For example, you can configure resolution, simplex or duplex options, paper selection, color settings, printable areas for page margins, and so on. Outputs are stored in the Library under **Environment > Delivery > Outputs**. Additionally, output can also refer to electronic, print, or multi-channel results of an engine run that represent a preview of the customer communication in the design environment or produce the actual customer communication in the production environment. The import utility can convert most xPression output definition settings into equivalent Design and Production output objects.
Output file	A file that is created when the engine runs in production mode.
Output queue	A Library object that contains the objects and settings used for high-volume document production. The most common type of object included is the output driver (for example, PDF or HTML), but it can also include inserters, multiple-ups, and output breaks. Output queues are stored in the Library under **Environment > Delivery > Output Queues**.
Package	The action a user performs to gather all objects needed for an Exstream application in one file. This is the preliminary step to running the engine for design testing or production.
Package file	An Exstream file (*.pub) that contains all of the design and environment objects necessary for the engine to produce output.
Packaging switch	An instruction used from the command prompt or included within a control file to control application packaging.
Page	A Library object that generally corresponds to a sheet of paper. Pages contain design objects such as text and images, and can contain static content and dynamic content that can change based on customer data. One or more page objects make up a document. Page objects are stored in the Library under the "Pages" heading.
Page template	An environment object that defines a paper type and controls what type of page can be used. It also provides control over the page so that designers cannot change or delete specific content. Page templates are stored in the Library under **Environment > Design > Page Templates**.
Paper type	An environment object that defines the size, weight, and color of paper stock for printed output, or that defines the screen properties for electronic output. Paper types are stored in the Library under **Environment > Design > Paper Types**.
Paragraph	A Library object that contains a block of communication. A paragraph object might contain one or more text paragraphs. Paragraph objects are stored in the Library under the "Paragraphs" heading. Content items in xPression documents are imported as paragraph objects.
Parent object	A Library object that is built using subordinate objects, such as documents, which can contain pages, sections and paragraphs, and message objects.
Placeholder	An object used to reserve an area where external content will be imported at run time.
Placeholder variable	A variable that reserves a place for external content the engine imports during an engine run.

Exsteam term	Usage and definition
Production environment	A mainframe or production server where actual customer data is provided and distribution-ready output is produced. The production environment processes the documents for delivery after the application has been developed, tested, and approved in the design environment.
Production key	An alphanumeric string that enables access to both the design and production environments of Exstream.
Rule	A set of conditions that are used to control the inclusion or exclusion of a design object in the output in order to personalize documents for each customer. Rule objects are stored in the Library under the "Rules" heading. Rules in Exstream are analogous to Criteria in xPression.
Run time	The point when an engine run is initiated. Any referenced content designated for run-time import is placed into the output at this time.
Section	A Library object that contains paragraphs and, optionally, other section objects. It is used to create a hierarchy of content similar to headings and subheadings in a book. Section objects are stored in the Library under the "Sections" heading. Rules in xPression documents are imported as sections in Design Manager.
Style	A Library object that represents a specific character or paragraph format in a style sheet. Styles are stored in the Library under **Environment > Design > Styles**. Microsoft Word template styles in xPression documents are imported as style objects in Design Manager.
Style sheet	A Library object that contains a collection of styles and defines the character or paragraph formatting, such as font, size, color, indentation, or spacing. You can use style sheets to enforce font usage for branding purposes or to set paragraph spacing for optimal appearance of text in a design. Style sheets are stored in the Library under **Environment > Design > Style Sheets**.
Switch	Switches are instructions that can be used to run the production engine, customize database administration, or specify packaging options. Switches can be used within a control file or from the command prompt.
System variable	A pre-defined variable that is standard with Exstream software and is identified by the prefix SYS_.
Table	A design object that lets you present data in rows and columns. Exstream lets you create multiple types of tables.
Targeted flow	The placement of overflow into specific flow frames. It uses flow targets, which are stored in the Library under **Environment > Design > Flow Targets**.
Template	A page or message object that can be preset and used as a model for others to copy. Defining characteristics and setting objects within a template lets you create consistent pages and enforce style sheets. Templates are stored in the Library under **Environment > Design > Templates**.
Text message	A message object that typically contains only text, but other design objects such as tables, variables, and even graphics can still be added using Designer. Though you can use one, you are not required to use a template or a planned layout to create text messages. This versatility allows text messages to fill excess space on a page.

Exsteam term	Usage and definition
User-defined variable	A variable created by an Exstream user as needed. Unlike system variables, they do not come with the software and they can be moved, deleted, and renamed.
Variable	A Library object that represents data that changes at engine run time from sources such as customer data, current date and time, pages in a document, or a calculated value. Variables can make communication personalized for each customer. Variables are stored in the Library under the "Data Dictionary" heading. xPression variables are converted into equivalent Design and Production variables.
Workflow	A collaborative development process wherein objects progress through different stages of development. The workflow states of imported xPression content items are mapped to equivalent workflow states in Design Manager.

Chapter 3: Importing InDesign designs

The InDesign Converter lets you import InDesign files into the Design and Production environment. The InDesign Converter analyzes a specified InDesign file and creates a DXF file, which you can then import into Design and Production.

This section discusses the InDesign Converter version 1.0.

Most InDesign design objects are supported in the Exstream Design and Production environment, and when they are converted, become editable objects. For example, text alignment settings on a page in InDesign are supported fully. Therefore, when a page is converted and imported into Exstream Design and Production, you can see the alignment of text and change it just as you would for pages created in Exstream Design and Production.

Some features used in InDesign files are not supported as Design and Production objects. Designs that use unsupported features can be converted as view-only bitmaps. For example, polygon frames are not supported by the InDesign Converter. Therefore, during the conversion, you can select to convert the frame as a bitmap. When the converted page is imported into the design environment, the polygon frame will be visible, but you will be able to interact with it only as an image. For example, you can change its location, but not its appearance. If you choose not to convert unsupported objects to bitmap format, those objects do not appear on the converted page. You can also choose to convert entire pages to bitmap format. Features that cannot be converted to bitmap do not appear on the converted page.

Before using the InDesign Converter, make sure your system meets the following requirements:

- Windows 7 or later
- InDesign CS2 version 4.02 or later

If you install the InDesign Converter on a Mac, it must use OS X 10.3 or later.

This section discusses the following topics:

3.1 Supported InDesign features

This section describes the InDesign features supported by the Exstream Design and Production environment. If your design uses a feature not listed here, you can use one of the following solutions to make the conversion process simpler:

- Adjust the design so it uses only supported features.
- Convert the object or page as a bitmap. The object or feature will not be editable in the Design and Production environment.

Before converting an InDesign file, make sure you familiarize yourself with the following information about how objects in the InDesign file will appear after conversion. The following tables list InDesign features and objects that are supported by the converter. The tables are organized by the following functions:

- Color types
- Frame features
- Line features
- Shape features
- Text features
- Miscellaneous objects

Features that are identified as "fully supported" are imported as editable objects in the Design and Production environment.

Supported color types

Feature	Notes about conversion behavior
ANPA colors	Converted to RGB colors
DIC colors	Converted to RGB colors
Focoltone colors	Converted to RGB colors
Gradients	Appear on the page only if you convert the entire page as bitmap
KHS colors	Converted to RGB colors
Lab colors	Converted to RGB colors
Mixed-Ink colors	Converted to RGB colors

Supported color types, continued

Feature	Notes about conversion behavior
"None" color	Appears on the page only if you convert the entire page as bitmap
Pantone colors	Converted to CMYK colors
System colors (Macintosh and Windows)	Converted to RGB colors
TOYO colors	Converted to RGB colors
TRUMATCH colors	Converted to RGB colors
Unnamed colors	Appear on the page only if you convert the entire page as bitmap
Web colors	Converted to RGB colors

Supported frame features

Feature	Notes about conversion behavior
Anchored frame	Supported if you select to convert the feature as bitmap (by selecting the **Anchored frames** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Angle (Rotation)	Converted as bitmap
Bevel join	Appears on the converted page only if you convert the entire page as bitmap
Compound path	Appears on the converted page only if you convert the entire page as bitmap
Ellipse frames	Fully supported
Fill	Fully supported
Fill tint	Fully supported
Miter join	Fully supported
Polygon frames	Supported if you select to convert the feature as bitmap (by selecting the **Polygon** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Rectangle frames	Fully supported
Round join	Appears on the page only if you convert the entire page as bitmap
Sheared frames	Appears on the page only if you convert the entire page as bitmap
Stroke	Fully supported

Supported frame features, continued

Feature	Notes about conversion behavior
Stroke gap colors	Supported if you select to convert the feature as bitmap (by selecting the **Stroke gap color** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Stroke style	Supported if you select to convert the feature as bitmap (by selecting the **Stroke style** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Stroke tint	Fully supported
Stroke weight	Fully supported

Supported line features

Feature	Notes about conversion behavior
Begin and end styles	The InDesign Converter converts line begin and end styles into the closest matching begin and end styles. The converter creates the begin and end styles from the interior of the length of the line, but Designer creates begin and end styles from the exterior length of a line. Therefore, you might notice some differences in the sizes of lines that use begin and end styles.
Line tool	Fully supported
Pencil tool	Fully supported
Pen tool lines	Converted as bitmap
Stroke	Fully supported
Stroke tint	Fully supported
Stroke type	Converted as bitmap
Stroke weight	Fully supported

Supported shape features

Feature	Notes about conversion behavior
Ellipses (circles)	Fully supported
Pen tool shapes	This object can be converted as a bitmap, or you can apply an open path to the object in InDesign so the object's shape and fill can be edited in Designer. If you apply an open path to it, the object might not be visible when you open the imported page in Designer. You can select the object in the Outline Viewer to locate it and restore its shape and fill.
Pathfinder shapes	The object will appear on the converted page if you convert the entire page as bitmap, or you can apply an open path in InDesign to retain the object's shape and fill during conversion. When you open the imported page in Designer, the object might not be visible. You can select the object in the Outline Viewer to locate it and restore its shape and fill.

Supported shape features, continued

Feature	Notes about conversion behavior
Polygons	This object can be converted as a bitmap, or you can apply an open path to the object in InDesign so the object's shape and fill can be edited in Designer. If you apply an open path to it, the object might not be visible when you open the imported page in Designer. You can select the object in the Outline Viewer to locate it and restore its shape and fill.
Rectangles	Fully supported
Shape	Appears on the converted page only if you convert the entire page as bitmap
Shear	Appears on the converted page only if you convert the entire page as bitmap
Transparency	Appears on the converted page only if you convert the entire page as bitmap

Supported text features

Feature	Notes about conversion behavior
All caps	Supported if you select to convert the feature as bitmap (by selecting the **All caps** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Baseline shift	Supported if you select to convert the feature as bitmap (by selecting the **Baseline shift** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Double-byte characters	Supported if you select to convert the feature as bitmap (by selecting the **Special characters** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Discretionary hyphens	Supported if you select to convert the feature as bitmap (by selecting the **Special Dashes** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Drop caps	Supported if you select to convert the feature as bitmap (by selecting the **Drop caps** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Em dashes	Supported if you select to convert the feature as bitmap (by selecting the **Special Dashes** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Em spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
En dashes	Supported if you select to convert the feature as bitmap (by selecting the **Special Dashes** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
En spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Figure spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Flowing text	Each flowing text box is converted into an individual, non-flowing text box. Each text box contains only the text that appeared in that text box before conversion. You can recreate text flow in Designer using flow frames.

Supported text features, continued

Feature	Notes about conversion behavior
Flush spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Hair spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Indents	Converted as bitmap
Kerning	Supported if you select to convert the feature as bitmap (by selecting the **Kerning** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Non-breaking hyphens	Supported if you select to convert the feature as bitmap (by selecting the **Special Dashes** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Non-breaking spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Outline	Supported if you select to convert the feature as bitmap (by selecting the **Outline** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Paragraph rules	Supported if you select to convert the feature as bitmap (by selecting the **Paragraph rules** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Punctuation spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Scaling	Supported if you select to convert the feature as bitmap (by selecting the **Scale** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Skew	Supported if you select to convert the feature as bitmap (by selecting the **Text skew** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Small caps	Supported if you select to convert the feature as bitmap (by selecting the **Small caps** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Space after	Converted as bitmap
Space before	Converted as bitmap

Supported text features, continued

Feature	Notes about conversion behavior
Tab settings	The following tab alignment settings are supported: • Center-justified tab • Left-justified tab • Right-justified tab The following features are supported if you select to convert them as bitmap. Otherwise, they are ignored: • Align on tabs • Tab leaders
Text alignment	The following text alignment settings are supported: • Bottom alignment (vertical) • Center alignment (horizontal and vertical) • Justified alignment (horizontal) • Left alignment (horizontal) • Right alignment (horizontal) • Top alignment (vertical) Full justify is supported if you select to convert it as bitmap (by using the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored, unless you convert the entire page as bitmap. The following text alignment settings are supported if you convert the entire page as a bitmap: • Forced justify • Justified alignment (vertical) • Justified with last line aligned center • Justified with last line aligned left • Justified with last line aligned right • Justify all
Text box	Each InDesign text box is converted as one Designer text box (with the exception of flowing text). If a text box contains unsupported characters or features and you select to convert these features to bitmap during the conversion, the entire text box is converted to bitmap format. Before the conversion, change the size of text boxes so they closely fit the text they contain. If the text boxes are significantly larger than the text, they can cause issues during the conversion.
Text skew	Supported if you select to convert the feature as bitmap (by selecting the **Text skew** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Text slant (with object)	Appears on the converted page only if you convert the entire page as bitmap
Text wrapping	Appears on the converted page only if you convert the entire page as bitmap

Supported text features, continued

Feature	Notes about conversion behavior
Thin spaces	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Type on a path	Appears on the converted page only if you convert the entire page as bitmap
Variables	Fully supported. To include variables in a converted document, surround the variable name in InDesign with brackets (for example, `<CustomerName>`).

Miscellaneous features

Feature	Notes about conversion behavior
Embedded objects	These types of objects might not convert correctly and are often separated during conversion.
Grouped objects	These types of objects might not convert correctly and are often separated during conversion.
Images with overlapping text	After the conversion, the overlapping text is treated as an uneditable part of the image.
Layers	All visible layers are converted into a single design layer in Designer. Hidden layers do not convert.
Objects that extend past the design area	These types of objects might not convert correctly.
Tables	Designer and InDesign draw tables differently. In InDesign, you embed tables in a text box, which lets you create additional text to surround the table. The InDesign Converter converts this object as two separate design objects: a text box and a table. The table is no longer embedded in the text box. After you have imported the document into Design and Production, you can apply dynamic table settings.

Note: If you choose to convert unsupported characters or features to bitmap and these features are located in a text box, the entire text box is converted to bitmap.

3.2 Converting the InDesign design to a DXF File

You use the InDesign Converter to convert files from InDesign format into DXF format. Before you run the InDesign Converter, you must resolve any issues that appear with a Missing or Modified status on the **Preflight** dialog box in InDesign. Objects with either a Missing or Modified status can cause errors in the conversion and will not appear when the DXF file is imported into Exstream Design and Production. In addition, you must install any fonts used in the InDesign file on the computer where the DXF file will be imported.

There are three ways you can convert files from InDesign format. The method you choose to use depends on how you want to use the design.

- **Convert the entire file into a bitmap**—The entire InDesign page is converted into a TIFF image, which you can then import into Designer. All InDesign objects appear in the TIFF; however, you cannot make any changes to the converted file. You might choose to use this method if you do not need to make any changes to the design after it is imported and if you want the converted file to appear exactly the same as the InDesign file.
- **Convert specific features to bitmap**—You specify which features are converted to bitmap. This method gives you the most control over the conversion process, and allows you to make changes to most design objects in Designer. You might choose to use this method if you want to retain as much of the original appearance of the design as possible, while still having the ability to make changes to the design in Designer. If you choose to use this method, make sure you are familiar with the supported objects and how they are treated by the InDesign converter.
- **Convert all supported features and ignore unsupported features**—The converter converts all supported features. Any features that are not natively supported in the Design and Production format, or that must be converted to bitmap format in order to appear in the converted file, are ignored. You might choose to use this method if your InDesign design does not contain many unsupported objects, if it will be easy to recreate the ignored objects in Designer, or if you must make extensive changes to the page in Designer in the future.

To convert the file from InDesign format:

1. In InDesign, from the **File** menu, select **Export to Exstream**.

   The **Export to Exstream** dialog box opens.

   > **Tip:** If you use a configuration file to automatically load pre-defined conversion settings, click **Load**, select the configuration file, and skip to step 3.

2. In the **Convert to Bitmap Setting**s area, specify how you want the InDesign Converter to treat objects that are not natively supported by Design and Production

To	Do this
Convert the entire file into a bitmap	a. Select **All**. b. In the **Convert to Bitmap Device** area, select the settings for the TIFF that will be created. **Tip:** Generally, the resolution you select should match the dots per inch (dpi) of the target output device. Also, keep in mind that high dpi resolution can result in very large image file sizes.

To	Do this
Convert specific features to bitmap format	a. Select **Some**. b. In the **Convert to Bitmap if checked** area, select the objects that you want the converter to convert to bitmap format. c. At the bottom of the **Convert to Bitmap if checked** area, select the **Embed Images in DXF** check box if you want images in the InDesign page to be embedded in the converted file. You must select this check box if you run InDesign and the design environment on different computers. However, if InDesign and Design and Production are located on the same computer, you can clear this check box to improve the speed of the conversion process. After the conversion, the converted file must be used on the computer that contains InDesign and the image files in order for the images to appear.
Convert all supported features and ignore unsupported features	a. Select **None**. b. At the bottom of the **Convert to Bitmap if checked** area, select the **Embed Images in DXF** check box if you want images in the InDesign page to be embedded in the converted file. You must select this check box if you run InDesign and the design environment on different computers. However, if InDesign and Design and Production are located on the same computer, you can clear this check box to improve the speed of the conversion process. After the conversion, the converted file must be used on the computer that contains InDesign and the image files in order for the images to appear.

Tip: If you want to save the conversion settings you just defined, click **Save** in the **Convert to Bitmap Settings** area. The conversion settings are saved as an INI file that you can use in future conversions to automatically load settings for the converter. This feature allows you to quickly load pre-defined settings for each conversion you must complete.

3. In the **File name** box, click  to specify the location and file name of the DXF file the InDesign Converter will create. If you use a Macintosh, you cannot enter a name manually in the **File name** box; you must click  to specify the location and file name. You will select this file when you import the DXF into Design and Production.
4. Click **Export**.

   The conversion process begins. The **Log** dialog box opens and provides details about the conversion. You can use the information it provides for troubleshooting.

When the conversion is complete, one DXF file for each InDesign file will be located in the directory you specified. You can then import the DXF file or files into Exstream Design and Production.

For more information about DXF files, see "Understanding the DXF format" on page 7.

3.3 Importing the generated DXF file

After you run the InDesign Converter, import the DXF file into Design and Production.

To import the DXF file:

1. In Design Manager, right-click the **Pages** heading and select **Import from DXF**.

   The **Import DXF** dialog box opens.

2. Navigate to the DXF file you want to import. You can select multiple files to import by holding down CTRL and selecting the files.

3. In the **Import** box, specify the Library folder into which you want to import the pages.

   a. Click  .

      The **Folders** dialog box opens.

   b. Select the folder and click **OK**.

      The **Folders** dialog box closes and the folder you selected appears in the **Import** box.

4. In the **Page Type** list, select the Exstream Design and Production page settings you want to apply to the imported page. You can change this setting later, but you must specify a setting option for the initial import.

   a. Depending on the way your design standards are enforced, select either **Use specified Paper Type** or **Use specified Page Template**.

   b. In the list below the **Page Type** list, select either the paper type or page template you want to apply to the imported pages.

5. In the **Variable Import** list, select how variables in the converted document are handled.

To	Do this
Ignore any variables in the original document	Select **Ignore variables**.
Honor only original variables in the file that exist in the Exstream Design and Production Library	a. Select **Map to existing variables only**. b. In the **Variable Folder** box, specify the Design Manager folder that contains the variables. i. Click  . The **Folders** dialog box opens. ii. Select the folder and click **OK**. The **Folders** dialog box closes and the folder you selected appears in the **Variable Folder** box.
Honor all variables from the original file and create new Exstream Design and Production variables for those that do not currently exist in the Library	a. Select **Create missing variables**. b. In the **Variable Folder** box, specify the Design Manager folder that contains the variables. i. Click  . The **Folders** dialog box opens. ii. Select the folder and click **OK**. The **Folders** dialog box closes and the folder you selected appears in the **Variable Folder** box.

6. Click **Open**.

The **Log** dialog box displays informational messages about the import process. The files then appear in the Design Manager Library. If the DXF file contains multiple pages, Exstream Design and Production names the imported pages according to their order in the DXF file. For example, if you import a DXF file named `Test` that contains three pages, the file names are `TEST.DXF.page.1`, `TEST.DXF.page.2`, and `TEST.DXF.page.3`. You can rename the pages in Design Manager after they have been imported.

After you import the files, make sure to visually check the pages and make adjustments to objects as needed.

Chapter 4: Importing Metacode designs

The Metacode Converter lets you import normalized Metacode files into the Exstream Design and Production environment. The Metacode Converter analyzes a specified Metacode file and creates a DXF file, which you can then import into Exstream Design and Production.

This section discusses the Metacode Converter version 1.2.

Most Metacode design objects are supported in the Exstream Design and Production environment, and when they are converted, the objects become editable. For example, when a page is converted and imported into Design and Production, you can see the text and change it just as you would for pages created in Exstream Design and Production.

The Metacode Converter supports most binary integer byte order types, block sizes, and highlight colors. The converter retrieves the font metrics, variables, and other data for the converted files so that after the conversion, the imported file appears as similar as possible to the original file.

This section discusses the following topics:

4.1 Supported Metacode formats

The following table describes the Metacode formats that are supported by the Metacode Converter.

Supported Metacode Formats

File type	Description
4 byte, big endian, inclusive	Record length indicator is 4 bytes long; the most significant bytes first; length includes the record length
4 byte, big endian, exclusive	Record length indicator is 4 bytes long; the most significant bytes first; length does not include the record length
4 byte, little endian, inclusive	Record length indicator is 4 bytes long; the least significant bytes first; length does not include the record length

Supported Metacode Formats, continued

File type	Description
4 byte, little endian, inclusive	Record length indicator is 4 bytes long; the least significant bytes first; length does not include the record length
2 byte, big endian, inclusive	Record length indicator is 2 bytes long; the most significant bytes first; length includes the record length
2 byte, big endian exclusive	Record length indicator is 2 bytes long; the most significant bytes first; length does not include the record length
2 byte, little endian, inclusive	Record length indicator is 2 bytes long; the least significant bytes first; length includes the record length
2 byte, little endian, exclusive	Record length indicator is 2 bytes long; the least significant bytes first; length does not indicate the record length
MVS FTP records	Record length indicator starts with 80, followed by a 2-byte length count (exclusive); can be successfully uploaded to mainframe without unblocking, or downloaded to a personal computer with no blocking required
Blocked MVS FTP records	Supports z/OS File Transfer Protocol (FTP) records that are blocked
CR/LF delimited records	Records are separated by `0D0A` (carriage return/line feed)
LF delimited records	Records are separated by line feed
Fixed-length records	Records are padded as necessary to force them to an equal size
Variable-length records	Two-byte block length is followed by a 2-byte pad; the most significant bytes first; length does not include the record length
Barr short format	Spooling format of fixed record length (255 bytes) with 4-byte header; the least significant bytes first; 1-byte start length (does not include the byte itself); and 1- byte end length (does not include the byte itself)
Barr long format	Spooling format of fixed record length (64K bytes) with 2-byte start length; the least significant bytes first; and 2-byte end length (does not include the bytes)

If your files are in a different format, you can specify a custom format.

For information on specifying a custom format, see "Defining a Custom Blocking Format" on page 66.

4.2 Preparing Metacode files for conversion

Metacode files must reside on Windows in order to convert them using the Metacode Converter. The files can have any filename extension, such as .met or .nor, but they must be normalized.

If the Metacode files you want to convert are stored in an Electronic Data Library (EDL) on the mainframe, you must copy the files from the EDL to a Windows file system. The quickest way to copy files to a Windows system is to extract the Metacode documents individually from the EDL using the VLAM Commander. However, you can also extract individual forms to either a fixed length or variable length dataset. This method is less efficient since it pads the ends of records, but it is easier to manipulate. If you want to write to a variable length dataset, use the following sample JCL to add length fields to the record:

```
//EXGENR JOB 'UE',(00000),CLASS=A,MSGCLASS=X
/*
/*
//STEP1      EXEC PGM=IEBGENER,REGION=4M
//SYSIN      DD DUMMY
//SYSPRINT   DD SYSOUT=*
//*
//*NOTE: THE REAL DCB OF SYSUI1 is VB 300 3120. USE THE FOLLOWING
//* TO GET IEBGENER TO PROVIDE FOUR BYTE RDW AND BDW IN THE OUTPUT
//SYSUT1     DD DSN=EXSTREAM.METAIN(SAMPLE),DISP=SHR,
//            DCB=(LRECL=3120,RECFM=U)
//*
//*NOTE: THE DCB FOR SYSUT2 WILL BE COPIED FROM SYSUT1 BY IEBGENER
//SYSUT2     DD DSN=EXSTREAM.OUTLOD1(NEWMETA),DISP=(NEW,CATLG),
//            SPACE=(TRK,(15,15,3)),UNIT=SYSDA
//*
//*THE FOLLOWING EXAMPLE IS FOR AN EXISTING DATASET
//*SYSUT2        DSN=EXSTREAM.OUTLOD1(NEWMETA),DISP=SHR
```

Complete a binary download to copy these files to a Windows directory.

4.3 Converting the Metacode design to a DXF file

Because the Metacode format is complex, you must provide information about the file format and contents before running the converter.

To convert designs from Metacode format, complete the following tasks as needed:

- “Using a Configuration File to Automatically Load Metacode Conversion Settings” on the next page
- “Specifying the Format of Metacode Files That Will Be Converted” on the next page
- “Specifying the Output Directory for the DXF Files” on page 67
- “Mapping Metacode Fonts for Conversion” on page 67

- "Specifying Custom Variable Names for Converted Metacode Variables" on page 70
- "Saving the Settings and Running the Metacode Converter" on page 71

Note: The Metacode Converter provides an **Images** tab. You do not need to interact with this tab because its functionality is not supported in this release.

If you use configuration files to automatically load converter settings, you can specify the format of files and then run the converter. You do not need to complete the other steps. Configuration files are saved settings, usually from previous conversions.

For information on using a configuration file, see "Using a Configuration File to Automatically Load Metacode Conversion Settings" below.

4.3.1 Using a Configuration File to Automatically Load Metacode Conversion Settings

1. Open the Metacode Converter. By default, this program is installed in the Exstream directory.
2. Click **File > Open Configuration**.

   The **Open** dialog box opens.
3. Select the configuration file and click **Open**.

   The **Open** dialog box closes and the saved configuration settings populate the fields in the Metacode Converter.
4. Check the **Fonts** tab and verify that all the entries appear in black text. You must remap entries that appear in red.

Tip: If previously mapped fonts appear to be unmapped when you open a configuration file, the internal font name differs from the font's file name. Rename the font file name to match the internal name or use a hex editor to change the font's internal name.

For information on mapping fonts, see "Mapping Metacode Fonts for Conversion" on page 67.

If you make changes to the converter settings and then save the configuration file, the new settings are appended to the original configuration file.

For information on saving the converter settings in the configuration file, see "Saving the Settings and Running the Metacode Converter" on page 71.

4.3.2 Specifying the Format of Metacode Files That Will Be Converted

1. Open the Metacode Converter. By default, this program is installed in the Exstream directory.
2. On the **Inputs** tab, select the file you want to convert. You can use one or both of the methods described in the table as many times as needed to specify as many files as you want to convert in one batch process. If you want to convert all the files in a folder, the files must all have the same properties, such as binary integer byte order and highlight color.

To	Do this
Manually select one or more Metacode files to convert	a. Click **Add Files**. The **Open** dialog box opens. b. Locate the file, select it, and click **Open**. You can select multiple files by holding down the CTRL key and selecting the files. The **Open** dialog box closes and the **MetaCode Input Format** dialog box opens.
Convert all of the Metacode files in a specified folder (and all of its subfolders)	a. Click **Add Folder**. The **Browse for Folder** dialog box opens. b. Locate the folder that contains the PDF, select it, and click **OK**. The **Browse for Folder** dialog box closes and the **MetaCode Input Format** dialog box opens.

3. From the box at the top of the **MetaCode Input Format** dialog box, select the binary integer byte order in the file(s). The option you select specifies the byte order implicit in the character set.

   For information on the formats supported by the Metacode Converter, see "Supported Metacode formats" on page 61.
4. If the option you selected requires you to specify the block size, enter it in the **Block size** box.
5. In the **DJDE Iden** box, enter a Metacode string identity to include in the DXF file. Normally, this string contains two literal characters followed by the Dynamic Job Description Entry (DJDE). If the corresponding Xerox JSL file is available for the file you are converting, you can find the mandatory **DJDE Iden** value in that file.
6. In the **Offset** box, enter the number of characters in the Metacode string the converter must skip before reading actual data. If the corresponding Xerox JSL file is available for the file you are converting, you can find the mandatory **Offset** value in that file.
7. In the **Skip** box, enter the starting position of the IDEN string. The initial position is zero. If

the corresponding Xerox JSL file is available for the file you are converting, you can find the mandatory **Skip** value in that file.

8. From the **DJDE** drop-down list, select the character set used by the file.
9. The **Highlight** color box displays the highlight color. If the displayed color is incorrect or missing, click **Set color** to select the correct highlight color.
10. From the **Carriage control** drop-down list, select the code type used for carriage return controls.
11. Click **OK**.

    The **MetaCode Input Format** dialog box closes and the file(s) you selected appear on the **Inputs** tab.

Defining a Custom Blocking Format

If the formats the Metacode Converter provides do not offer the blocking format you need, you can define a custom format for the converter to use during the conversion.

> **Tip:** If you create a custom format that you must use for future conversions, consider saving the converter settings before running the converter. Saving the custom format needed for your organization's file specifications lets you avoid re-creating the custom format each time you run the converter.

For information on saving the converter settings, see "Saving the Settings and Running the Metacode Converter" on page 71.

To define a custom format:

1. On the **MetaCode Input Format** dialog box, double-click the **Custom format** option in the box at the top of the dialog box.

   The **Custom Record Format** dialog box opens.
2. If records place the most significant bytes first, select the **Big Endian** check box. Clear the check box if records place the least significant bytes first.
3. If the length does not include the record length indicator, select the **Exclusive** check box. Clear the check box if the length includes the record length indicator.
4. In the **Initial Record Skip Bytes** box, enter the number of bytes that must be skipped before the first record begins.
5. Use the remaining properties to identify whether the file uses indicator bytes to indicate record lengths and how these bytes should be handled:

If	Do this
The file uses indicator bytes	a. Select the **Indicator Bytes** check box. b. From the adjacent drop-down list, select the number of bytes used to indicate the record length. c. If records are blocked in the file, select the **Blocked** check box in the **Blocking** area. d. If the records are a fixed length, select the **Fixed Length** check box in the **Blocking** area and enter the size of the records in the adjacent size box.
The file does not use indicator bytes	In the **Carriage Controls** area, select the method used to indicate the end of a record. You can select either the **LF** radio button or the **CRLF** radio button.

6. Click **OK**.

   The **Custom Record Format** dialog box closes.

4.3.3 Specifying the Output Directory for the DXF Files

By default, the Metacode converter gives the DXF file it creates the same name as the original Metacode file (for example, `Sample.met` becomes `Sample.dxf`) and places it in the same directory as the original. To change the location where Metacode files are placed, or to change the name given to DXF files, use the options on the **Outputs** tab.

To	Do this
Specify a unique directory and file name for individual DXF files that will be created	1. Select the Metacode file for which you want to specify a directory. 2. Click **Set File**. The **Save As** dialog box opens. 3. Locate the directory where you want the DXF file to be located, and specify a name for the file. 4. Click **Save**. The **Save As** dialog box closes and the name and directory you specified for the file appear in the **Outputs** box.

To	Do this
Specify a unique directory for all the DXF files that will be created	1. Click **Select All**. The Metacode files in the **Outputs** box are selected for you. 2. Click **Set Folder**. **The Browse for Folder** dialog box opens. 3. Locate the folder where you want the DXF files to be placed, select it, and click **OK**. The **Browse for Folder** dialog box closes and the location you specified appears in the **Location** column of the **Outputs** box. **Tip:** To override this setting for specific file, follow the steps above to specify a unique directory and file name for specific files.

4.3.4 Mapping Metacode Fonts for Conversion

Because the fonts used in Metacode files are not Windows fonts, you must provide the True Type equivalents of the fonts used in the Metacode design. After you have specified the Metacode files you want to convert, the **Fonts** tab of the Metacode Converter is populated with all the fonts found in the files. You can use this tab to open a .fnt file for each Metacode font. The converter analyzes the .fnt files to gather specific metrics for each font. Then you specify the Windows font you want to use for each Metacode font.

The **Fonts** tab uses the following color legend to identify whether the fonts have been mapped:

- **Red**—Font has been read from the Metacode file, but you have not provided its metrics or specified its Windows equivalent
- **Orange**—Font has been read from the Metacode file, you have provided its metrics, but you have not specified its Windows equivalent
- **Black**—All the necessary font information has been provided and the font is ready for the conversion process

Before mapping the fonts, make sure all the .fnt files used in the Metacode files are available on the computer where you will run the converter.

To map the Metacode fonts:

1. Click the **Fonts** tab.
2. Open the font information for one or all of the unmapped fonts.

To	Do this
Open font information for a specific font	a. Select a font whose information has not been provided to the converter. These fonts appear in red. b. Click **Load Font**. The **Open** dialog box opens. c. Select the .fnt file associated with the Metacode font and click **Open**. The **Open** dialog box closes and the **Type** column is updated to indicate whether the font is a text (character-based) or logo (image-based) type font. The font entry also changes to orange to indicate that you must specify its Windows equivalent. d. Repeat step a through step c for all the files in the **Fonts** box, until all the font entries are orange.
Open font information for all the fonts and allow the converter to automatically associated the information with each font **Note:** The .fnt files must all be located in the same folder to use this method.	a. Click **Select all**. b. Click **Load Font**. The **Open** dialog box opens. c. Select the folder that contains all of the font information and click **Open**. The **Open** dialog box closes and **Type** column is updated to indicate whether the font is a text (character-based) or logo (image-based) type font. The font entries also change to orange to indicate that you must specify their Windows equivalents.

3. Select a font for which you must identify a Windows equivalent. These fonts appear in orange.
4. Click **Set Substitute**.

   The **Font** dialog box opens.
5. Use the **Font** dialog box to select the Windows equivalent font and font style.
   a. In the **Font** box, select the equivalent font. If the font is a logo font, it is recommended that you select a unique Windows font that will make it easy to find the font in Designer. After the page is converted and imported into Designer, you can replace each instance of this unique font with an image.
   b. In the **Font style** box, select the style.
   c. If you know the equivalent font size, enter it in the **Size** box. If you are not sure of the size, you do not have to specify it, and the converter will determine it automatically.
   d. Click **OK**.

      The **Font** dialog box closes and a message opens.
6. Click **Yes** if you want the converter to determine the font size. If you want the converter to use the font size you specified, click **No**.

The message closes and the **Substitute** column is updated to reflect the substitution font you specified. The font entry also changes to black to indicate that the font is ready for the conversion process.

7. Repeat step 3 through step 6 for every font in the **Fonts** box. If you do not provide a Windows equivalent for all fonts, the conversion will fail.

For information on the extra steps you must carry out to map fonts with character mapping variations, see "Mapping Metacode Fonts with Character Set Variations" below.

Mapping Metacode Fonts with Character Set Variations

If the character set of a Metacode font varies from the True Type equivalent, you must provide additional information so the converter handles the font correctly. Before mapping these types of fonts, make sure you know the ASCII equivalents for the characters that vary.

To map fonts with character set variations:

1. In the **Fonts** box, select the font.
2. Click **Set Characters**.

   The **Character Mapping** dialog box opens.
3. Click **Add**.

   A second **Character Mapping** dialog box opens.
4. In the **MetaCode** box, enter the hexadecimal notation for the Metacode character. You must precede the value with `0x` to indicate that the value is hexadecimal.

   > **Tip:** You can use the Xerox SAM command to help you identify Metacode character values.

5. In the **DXF** box, enter the Unicode code point of the ASCII equivalent. You do not have to include the U+ prefix with the value you enter.

   > **Tip:** You can find the Unicode code point value by using the Windows character map program.

6. Click **OK**.

   The **Character Mapping** dialog box closes and the character mapping you entered appears in the **Character Mapping** dialog box.
7. Click **OK**.

The **Character Mapping** dialog box closes.

8. Repeat step 1 through step 7 to provide the ASCII equivalent for as many fonts as needed.

4.3.5 Specifying Custom Variable Names for Converted Metacode Variables

When you import a DXF file into Design and Production, Exstream Design and Production can automatically create variables in the design database for all of the variables contained in the DXF. When Design and Production creates variables automatically, it names each variable based on the corresponding Metacode variable name and replaces unsupported characters, such as spaces and periods, with underscores. However, you can override the default naming with custom names. You define the custom variable names in the Metacode Converter, and when the DXF file is imported, Design and Production will name the variables according to the instructions provided in the DXF file.

To specify custom variable names, complete the following tasks as needed:

- "Specifying a Prefix for All Variables" below (for example, all variables in the converted document have the prefix `Meta_`)
- "Providing a Unique Name for a Specific Variable" on the next page

Specifying a Prefix for All Variables

1. Click the **Variables** tab.

   A list of all the variables found in the input files appears in the box.

2. Click **Set Prefix**.

   The **Set Variable Prefix** dialog box opens.

3. In the **Variable Prefix** box, enter the prefix you want to apply to the variable. Follow the Exstream Design and Production rules for variable naming.

   The variables in the **Sample variables** box change to demonstrate how variables will appear with the prefix.

4. Click **OK**.

   The **Set Prefix** dialog box closes and the variable names in the **Substitute** column are updated to reflect how the variables will appear after the DXF is imported into Design and Production.

Providing a Unique Name for a Specific Variable

1. Click the **Variables** tab.

   A list of all the variables found in the input files appears in the box.

2. Select the variable whose name you want to change.

3. Click **Set Substitute**.

   The **Set Variable Mapping** dialog box opens.

4. In the **DXF variable** box, enter the new name for the variable. Follow the Design and Production rules for variable naming.

5. If you know the variable's type, you can optionally select it from the **Variable type** drop-down list. By default, all variables are imported as string type variables. If you do not know the variable type or you do not want to set it before the conversion, you can change the variable type in Design Manager after you import the DXF file.

6. Click **OK**.

   The **Set Variable Mapping** dialog box closes and the new variable name appears in the **Substitute** column.

4.3.6 Saving the Settings and Running the Metacode Converter

Before running the converter, you can save the configuration settings you have defined in the converter. This optional step can save you time if you will convert similar files in the future.

To save the conversion settings:

1. Click **File > Save Configuration As**.

   The Save As dialog box opens.

2. Specify the file name and the location.

3. Click **Save**.

   The **Save As** dialog box closes. The conversion settings are saved as a .conf file that you can use in future conversions to automatically load settings for the conversion.

4. When you are ready to run the conversion, click **Convert**.

   The **Converting Metacode to DXF** dialog box opens and displays information about the conversion process. When the conversion is complete, one DXF file for each Metacode file will be located in the directory you specified. You can then import the DXF file or files into Exstream Design and Production.

For more information on DXF files, see “Understanding the DXF format” on page 7.

4.4 Importing the generated DXF file

After you run the Metacode Converter, import the DXF file into Design and Production.

To import the DXF file:

1. In Design Manager, right-click the **Pages** heading and select **Import from DXF**.

   The **Import DXF** dialog box opens.

2. Navigate to the DXF file you want to import. You can select multiple files to import by holding down CTRL and selecting the files.

3. In the **Import** box, specify the Library folder into which you want to import the pages.

   a. Click  .

      The **Folders** dialog box opens.

   b. Select the folder and click **OK**.

      The **Folders** dialog box closes and the folder you selected appears in the **Import** box.

4. In the **Page Type** list, select the Exstream Design and Production page settings you want to apply to the imported page. You can change this setting later, but you must specify a setting option for the initial import.

   a. Depending on the way your design standards are enforced, select either **Use specified Paper Type** or **Use specified Page Template**.

   b. In the list below the **Page Type** list, select either the paper type or page template you want to apply to the imported pages.

5. In the **Variable Import** list, select how variables in the converted document are handled.

To	Do this
Ignore any variables in the original document	Select **Ignore variables**.
Honor only original variables in the file that exist in the Exstream Design and Production Library	a. Select **Map to existing variables only**. b. In the **Variable Folder** box, specify the Design Manager folder that contains the variables. i. Click  . The **Folders** dialog box opens. ii. Select the folder and click **OK**. The **Folders** dialog box closes and the folder you selected appears in the **Variable Folder** box.
Honor all variables from the original file and create new Exstream Design and Production variables for those that do not currently exist in the Library	a. Select **Create missing variables**. b. In the **Variable Folder** box, specify the Design Manager folder that contains the variables. i. Click  . The **Folders** dialog box opens. ii. Select the folder and click **OK**. The **Folders** dialog box closes and the folder you selected appears in the **Variable Folder** box.

6. Click **Open**.

The **Log** dialog box displays informational messages about the import process. The files then appear in the Design Manager Library. If the DXF file contains multiple pages, Exstream Design and Production names the imported pages according to their order in the DXF file. For example, if you import a DXF file named `Test` that contains three pages, the file names are `TEST.DXF.page.1`, `TEST.DXF.page.2`, and `TEST.DXF.page.3`. You can rename the pages in Design Manager after they have been imported.

After you import the files, check make sure to visually check the pages and make adjustments to objects as needed.

4.5 Cleaning up the imported DXF file

Although many objects in a Metacode file are converted cleanly and appear exactly the same in the Design and Production environment as they did previously, some objects might need to be

adjusted. In particular, logo (image) fonts often do not appear the same after the conversion. You can replace these fonts with Exstream Design and Production-supported graphics after the conversion.

Text is also often placed into multiple text boxes. Designer provides tools you can use to easily clean up these types of text boxes so that the text is located in only one text box. Combining text into one text box makes it easier to change the text properties and to reuse the content in future applications.

You can use Designer's tools to combine multiple text boxes into one of the following types of objects:

- **A single line of text**—Allows you to retain the absolute positioning of the text
- **A single paragraph**—Provides more control over the text flow and positioning

4.5.1 Combining Individual Text Boxes into a Line

In order to use this method, the text boxes you want to combine must be located on the same horizontal line. After text has been combined into a line, Designer removes the absolute positioning data to make it easier to edit the text.

To combine text into a line:

1. Press and hold CTRL and then select the text boxes you want to combine. You can select the text boxes on the design page or in the Outline Viewer.
2. From the **Tools** menu, select **Import Cleanup > Combine Text into Line**.

   The text boxes you selected are combined into a single line in a single text box.

Tip: If you receive a message stating that the text boxes cannot be combined into a line, the selected objects might not be aligned horizontally. Use the method for combining text into a paragraph instead.

4.5.2 Combining Individual Text Boxes into a Paragraph

1. Press and hold CTRL and then select the text boxes you want to combine. You can select the text boxes on the design page or in the Outline Viewer.
2. From the **Tools** menu, select **Import Cleanup > Combine Text into Paragraph**.

   The **Import Cleanup** dialog box opens.
3. Make selections on the **Import Cleanup** dialog box to specify the way you want the cleanup to be handled.

To	Do this
Insert a soft-return line break at the end of each line of text	Select the **Add line breaks between lines** check box.
Force lines to wrap with the natural flow of text	Clear the **Add line breaks between lines** check box.
Preserve the position of text as it appears in the original document	Select the **Preserve absolute positions** check box.
Force paragraphs to merge with a regular flow of text **Tip:** Use this setting if you plan to edit the text.	Clear the **Preserve absolute positions** check box

4. Click **OK**.

   The **Import Cleanup** dialog box close and the text is combined in a single text box according to the settings you specified.

Chapter 5: Importing OGL designs

The OGL Converter lets you import IBM OGL/370 overlays into the Exstream Design and Production environment. The OGL Converter analyzes a specified OGL file and creates a DXF file, which you can then import into Design and Production.

This section discusses the OGL Converter version 1.0.

Most OGL design parameters are supported in the Design and Production environment, and when they are converted, become editable objects. For example, OGL box and circle parameters are supported fully. Therefore, when a page is converted and imported into Design and Production, you can see the properties of these objects and change them just as you would for pages created in Design and Production.

Some features used in OGL files are not supported as Design and Production objects. The OGL converter ignores these features. You can recreate them after you import the converted file to Design and Production.

In addition, the OGL overlays must be in ASCII format for the OGL Converter. The OGL Converter operates on source overlays, not compiled overlays.

Before using the OGL Converter, make sure your system meets the following requirements:

- Operating system Windows 7 or later
- Apache Xerces

This section discusses the following topics:

5.1 Supported OGL features

The following table describes the overlay features the OGL Converter supports:

Supported overlay features

Parameter type	Supported parameters
Control	NOSOI

Supported overlay features, continued

Parameter type	Supported parameters
Drawbox	Diagonal Height Rounded
Drawcircle	Portion (half, quarter, radius, whole)
Drawgraphic	Box Circle Ellipse Fillets Marker Path
Drawpath	Connection Miter To
Drawrule	Direction Length
Place	Group Segment
Position	Coordinate Origin Segment Settext
Setunits	Corner length Line space Units
Shade	Area Border thickness (bold, light, medium, n (pels)) Border type Bow width Type
Type	Font

5.2 Converting the OGL design to a DXF file

You use the OGL Converter to convert designs from OGL format to DXF format. Before running the converter, make sure the following prerequisite tasks are completed:

- The OGL Converter supports only black and white conversion. If an overlay contains color definitions, it can cause errors in the conversion process. Before running the converter, make sure that all color definitions are set to black-and-white color values.
- You must have a completed ScanFonts form available locally out before running the converted. A ScanFonts form contains information about the fonts used in an OGL overlay. A complete list of font forms is provided in the `ImportOGL.zip` file in the installation directory. If fonts are missing from the ScanFonts form, the log file the OGL Converter creates reports the missing fonts.

To convert a design from OGL format:

1. Open the OGL Converter. By default, this program is installed in the Exstream directory.

   The **Convert OGL/370** dialog box opens.

2. In the file browser box, select the OGL file you want to convert.

   The file name appears in the **File name** box.

3. In the **Output folder** box, specify the directory where the DXF file will be placed. If you do not specify an output directory, the DXF file is created in the folder where the converted overlay is located.

   a. Click  .

      The **Open** dialog box opens.

   b. Select the directory and click **Open**.

      The **Open** dialog box closes and the directory name appears in the **Output folder** box.

4. In the **Font form** box, select the ScanFonts form that contains the font information for the selected OGL overlay.
5. In the **Image folder** box, select the PSEG images associated with the overlay.
6. Using the **Log file** box and check boxes, set up the options for the log file.

   a. In the **Log file** box, specify the name of the log file that is generated by OGL Converter and the directory where it will be placed. If you do not specify a directory, the log file is created in the folder where the DXF file is placed.

   b. If you specified an existing log file in the **Log file** box and you want the new log

information to be added to the end of the file, select the **Append to log** check box. If you clear the **Append to log** check box, the OGL Converter overwrites the existing log information with the new log information.

c. If you want the log file to open automatically after the conversion is complete, select the **Show log** check box.

7. Click **Convert**.

When the conversion is complete, one DXF file for each OGL overlay will be located in the directory you specified. You can then import the DXF file or files into Exstream Design and Production.

For more information on DXF files, see "Understanding the DXF format" on page 7.

5.3 Importing the generated DXF file

After you run the OGL Converter, import the DXF file into Design and Production.

To import a DXF file into the Design and Production environment:

1. In Design Manager, right-click the **Pages** heading, and select **Import from DXF**.
2. In the **Import DXF** dialog box, navigate to the DXF file that you want to import. You can select multiple files to import by holding down CTRL and selecting them.
3. In the **Import** box, specify the Library folder into which you want to import the pages.

   a. Click  .

   The **Folders** dialog box opens.

   b. Select the folder and click **OK**.

   The **Folders** dialog box closes and the folder you selected appears in the **Import** box.

4. In the **Page Type** list, select the Design and Production page settings you want to apply to the imported page. You can change this setting later, but you must specify a setting option for the initial import.

   a. Depending on the way your design standards are enforced, select either **Use specified Paper Type** or **Use specified Page Template**.

   b. In the list below the **Page Type** list, select either the paper type or page template you want to apply to the imported pages.

> **Note:** Do not make any selections in the **Variable Import** list. Variables are not converted from OGL format.

5. Click **Open**.

The **Log** dialog box displays informational messages about the import process. The imported objects then appear in the Design Manager Library. If the DXF file contains multiple pages, Design and Production names the imported pages according to their order in the DXF file. For example, if you import a DXF file named `Test` that contains three pages, the file names are `TEST.DXF.page.1`, `TEST.DXF.page.2`, and `TEST.DXF.page.3`. You can rename the pages in Design Manager after they have been imported.

Chapter 6: Importing PDF designs

The PDF Converter lets you import and use PDF files in the Exstream Design and Production design environment. This method of using PDF designs in Exstream Design and Production applications is different from using other Design and Production features, such as dynamically importing PDF files, because it converts a PDF into documents, pages, and components that can be added to the Design and Production design database and reused in future applications.

The PDF Converter analyzes the PDF and creates a DXF file, which you can then import into Design and Production. You can specify how the converter handles some objects in the PDF, such as fonts and form fields, when the conversion occurs. Many PDF objects can be converted in such a way that they seamlessly integrate into the Design and Production environment. For example, if you convert a PDF that contains form fields, the converter can create variables for the form fields so that the PDF design can be integrated into an interactive application.

Most of the objects used in PDF format are supported in the Exstream Design and Production environment, and when they are converted and imported, the objects become editable Design and Production objects. However, since PDF files can originate from a variety of software, the PDF Converter might not be able to reproduce advanced design objects identically in the Design and Production environment.

This section discusses the PDF Converter version 3.0. For information on previous versions of the PDF Converter, see the PDF Converter guide that corresponds to the version you are using.

Before using the PDF Converter, make sure your system meets the following requirements:

- Exstream Design and Production, version 8.0.301 or later
- Windows 7 or later

The PDF Converter supports PDFs at version 1.6 or earlier.

This section discusses the following topics:

6.1 Licensing requirements for PDF Converter

When you first run PDF Converter, you are prompted to select a valid key for Design and Production. If you want to change your key after you have logged in to PDF Converter, select **Tools > Change Key**.

After you have selected a key, you might also be prompted for a license file, depending on whether you are using node-locked or floating licensing.

If you are using floating licensing and you have not previously opened Design and Production, PDF Converter will prompt you to open Design Manager at least once. If the workstation license file is not in the license folder, PDF Converter will also prompt you to select the workstation license file to copy to the license folder. If you want to use the borrow and return functionality for a floating PDF Converter license, you can borrow and return PDF Converter licenses by opening Design Manager and selecting **Tools > Licensing > Borrow floating licenses** or **Tools > Licensing > Return floating licenses**.

If you are using node-locked licensing, you must obtain a PDF Converter license in Design Manager by selecting **Tools > Licensing > License node-locked features**. Similarly, you can unlicense PDF Converter in Design Manager by selecting **Tools > Licensing > Unlicense node-locked features.**

Note: You must have Design Manager version 8.0.301 or later to manage some PDF Converter licensing activity. If you want to continue production work in a version of Exstream earlier than Exstream version 8.0.301, you can install a newer version of Exstream to manage licenses without affecting prior existing installations.

6.2 Planning your PDF conversion goals

The PDF Converter supports a variety of different conversion goals. For example, if your goal is to convert a PDF and achieve an exact replica of the original PDF in the Design and Production library, you can do that. On the other hand, if the appearance of text is flexible but you want to retain as much editing ability as possible, you can do that too. The PDF Converter supports each approach to conversion and gives you control over how different parts of the design are converted.

The following list describes the various conversion goals the converter can help you achieve and how the converter works in order to accomplish each goal:

- **I want an exact conversion**—The PDF Converter converts the entire PDF file as a bitmap image (the file is rasterized). There is no deviation in the placement of text, images, or other

objects in the converted file. In this scenario, the PDF Converter rasterizes the entire PDF file and each page is imported into Design and Production as a noneditable image. When you set up the converter to convert a file this way, you do not need to specify any other conversion settings, such as how fonts or text are handled during the conversion.

- **I want a conversion in which most text is editable**—The PDF Converter converts objects not supported in Design and Production to a bitmap image. The bitmap image appears in the background of the converted file so that other objects can be edited in the design environment. For example, if a paragraph contains fonts that are slanted 20 degrees, the slanted text is rasterized. All other text in the PDF (text supported in the design environment) is converted as editable text. When the DXF file is imported into Designer, the text is fully editable and you can apply any of the Designer formatting properties to it.

  If font metrics on the system used to create the PDF vary from those on the system where your design environment resides, the text placement might vary slightly within a paragraph. For example, the word that appears at the end of the line in the PDF file might appear at the beginning of the following line after the conversion.

- **I want a conversion in which all text is editable**—The PDF Converter converts all text in the PDF as editable text. When it encounters text in a format that is not supported in Design and Production, it converts the text into the closest possible supported match. For example, if a paragraph contains fonts that are slanted 20 degrees, the converter changes the slanted to text to appear at a supported angle. With this method, fonts with formatting that Exstream Design and Production does not support will appear differently after the conversion and the text placement might be slightly different than the placement in the original PDF. In addition, if font metrics on the system used to create the PDF vary from those on the system where your design environment resides, the text placement might vary slightly within a paragraph. For example, the word that appears at the end of the line in the PDF file might appear at the beginning of the following line after the conversion.

For information about ways you can help retain the original text placement during the conversion, see "Specifying How PDF Content is Converted" on page 89.

6.3 Optimizing fidelity during a conversion

Each time the converter must make a font substitution or font adjustment during the conversion, fidelity is lost. The converter lets you manually control font substitutions for individual fonts, or you can set up the converter to make adjustments for any unsupported text features (such as condensed fonts) automatically so that your text will remain editable after the conversion. Even slight differences in the font metrics of the original PDF fonts and the substituted fonts can affect the way text is justified and spaced on the page.

You can also optimize conversion fidelity by making sure that the fonts used in the original PDF are available on the computer where the converted file will be used. If the original fonts are not available, Designer will use its default substitute font, which might cause the converted file to appear differently. If the computer where you use the converted file has all of the original fonts available but you still see slight differences in the appearance of text, keep in mind that the font

metrics might be different in the system that created the original PDF and your original system. For example, the metrics for a font might vary greatly between operating systems, such as Windows and Macintosh. They might also vary between different versions of the same operating system. The easiest way to ensure that the font metrics are the same is to use the converted file on the same system on which the PDF was created. However, if you do not use the converted file on the same system, you can substitute fonts and use tools in Designer to adjust the appearance of text if needed. Another way to help protect fidelity during conversion is to make sure that the 14 standard PDF fonts are installed on the system where the DXF files will be used.

6.4 Object-specific conversion behavior

The PDF Converter treats some features used in PDF files uniquely. Other objects are not supported by the converter or as Design and Production objects. Before running the converter, make sure you are familiar with how the converter handles specific objects found in your PDF.

The following objects have unique conversion behaviors and might appear differently after the conversion:

- **Lines in the same object with different weights**—These objects have the same weight after the conversion. For example, suppose a table border uses a .5 line at the top of the table frame and a 1.0 line at the bottom of the table frame. After the conversion, all of the lines in the frame have a consistent weight of 1.0.
- **Text slanted more than 20 degrees or text that is compressed**—You can allow the converter to modify text in these formats so they can be converted as editable text by selecting the **Normalize all unsupported text features** check box on the **Outputs** tab of the converter. Text slanted less than 20 degrees is treated as italic.
- **Single phrases of rotated text**—This type of text is supported if the phrase is made up of a group of words with the same font, and it does not extend past one line. If the PDF contains multiple lines of rotated text, each line or phrase within the line is placed in its own text box.
- **Table cells not surrounded by a frame line (border) on all sides of the cell**—If a cell is not surrounded with a border, the table is drawn as separate objects, such as text boxes and lines.
- **Justified paragraphs that use soft returns**—Paragraphs using this type of alignment appear as left aligned in Designer after the conversion.
- **Images referenced as PDF form objects**—These types of images are not placed in the background bitmap image. Instead, each image is placed in the converted file as a JPEG with a quality setting of 100 and is stretched to fit the appropriate rectangle in Designer. Form images that are located on more than one page are automatically converted to Library components. The converter adds references to the image on each page where the image is used.
- **Hyperlinks**—The text that appears in a hyperlink on a page is converted, but the link is not

active. Use the design tools in Designer to re-create the hyperlink in the imported document. For more information on adding hyperlinks to a design, see *Designing Customer Communications* in the Exstream Design and Production documentation.

- **Hierarchical numbering**—Numbered lists with multiple levels, such as 1.1.2, are converted as text with hanging indents.
- **Subset fonts**—These types of fonts can produce unpredictable characters, depending on the way the font was set up in the original PDF. If you find that the subset fonts do not convert well, you can choose to rasterize those fonts using the **Fonts to be imaged** area on the **Fonts** tab of the PDF Converter.
- **Text with emulated bold formatting**—Bold text that is produced by offsetting the same character several times is converted as a single character with a bold style. Bold text that is converted this way is identified in the log file with a `bold(generated)` tag. If the font does not support a bold style, the text will appear as normal (not bold) text.
- **Ligatures**—Ligatures are a group of two or more characters that have been combined into a single character by the maker of a font. The PDF Converter automatically converts the following ligature characters into their individual component characters, allowing Designer to recognize words containing ligature characters during functions such as spell check, text search, and text rule comparisons.

  Ligature conversion

Unicode	Characters
0132	IJ
0133	ij
FB00	ff
FB01	fi
FB02	fl
FB03	ffi
FB04	ffl
FB05	ft
FB06	st

The following objects and features are automatically rasterized if you select **Use** or **All objects** from the **Background image use** drop-down list and appear in the background image in the converted file:

- Compressed fonts
- Text in a bezier clipping path

- Text slanted more than 15 degrees

6.5 Converting the PDF design to a DXF file

To process a PDF design so it can be used in a Design and Production application and stored in the design environment, you must complete the following tasks:

1. "Specifying the Conversion Settings" below
2. "Saving the conversion settings" on page 97
3. "Running the PDF Converter" on page 98
4. "Importing the generated DXF file" on page 98

You can also complete the following optional task as needed:

- "Cleaning up the imported pages in Designer" on page 100

After importing the DXF file, you should check the file and make any adjustments to the design as needed.

6.5.1 Specifying the Conversion Settings

Because PDF is a complex file format and PDFs can contain many types of objects, you must first provide information about the file and its content before running the converter. Next, the information you provide allows the converter to recreate the original design as closely as possible and gives you more control over the way specific types of objects appear after the conversion.

To convert designs from PDF format, complete the following tasks as needed:

- "Specifying the Files You Want to Convert" on the next page
- "Using a Configuration File to Automatically Load Conversion Settings" on page 88
- "Specifying the Directory Where the DXF File is Created" on page 88
- "Specifying How PDF Content is Converted" on page 89
- "Specifying How Fonts are Converted" on page 92
- "Specifying How PDF Fields are Converted" on page 94
- "Specifying How Variables are Created" on page 96

Specifying the Files You Want to Convert

The PDF Converter lets you convert PDF files one at a time so you can specify the settings needed for unique files. If you are converting several files that use the same conversion settings, you can also convert PDF files in batch mode.

Tip: If many files use the same conversion settings, consider creating and using a configuration file to help save time during the conversion process.

For information about configuration files, see "Using a Configuration File to Automatically Load Conversion Settings" on the next page.

To specify the files you want to convert:

1. Open the PDF Converter. By default, this program is installed in your Exstream directory.
2. Click the **Inputs** tab.
3. Using the buttons below the box, specify whether you want to convert one file or many files.

To	Do this
Convert one file	a. Click **Add Files**. The **Open** dialog box opens. b. Select the file you want to convert and click **Open**. The **Open** dialog box closes and the file you selected appears in the box on the **Input** tab.
Convert several files that are located in different folders	a. Click **Add Files**. The **Open** dialog box opens. b. Press and hold CTRL and select the files you want to convert. c. Click **Open**. The **Open** dialog box closes and the files you selected appear in the box on the **Input** tab.
Convert all the files that are located in a specific folder	a. Click **Add Folder**. The **Browse for Folder** dialog box opens. b. Select the folder that contains the files you want to convert and click **OK**. c. The converter analyzes the folder and adds all the PDF files within the folder to the **Input** tab.

4. If you are converting a large PDF file or if you want to convert only a subset of a PDF, enter a range or a comma delimited series of pages to convert in the **Page numbers to convert** box. For example, to convert pages 1 through 4, page 7, and pages 12 through 15, enter `1-`

`4, 7, 12-15` in the **Page numbers to convert** box. If you specified multiple PDF files to convert, the subset numbers you specify apply to all of the files listed in the **Inputs** box.

Using a Configuration File to Automatically Load Conversion Settings

If you are converting many legacy files that are similar and you can use the same conversion settings for all the files, you might choose to create and use a configuration file that provides the conversion settings automatically. A configuration file is created by saving the conversion settings you specified for a particular file.

For information about creating a configuration file, see "Saving the conversion settings" on page 97.

To use a configuration file to automatically load conversion settings:

1. At the bottom of the converter window, click **Configuration**.

   The **Open** dialog box opens.

2. Select the configuration file and click **Open**.

   The **Open** dialog box closes and the conversion settings saved in the configuration file are applied to the converter.

If you do not need to make adjustments to the settings, you can run the converter. However, if you need to make slight adjustments to accommodate the specific files you are converting, you can use the rest of the information in this section to set up the converter.

Specifying the Directory Where the DXF File is Created

When the PDF Converter runs, it creates a DXF file (or multiple files if you convert multiple PDFs). You must specify the directory in which you want the DXF file to be created. If you do not specify a directory, the DXF file is automatically placed in the same directory as the source PDF file.

To specify where the DXF file is created:

1. Click the **Outputs** tab.
2. Next to the **DXF output folder** box, click **Browse**.

   The **Browse for Folder** dialog box opens.

3. Select the folder where you want the DXF file to be placed.
4. Click **OK**.

   The **Browse for Folder** dialog box closes and the folder you selected appears in the **DXF output folder** box.

Specifying How PDF Content is Converted

The first step in specifying how PDF content is converted is to decide how you want the converted file to appear and behave when it is imported into Exstream Design and Production. For example, suppose your goal is simply to retain the PDF design and use it in applications without making any changes to its content. The choices you make when specifying how the content will be converted will be different in this case than if your goal is to edit and update the PDF design in Exstream Design and Production.

For information about the conversion methods and the conversion goals they can help you meet, see "Planning your PDF conversion goals" on page 82.

After deciding on your goals for the conversion, you can specify how the converter handles specific types of objects in the file. For example, you can specify languages based on the text direction, such as Arabic, which is a right-to-left text. When you select a language from the **Alphabet script** drop-down list, the DXF file specifies the direction of the text so that Design Manager can apply the text to the appropriate language layer. If the a language layer is not available for the specified language, Design Manager creates a new language object and new language layer. The DXF is then imported into the new language layer. Because PDF files can be created from a variety of different PDF generation software programs and each program creates PDFs differently, you can dictate the way the converter handles many of these objects to ensure the post-conversion design has the appearance and flexibility you require.

To specify how PDF content is converted:

1. Click the **Outputs** tab.
2. From the **Alphabet script** drop-down list, select one of the following options, depending on the type of text you want to convert:
   - **Latin**—Select **Latin** when the PDF you want to convert contains any type of left-to-right text. **Latin** is the default.
   - **Arabic**—Select **Arabic** when the PDF you want to convert contains Arabic (right-to-left text).
   - **Hebrew**—Select **Hebrew** when the PDF you want to convert contains Hebrew (right-to-left text).
3. To add an optional prefix to the document and page names, enter the prefix to be used for document and page names in the **Container page/document prefix** box. This prefix will appear in the objects's names in the Exstream Design and Production environment. For example, if you use an internal naming scheme for pages and documents, you can add a prefix so the converted files will match this scheme.
4. From the **Output encoding** drop-down list, select the type of encoding the PDF file uses:
   - **Windows-1252**—The file is single-byte.
   - **UTF-8**—The file is double-byte or you are converting a mix of double-byte and single-byte files. If you select to use UTF-8 encoding during the conversion, you might not be

able to see all of the double-byte characters.

- **UTF-16**—The file is double-byte or you are converting a mix of double-byte and single-byte files. If you select to use UTF-16 encoding during the conversion, the produced DXF file will be larger but you will be able to view and edit all of the double-byte characters (if they are not rasterized). If you selected **Arabic** or **Hebrew** from the **Alphabet script** drop-down list, select the **UTF-16** encoding option.

5. From the **Background image use** drop-down list, select one of the following options to control the fidelity of the conversion and how the text in the PDF is converted:
   - **All objects**—The entire PDF is rasterized. The DXF that is imported into Designer appears identical to the original PDF, but no text is editable.
   - **Use**—Parts of the PDF are rasterized. If you clear the **Normalize all unsupported text features** check box, text in unsupported formats, such as compressed text, is rasterized and appears on the background image. If you select the **Normalize all unsupported text features** check box, the converter modifies text in unsupported formats so it is converted as editable text.
   - **Ignore**—Text in unsupported formats is ignored and does not appear in the converted file. This option might be the best solution for your conversion if the PDF contains a watermark or other background text that you do not want to appear in the converted file.
6. If you selected **Use** or **Ignore** from the **Background image use** drop-down list, use the other options on the **Outputs** tab to control how specific objects in the file are converted.

To	Do this
Control when line drawings or shapes are rasterized	In the **Minimum lines/polys to draw as image** box, enter the fewest number of line objects in a localized area that the converter rasterizes. When the converter finds the specified number of lines in a localized area, it converts the lines to a single image. For example, if the original PDF creation software created a logo using line objects, you might choose to enter `100` so that all of the lines that make up the logo are rasterized into a single image. If you do not want the converter to rasterize any lines or polygons, regardless of how many exist, enter `0`. If the lines are dispersed throughout the page, however, they are not converted to an image, even if they exceed the number in the **Minimum lines/polys to draw as image** box. **Tip:** If you are not sure of the best value to enter for this option, try starting with a value of `100`. If you find that the converter still creates many line and polygon objects and the imported file is slow or cumbersome to work with in Designer, enter a lower value. The best value depends on your unique PDF file and your conversion goals.

To	Do this
Control how bullets in the PDF are converted (in single-byte files only)	If you selected **Windows-1252** from the **Output encoding** drop-down, select one of the following options from the **Bullets** drop-down list to specify how the converter treats bullets in the PDF file: • **Standard circles**—The converter converts bullets to the standard bullet for the PDF encoding (that is, Hex 95 for Windows-1252 encoding or Hex 2022 for UTF-8 or UTF-16 encoding). When the converter analyzes the PDF, any object that it recognizes as a bullet character is converted to the Windows standard. However, if the bullet character font is Symbol or Zapf Dingbat, then the converter uses the character as-is. • **As is**—The converter uses the bullet character exactly as it occurs in the PDF file. If the character used as the bullet is in a subsetted font and was mapped differently in the code page used to create the PDF, the bullet might appear differently after the conversion because the converter uses the standard code page during the conversion. If you selected **Arabic** or **Hebrew** from the **Alphabet script** drop-down list, you must select the **As-is** option. **Tip:** If a unique or unavailable font is used to create the original bullet, you can choose to rasterize that font to preserve the bullet's appearance. However, the paragraph will not be imported as a bulleted paragraph. Instead, the bullet characters will appear as part of the background image. If you make additional changes to the text, keep in mind that as text moves on the page, the bullets might be misaligned.
Control how numbered lists in the PDF are converted	From the **Numbering** drop-down list, select from the following options whether numbered lists in the PDF appear when imported into a document that contains other numbered lists: • **User set**—The converter honors the numbering found in the PDF. For example, if the DXF file is imported into a document where another numbered list exists previously, the converted numbered list retains its numbering, even if it conflicts with the existing numbering. • **Automatic**—The converter allows the numbered list to be merged when the DXF file is imported. For example, a numbered list that appears as `1-6` is updated to appear as `12-17` when the DXF file is imported and a previous numbered list in the document ends at `11`.
Specify how text formatting not supported by the converter is handled during the conversion	If you want the converter to modify unsupported text to appear like the surrounded text (for example, slanted text adjacent to non-slanted text is not slanted after the conversion), select the **Normalize all unsupported text features** check box. You can use the editing tools in Designer to apply formatting to this text after you import the DXF file into Design and Production. If you do not want the appearance of the text to change, clear the **Normalize all unsupported text features** check box. In this case, depending on your selection from the **Background image use** drop-down list, unsupported text is either ignored or rasterized and becomes part of the non-editable background image.

7. Use the **Honor line breaks** option to control how the converter handles line breaks in the PDF:

To	Do this
Allow different words to wrap at the end of lines and text spacing to vary slightly within a paragraph	Clear the **Honor line breaks** check box. If the PDF you are converting is comprised mainly of flowing paragraphs and your primary goal of the conversion is to leverage the content (rather than recreating the exact appearance of the original file), you should clear the **Honor line breaks** check box. If you clear the **Honor line breaks** check box, the same content will appear in the paragraphs, but the text spacing might not appear exactly the same and the words that wrap at the end of lines might be different. The converter will always insert margins on the right and/or left sides of the paragraph and/or text box or table cell to keep the text spacing as close to the original as possible. However, if the font metrics are inaccurate, you might need to make minor adjustments to the text box size, the text box margins, the paragraph margins, or the font size in Designer in order to recreate the original appearance. **Tip:** If you are converting a PDF with double-byte characters that do not have spaces between words, you should select the **Honor line breaks** check box.
Recreate the line breaks and text spacing exactly	Select the **Honor line breaks** check box. If you select the **Honor line breaks** check box, the converter inserts a soft return at the end of every line and can add left or right margins as needed to recreate the exact spacing of the text within the line. In most cases, you should select **Honor line breaks** because fewer margins make it easier to interact with the text in Designer. **Note:** The converter always inserts soft returns at the end of lines in centered paragraphs, regardless of your selection in the **Honor line breaks** check box.

Specifying How Fonts are Converted

Because PDF files can contain fonts that you might not use in the design environment, you can specify font substitutions for the converter to use during the conversion. For example, suppose most of the text in the original PDF was created using a font your organization does not use. You can substitute your corporate font for the unavailable font. Then, after the DXF file is imported, the text in the file appears in your corporate font.

In addition, if you want specific fonts to be rasterized, you can specify the fonts you want to be converted to an image automatically. For example, if a font your organization does not own was used to create an object (such as a logo or bullet), you can set up the converter to always rasterize that font when it is encountered in the file. That way, other text in the file can still be editable after the conversion, but you can retain the appearance of the unique objects. You might also choose to rasterize barcode fonts so the barcode's accuracy is maintained during the conversion.

To specify how fonts are converted:

1. Click the **Fonts** tab.
2. Use the properties on the **Fonts** tab to specify how fonts are converted. You can perform one or more of the following tasks to achieve the appearance you want:

To	Do this
Substitute fonts in a specified font family for fonts in a specified family **Tip:** You can use a PDF viewing program to see a list of all of the fonts used within a PDF file.	a. Below the **Font substitutions** box, click + . The **Font Mapping** dialog box opens. b. In the **PDF font** box, enter the name of the PDF font used in the PDF for which you want to substitute a font. c. In the **DXF font** box, enter the name of the font you want to substitute for the font used in the PDF. The DXF font should match the name of the font on the system that will be used to import the DXF into Design and Production. d. Click **OK**. The **Font Mapping** dialog box closes and the substitution information you provided appears in the **Font substitutions** box. e. Repeat step a through step d to add as many substitutions as needed. f. Use the  and  buttons to adjust the order of the font substitutions. Place more specific PDF font names, such as `Arial Narrow Bold`, before generic PDF font names, such as `Arial`. When the converter runs, it substitutes fonts in the order specified in the **Fonts substitutions** box. When the converter runs, it substitutes the font name you specified for the existing font name. For example, suppose the PDF uses the Arial font family (including Arial Black and Arial Narrow). If you specify Times New Roman as the substitution family, the converter will replace all instances of "Arial" with "Times New Roman." If the substitution creates an invalid font name (for example, Times New Roman Narrow), the default substitution font is used when the DXF file is imported into Exstream Design and Production. The log file displays information on each font substitution the converter makes.
Specify fonts you want to be rasterized	a. Below the **Font to be imaged** box, click + . The **Font to Image** dialog box opens. b. In the **PDF font (substring match)** box, enter the name of the font you want to be rasterized automatically. This name is compared to the names of the fonts found in the PDF file by substring matching. If the text is found anywhere in a font name, then the name is considered to be a match. c. Click **OK**. The **Font to Image** dialog box closes and the font information you provided appears in the **Fonts to be imaged** box. When the converter runs, it will rasterize the font you specified.

3. By default, the PDF Converter rounds some font sizes to make the placement of converted text match the appearance of the original PDF. However, you can control how the font sizes are rounded during the conversion by selecting the size rounding you want to be used from the **Font size rounding** drop-down list. If you select **Dynamic**, the converter rounds fonts under 10.75 point size by tenths of a point. If the font is over 10.75 point size, the converter rounds by half of a point.

Specifying How PDF Fields are Converted

If the PDF file you are converting contains AcroForm fields, the form fields can be converted to Design and Production objects and used in Exstream Design and Production applications, particularly interactive applications. The PDF Converter supports the following types of AcroForm fields:

- Text
- Drop-down
- Signature
- Push button
- Radio button
- Check box

If the field objects in a PDF are interactive, the converter can automatically create variables for them. When the DXF file is imported into Design and Production, the variable created for the form field is automatically associated with the appropriate field object. If the field is read-only, the converter creates the variable only.

To specify how PDF fields are converted:

1. Click the **Fields** tab.
2. From the **Push buttons style** drop-down list, select from the following options how the converter handles buttons:

To	Do this
Allow the converter to draw the button using individual objects, such as lines and text (The button might appear slightly differently than the button in the PDF. However, you can use the editing tools in Designerto adjust it.)	Select **Drawn**.
Allow the converter to copy the images for the button from the PDF (If the converter cannot access the associated images from the PDF or if they do not exist in the PDF, the converter defaults to the behavior of the **Drawn** setting.)	Select **Images**.

3. From the **Radio button style** drop-down list, select from the following options how the converter handles radio buttons:

To	Example	Do this
Allow the converter to substitute a three-dimensional radio button		Select **Drawn (3d radio)**.

To	Example	Do this
Allow the converter to substitute a two-dimensional radio button		Select **Drawn (flat radio)**.
Allow the converter to substitute the standard radio button used by Designer		Select **Standard images**.
Allow the converter to copy the images for the radio button from the PDF (If the converter cannot access the associated images from the PDF or if they do not exist in the PDF, the converter defaults to the behavior of the **Standard images** setting.)	n/a	Select **Images**.

4. From the **Checkbox style** drop-down list, select from the following options how the converter handles check boxes:

To	Example	Do this
Allow the converter to substitute a two-dimensional check box		Select **Drawn (2d checkbox)**.
Allow the converter to substitute a three-dimensional check box		Select **Drawn (3d checkbox)**.
Allow the converter to substitute a check box that uses a thin "X" when selected		Select **Drawn (X check)**.
Allow the converter to substitute a check box that uses a thick "X" when selected		Select **Drawn (wide X check)**.
Allow the converter to substitute a check box that becomes filled when selected		Select **Drawn (filled check)**.
Allow the converter to substitute the standard check box used by Designer		Select **Standard images**.
Allow the converter to copy the images for the check boxes from the PDF (If the converter cannot access the associated images from the PDF or if they do not exist in the PDF, the converter defaults to the behavior of the **Standard images** setting.) **Tip:** If the check boxes in the PDF are very small, selecting one of the drawn options typically produces better results than selecting **Standard images**.	n/a	Select **Images**.

5. If you want the converter to create a variable for each field, select the **Generate variables for fields** check box. When you select this check box, the converter automatically generates a string variable for all the fields (except for push buttons). If the field is named in the PDF, this name is used for the variable. If the field is not named in the PDF, the

converter generates a name automatically. By default, the variables for all fields are created as string variables. The converter gives them names in the following structure: `_[FieldType]#` (for example, `_Field4` or `_Choice7`). If you want more control over the way variables are created, use the **Variables** tab to customize their data type and format.

For information about customizing the data type and format of variables, see "Specifying How Variables are Created" below.

6. If you selected the **Generate variables for fields** check box, select the **Use field values as initial variable values** check box if you want the converter to use the field value as the initial value for the variable it creates for each field. For example, if a value was already specified for a radio button, select this check box to make the existing value the default value in the generated variable. If you clear the check box, the variables are not given initial values.
7. If you selected the **Generate variables for fields** check box, select the **Variables for check boxes are boolean** check box if you want the converter to use a Boolean variable for each check box field. If you clear the check box and do not provide data type information on the **Variables** tab, the converter creates all variables as string variables.

Specifying How Variables are Created

If the PDF you are converting contains form fields, the PDF Converter can automatically generate variables for each field. By default, these variables are string variables and take on the name of the field as specified in the PDF. You can override this default behavior by specifying search strings that, when encountered in the field names, indicate the variable should be given a data type and format that you specify.

You can also completely rename variables. For example, you might want to rename variables if fields in the PDF have the same name as built-in functions in Exstream Design and Production.

When you import the DXF file into Design and Production, the variables are added to the Design and Production variable library (the Data Dictionary) and you can use them in applications, just as you would use other variables.

To specify how variables are created:

1. Click the **Variables** tab.
2. In the appropriate variable type box, enter the search string the converter uses to identify a class of variables. For example, if a field is named `Invoice Total`, you can enter `Total` in the **Currency** box. When the converter runs, it analyzes the fields in the PDF. When it finds a field containing the text `Total`, the converter creates a currency variable.

   You can separate multiple search strings of the same type with a comma. The longest search strings you specify are searched first so that if the same text is used in a shorter search string, the most specific instance of the string is matched. For example, if you have entered the following strings: `Date12`, `Date1234`, `Date123`, and `Date1`, then the search strings are matched in the following order: `Date1234`, `Date123`, `Date12`, and finally `Date1`.

3. If you want the search string you entered to be used only when it appears at the end of the field name, select the **Ending only?** check box. For example, if you supply the word `Date` as the search string and the PDF contains both `InvoiceDate` and `DateOfBirth` fields, you can select the **Ending only?** check box to match only the `InvoiceDate` field name. Otherwise, the converter searches all the text in the field name for the search string.
4. In the **Default format** box, enter the digits that specify the default format of the variable. Depending on the data type of the variable, different digit options are valid.

   For information on the digits you can use to specify the formatting, see "Variable formatting codes" on page 101.
5. If you specified variables in the **Currency** or **Floating** boxes, enter in the **Default digits** box the number of digits to the right of the decimal place that are generated.
6. If you want to provide a prefix to be added to all the created variables, enter the prefix in the **Variable prefix** box. This prefix will appear in the variable's name in the Exstream Design and Production environment. For example, if you use an internal naming scheme for variables, you can add a prefix so the created variables will match this scheme.
7. If any of the names of the variables that will be created are the same as other objects in the Design and Production environment (particularly built-in function names), use the **Variable overrides** properties to provide an alternate name for the variable.
   a. Below the **Variable overrides** box, click + .

      The **Set Variable Mapping** dialog box opens.
   b. In the **PDF variable** box, enter the name of the field found in the PDF file.
   c. In the **DXF variable** box, enter the alternate name to be used in the DXF file.
   d. From the **Variable type** drop-down list, select the variable type.
   e. In the **Format** box, enter the digits that specify the default format of the variable. Depending on the data type of the variable, different digit options are valid.
   f. If you selected **Currency** or **Floating** from the **Variable type** drop-down list, you must specify the number of digits that are generated to the right of the decimal point. In the **Digits** box, enter this number.
   g. Click **OK**.
8. The **Set Variable Mapping** dialog box closes and the name substitution information you provided appears in the **Variable overrides** box.

6.5.2 Saving the conversion settings

Before running the converter, you can save the conversion settings you have defined. This optional step can save you time if you will convert similar files in the future.

To save the conversion settings:

1. Click **File > Save Configuration As**.

   The **Save As** dialog box opens.

2. Specify the file name and location.

3. Click **Save**.

   The **Save As** dialog box closes. The conversion settings are saved as a . p2d file that you can use in future conversions to automatically load settings for the conversion.

6.5.3 Running the PDF Converter

When you are ready to run the conversion, click **Convert**. The **Converting PDF to DXF** dialog box opens and displays information about the conversion process. Information about font substitution also appears here. When the conversion is complete, one DXF file for each PDF file will be located in the directory you specified, or in the directory where the source PDF is located if you did not specify a directory. You can then import the DXF file(s) into Design and Production. If you are converting multiple PDFs and one fails, the conversion ends and the remaining PDFs in the list are not converted. If this circumstance occurs, you can use the log file to identify the issue that caused the converter to fail and to identify which PDFs were converted before the conversion process stopped.

Tip: If you encounter performance issues during the conversion, or if you receive "Out of memory" errors in the conversion log, enter a subset of pages in the **Page numbers to convert** box on the **Inputs** tab to break the PDF into smaller, separate files.

6.6 Importing the generated DXF file

After you run the PDF Converter, you can import the generated DXF file into the Exstream Design and Production environment.

Before importing the DXF file, however, make sure that the fonts used in the DXF file are available in the Design and Production database. You can add fonts to Design and Production by completing one of the following actions:

- Manually add the fonts in Design Manager.
- On the **Design Manager** tab of the **System Configuration** dialog box, select the **Allow user fonts** check box. This option allows Design and Production to add the fonts found in the DXF file to the **Fonts** heading in the Library automatically. If you use this method to add fonts to the database, make sure that the user importing the DXF file into Design and Production has permissions to create fonts.

If you do not make accommodations for fonts to be added to Design Manager, the appearance of the imported file might be affected because of fonts substituted by Design and Production.

To import the DXF file into Design and Production:

1. In Design Manager, right-click the **Pages** heading and select **Import from DXF**.

   The **Import DXF** dialog box opens.

2. Navigate to the DXF file you want to import. You can select multiple files to import by holding down CTRL and selecting the files.

3. In the **Import** box, specify the Library folder into which you want to import the pages. If you will import multiple DXF files in which variables were created, choose a separate folder for each DXF file to avoid overwriting the variables.

   a. Click  .

      The **Folders** dialog box opens.

   b. Select the folder and click **OK**.

      The **Folders** dialog box closes and the folder you selected appears in the **Import** box.

4. In the **Page Type** list, select the Design and Production page settings you want to apply to the imported page. You can change this setting later, but you must specify a setting for the initial import. If the DXF file contains data on the paper size to use, this setting takes precedence over your selection on the **Page Type** drop-down list.

   a. Depending on the way your design standards are enforced, select either **Use specified Paper Type** or **Use specified Page Template**.

   b. In the list below the **Page Type** list, select either the paper type or page template you want to apply to the imported pages.

5. In the **Variable Import** list, select **Create missing variables**. This option lets Design and Production create variables as needed to maintain the appearance of the original design and make use of the variables created during the conversion. For example, if the PDF you converted contained blank spaces in it (for example, the phrase `an interest rate of _ _%`), the **Create missing variables** option lets Design and Production use a variable to reserve the space for the blank area. Otherwise, an improperly formatted paragraph containing tabs to create the spaces might be created in the file.

6. Click **Open**.

The **Log** dialog box opens and you receive informational messages about the import process. The files then appear in the Design Manager Library. If the DXF file contains multiple pages, Design and Production names the imported pages according to their order in the DXF file. For example, if you import a DXF file named `Test` that contains three pages, the file names are `TEST`, `TEST p2`, and `TEST p3`. You can rename the pages in Design Manager after they have been imported.

After you import the files, make sure to visually check the pages and make adjustments to objects as needed. If the converter created variables, you might need to refresh Designer to see the new variables.

6.7 Cleaning up the imported pages in Designer

If you focus your conversion choices to maintain text editability, you might need to make minor adjustments to the converted file after you import it. If adjustments are required, most of them will be small adjustments to text box sizes. In particular, if you cleared the **Honor line breaks** check box, the same content will appear in the paragraphs, but the text spacing might not appear exactly the same and the words that wrap at the end of lines might be different. If the font metrics differ between the font used to create the PDF and the font on the computer running Design and Production, you might need to make minor adjustments to the text box size, the text box margins, the paragraph margins, or the font size in Designer in order to recreate the original appearance.

The following list provides some tips for carrying out other clean-up tasks that might occasionally be necessary for some types of PDF files:

- If you find during the cleanup that objects appear to be missing from your design, they might be present but behind another object. To adjust the design so the objects are visible, right-click the object where the missing object should be and select **Order > Move to Back**.
- Since compressed fonts are not supported in Design and Production, PDF files that use compressed fonts might appear differently after conversion. One way to help recreate the original appearance is to use the tracking feature in Designer to adjust the character spacing. To use tracking to clean up previously-compressed fonts:
    a. In Designer, select the text that was compressed in the original PDF.
    b. Right-click the selected text and select **Font**.

       The **Select Font** dialog box opens.
    c. From the **Tracking** drop-down list, select **Condensed**.
    d. In the adjacent box, enter the point amount of the space you want to be reduced between characters. For example, to reduce the space between characters by one point, enter `1` in the box.
    e. Click **OK**.

       The **Select Font** dialog box closes and the character spacing is adjusted.

6.8 Variable formatting codes

The following table lists the formatting codes you can use to specify a default format for the created variables.

Formatting codes

Code	Abbreviation	Format
0	Custom	Do not use.
1	Keep blanks	Leaves spaces the way they appear in the string
2	Trim blanks	Includes only the characters and removes spaces found at the beginning and end of the string
3	Trim leading blanks	Removes all spaces found at the beginning of the string before the first character
4	Trim trailing blanks	Removes all spaces found after the string of characters
5	General number	Formats the number in Arabic numerals
6	Fixed decimal	Forces zeros to pad the value up to the number specified in the **Default digits** box
7	T or F	Returns T when the condition is true and F when the condition is false
8	Y or N	Returns Y when the conditions is true and N when the condition is false
9	1 or 0	Returns 1 when the condition is true and 0 when the condition is false
10	True or False	Returns True when the condition is true and False when the condition is false
11	Yes or No	Returns Yes when the condition is true and No when the condition is false
12	Locale	Formats the variable based on the chosen locale for the customer
13	Packed (left to right)	Reads from the left nibble of the first byte to the right nibble of the last byte
14	Packed (right to left)	Reads from the right nibble of the first byte to the right nibble of the last byte
15	Binint	Provides support for a number in binary format
16	Binfloat	Provides support for floating data in binary format
17	Bindouble	Provides support for a eight-byte binary data

Formatting codes, continued

Code	Abbreviation	Format
18	Currency	Uses the currency specified by the locale
19	Significant decimal	Inserts the least number of decimal values possible after the whole number, up to the decimal places specified in the **Default digits** box. Digits exceeding the specified value are rounded up and any zeros at the end of the decimal value are removed. For example, if you specify 2 in the **Default digits** box and the value of the variable is 5.303, it appears as 5.3. In the same scenario, if the value of the variable is 5.003, it appears as 5.
20	Intdecimal	Inserts a whole number, if the number does not have a fraction, and inserts the proper currency symbol for the locale. If the number has a fraction, it is formatted as a floating number. Zeros are kept, up to the number specified in the **Default digits** box.
21	(Lower, keep blanks)	Uses lowercase letters for text and leaves spaces the way they appear in the string
22	(Lower, keep trim blanks)	Uses lowercase letters for text and removes any leading and trailing spaces in the string
23	(Lower, trim leading)	Uses lowercase letters for text and removes any leading spaces found in the string
24	(Lower, trim trailing)	Uses lowercase letters for text and removes any trailing spaces found in the string
25	MixNoTrim	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Spaces are kept the way they appear in the string.
26	MixTrimBoth	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Any leading and trailing spaces in the string are removed.
27	MixTrimLead	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Any leading spaces found in the string are removed.
28	MixTrimTrail	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Any trailing spaces found in the string are removed.
29	Packed (left to left)	Reads from the left nibble of the first byte to the left nibble of the last byte
30	Packed (right to right)	Reads from the right nibble of the first byte to the right nibble of the last byte
31	Zoned	Supports decimal and currency formatted fields on a mainframe system

Formatting codes, continued

Code	Abbreviation	Format
32	Alpha Upper	Includes integer values as alpha characters. Each integer is associated with an uppercase letter of the alphabet. If a number greater than 26 is encountered, the alphabet starts over. For example, if 1 is encountered in the data file, it appears in the output as A. If 27 is encountered in the data file, it appears in the output as AA. (The first letter represents that the entire alphabet has been used. The second letter represents the remainder of the integer, minus 26).
33	Alpha Lower	Includes integer values as alpha characters. Each integer is associated with a lowercase letter of the alphabet. If a number greater than 26 is encountered, the alphabet starts over. For example, if 1 is encountered in the data file, it appears in the output as a. If 27 is encountered in the data file, it appears in the output as aa. (The first letter represents that the entire alphabet has been used. The second letter represents the remainder of the integer, minus 26).
34	Roman Upper	Includes integer values as uppercase Roman numerals
35	Roman Lower	Includes integer values as lowercase Roman numerals
36	Text Upper	Includes integer values as uppercase words. For example, if 1 is encountered, it appears in the output as ONE.
37	Text Mixed	Includes integer values as mixed case words. For example, if 1 is encountered in the data file, it appears in the output as One.
38	Text Lower	Includes integer values as lowercase words. For example, if 1 is encountered in the data file, it appears in the output as one.
39	Hex	Supports integer data written in hexadecimal format
40	COBOL Signed (trailing)	Supports EBCDIC formatted COBOL integer data with a plus character at the end
41	Binary unsigned byte	Provides support for a binary value stored in two bytes with no sign characters
42	Binary short	Provides support for a numeric value stored in two bytes where the first bit is a sign character
43	Percentage	Adds a percent (%) sign at the end of the number
44	Percentage x 100	Adds a percent (%) sign at the end of the number and multiplies by 100
45	Fixed decimal with currency	Inserts a fixed decimal point and the proper currency symbol for the locale. The locale does not have to be set (it can be left at the default).
46	Upper, keep blanks	Uses uppercase letters and leaves spaces the way they appear in the string

Formatting codes, continued

Code	Abbreviation	Format
47	Upper, trim blanks	Uses uppercase letters and removes any leading and trailing spaces found in the string
48	Upper, trim leading	Uses uppercase letters and removes any leading spaces found in the string
49	Upper, trim trailing	Uses uppercase letters for text and removes any trailing spaces found in the string
50	COBOL Signed (Leading)	Supports COBOL integer data with a plus character at the beginning
51	COBOL Sep-Leading Sign	Sets the data negative from the minus character at the beginning
52	COBOL Separate Sign	Sets the data negative from the minus character at the end
53	COBOL PHASE3 Signed	Provides COBOL support for a proprietary format
54	COBOL PHASE3 Signed Trailing	Provides COBOL support for a proprietary format
55	Unknown	Do not use.
56	Mix Ex NO Trim	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Spaces are kept the way they appear in the string, and any words found in an exceptions file are exempted.
57	Mix Ex Trim Both	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Any leading and trailing spaces in the string are removed, and any words found in an exceptions file are exempted.
58	Mix Ex Trim Lead	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Any leading spaces found in the string are removed, and any words found in an exceptions file are exempted.
59	Mix Ex Trim Trail	Uses uppercase letters for the first letter and lowercase for the rest of the letters. Any trailing spaces found in the string are removed, and any words found in an exceptions file are excluded.
60	Binary byte	Provides support for a numeric value stores in a single byte where the first bit is treated as a sign (value range is -127 to +127)
61	Binary unsigned byte	Provides support for numeric value stored in a single byte with no sign in the first bit
62	Absolute value	Includes the absolute value of numbers (to produce a negative number without a negative sign). For example, if you use this formatting option, -1 appears as 1.
63	PhoneParenDash	The data is a phone number with the area code in parentheses and a hyphen between the first three and last four digits of the number (for example, (999) 999-9999).

Formatting codes, continued

Code	Abbreviation	Format
64	PhoneParen	The data is a phone number with the area code in parentheses and a space between the first three and last four digits of the number (for example, (999) 999 9999).
65	PhoneDash	The data is a phone number with the area code without parentheses and a hyphen between the first three and last four digits of the number (for example, 999 999-9999).
66	Phone	The data is a phone number with the area code without parentheses and a space between the first three and last four digits of the number (for example, 999 999 9999).
67	Exceptions only, keep blanks	Exempts any words that are found in an exceptions file and leaves spaces the way they appear in the string
68	Exceptions only, trim blanks	Exempts any words that are found in an exceptions file and removes any leading and trailing spaces found in the string
69	Exceptions only, trim leading	Exempts any words that are found in an exceptions file and removes any leading spaces found in the string
70	Exceptions only, trim trailing	Exempts any words that are found in an exceptions file and removes any trailing spaces found in the string
71	Lower + exceptions, keep blanks	Uses lowercase letters, except those words found in an exceptions file, and leaves spaces the way they appear in the string
72	Lower + exceptions, trim blanks	Uses lowercase letters, except those words found in an exceptions file, and removes any leading and trailing spaces found in the string
73	Lower + Exceptions, trim leading	Uses lowercase letters, except those words found in an exceptions file, and removes any leading spaces found in the string
74	Lower + Exceptions, trim trailing	Uses lowercase letters, except those words found in an exceptions file, and removes any trailing spaces found in the string
75	Upper + exceptions, keep blanks	Uses uppercase letters, except for those found in an exceptions file, and leaves spaces the way they appear in the string
76	Upper + exceptions, trim blanks	Uses uppercase letters, except for those found in an exceptions file, and removes any leading and trailing spaces found in the string
77	Upper + exceptions, trim leading	Uses uppercase letters, except for those words found in an exceptions file, and removes any leading blanks found in the string
78	Upper + exceptions, trim trailing	Uses uppercase letters, except for those words found in the exceptions file, and removes any trailing spaces found in the string
79	Sentence, keep blanks	Capitalizes the first word of each sentence and leaves spaces the way they appear in the string

Formatting codes, continued

Code	Abbreviation	Format
80	Sentence, trim blanks	Capitalizes the first word of each sentence and removes any leading and trailing spaces found in the string
81	Sentence, trim leading	Capitalizes the first word of each sentence and removes any leading spaces found in the string
82	Sentence, trim trailing	Capitalizes the first word of each sentence and removes any trailing spaces found in the string
83	Sentence + exceptions, keep blanks	Capitalizes the first word of each sentence, except for those words found in an exceptions file, and leaves spaces the way they appear in the string
84	Sentence + exceptions, trim blanks	Capitalizes the first word of each sentence, except for those words found in an exceptions file, and removes any leading and trailing spaces found in the string
85	Sentence + exceptions, trim leading	Capitalizes the first word of each sentence, except for those words found in an exceptions file, and removes any leading spaces found in the string
86	Sentence + exceptions, trim trailing	Capitalizes the first word of each sentence, except for those words found in an exceptions file, and removes any trailing spaces found in the string
87	COBOL Signed (Trailing ASCII)	Supports ASCII formatted COBOL integer data with a plus (+) character at the end
88	Bin Quad Word	Binary 64-bit integer
89	Reverse, keep blanks	Reverses the data and leaves spaces the way they appear in the string
90	Reverse, trim blanks	Reverses the data and removes any leading and trailing spaces found in the string
91	Reverse, trim leading	Reverses the data and removes any leading spaces found in the string
92	Reverse, trim trailing	Reverses the data and removes any trailing spaces found in the string
193	RawContent	Do not use.
194	Base 64 Encoded	Do not use.
195	ASCII 8 5 Coded	Uses ASCII 85 encoded content
196	INT Decimal with currency	Inserts a whole number, if the number does not have a fraction, and inserts the proper currency symbol for the locale. If the number has a fraction, it is formatted as a floating number. Zeros are retained, up to the number specified in the **Default digits** box.
197	RTFContent	Do not use.

Formatting codes, continued

Code	Abbreviation	Format
198	PlainTextContent	Do not use.
100	MONTHDYYYY	Date appears as MonthDYYYY
101	MODYYYY	Date appears as MODYYYY (for example, Apr 6 2001)
102	MsDsYY	Date appears as m/d/yy (for example, 4/6/01)
103	MMsDDsYY	Date appears as mm/dd/yy (for example, 04/06/01)
104	MMsDD	Date appears as mm/dd (for example, 04/06)
105	MMdDDdyy	Date appears as mm-dd-yy (for example, 04-06-01)
106	MMDDYY	Date appears as mmddyy (for example, 040601)
107	MMDDYYYY	Date appear as mmddyyyy (for example, 04062001)
108	DMONTHYYYY	Date appears as DMONTHYYYY (for example, 6 April 2001)
109	DDMONTHYYYY	Date appears as DDMONTHYYYY (for example, 06 April 2001)
110	DMOYYYY	Date appears as DMOYYYY (for example, 6 Apr 2001)
111	DDMOYYYY	Date appears as DDMOYYYY (for example, 04 Apr 2001)
112	DDsMMsYY	Date appears as dd/mm/yy (for example, 06/04/01)
113	DDsMM	Date appears as dd/mm (for example, 06/04)
114	DDdMMdYY	Date appears as dd-mm-yy (for example, 06-04-01)
115	DDMMYY	Date appears as ddmmyy (for example, 060401)
116	DDMMYYYY	Date appears as ddmmyyyy (for example, 06042001)
117	Julian	Date appears as the Julian date
118	MsDsYYbHHcMM	Date appears as m/d/yy hh:nn (for example, 4/6/01 12:00)
119	MsDsYYYY	Date appears as m/d/yyyy (for example, 4/6/2001)
120	DsMsYYYY	Date appears as d/m/yyyy (for example, 6/4/2001)
121	YYYYDDMM	Date appears as yyyyddmm (for example, 20010604)
122	YYYYMMDD	Date appears as yyyymmdd (for example, 20010406)

Formatting codes, continued

Code	Abbreviation	Format
123	MMDDYYYYHHMM	Date appears as mmddyyyyhhnn (for example, 040620011200)
124	HHMM	Date appears as hhnn (1200)
125	HHcMM	Date appears as hh:nn (12:00)
126	DDMMYYYYHHM	Date appears as ddmmyyyyhhnn (for example, 060420011200)
127	MONTHDDYYYY	Date appears as month/dd/yyyy (for example, April 6, 2001)
128	MMYY	Date appears as mmyy (for example, 0401)
129	MMsDDsYYYY	Date appears as mm/dd/yyyy (for example, 04/06/2001)
130	MMdDDdYYYY	Date appears as mm-dd-yyyy (for example, 04-06-2001)
131	PKMMDDYYYY	Date appears as packed MMDDYYYY
132	PKDDMMYYYY	Date appears as packed DDMMYYYY
133	PKYYYYMMDD	Date appears as packed YYYYMMDD
134	HHMMSSSS	Date appears as hhnnssss (for example, ss + ss/100)
135	HHMMSS	Date appears as hhnnss (for example, 123201)
136	HHcMMcSS	Date appears as hh:nnsss (for example, 12:32:01)
137	MOD	Date appears as MOD (for example, Apr 6)
138	MODD	Date appears as MODD (for example, Apr 06)
139	DMO	Date appears as DMO (for example, 6 Apr)
140	DDMO	Date appears as DDMO (for example, 06 Apr)
141	HHMMa	Date appears as hhnna or hhnnp for am/pm, respectively (for example, 1201a)
142	CSMMDDYYYY	Date appears as signed COBOL in the format MMDDYYYY
143	CSDDMMYYYYY	Date appears as signed COBOL in the format DDMMYYYY
144	CSYYYYMMDD	Date appears as signed COBOL in the format YYYYMMDD
145	MsYY	Date appears as m/yy (for example, 4/01)
146	MMsYY	Date appears as mm/yy (for example, 04/01)

Formatting codes, continued

Code	Abbreviation	Format
147	MsYYYY	Date appears as m/yyyy (for example, 4/2001)
148	MMsYYY	Date appears as mm/yyyy (for example, 04/2001)
149	DDsMMsYYYY	Date appears as dd/mm/yyyy (for example, 06/04/2001)
150	PKMMDDYYYYR	Date appears in packed right to left format as MMDDYYYY
151	PKDDMMYYYYR	Date appears in packed right to left format as DDMMYYYY
152	PKYYYYMMDDR	Date appears in packed right to left format as YYYYMMDD
153	System Default	Date appears in the default format
154	YYYYsMsD	Date appears as yyyy/m/d (for example, 2001/4/6)
155	YYYYsMMsDD	Date appears as yyyy/mm/dd (for example, 2001/04/06)
156	YYYYdMdD	Date appears as yyyy-m-d (for example, 2001-4-6)
157	YYYYdMMdDD	Date appears as yyyy-mm-dd (for example, 2001-04-06)
158	ODBC Date	Date appears as ODBC date in yyyy-mm-dd format
159	ODBC Time	Date appears as ODBC time in hh:nn:ss format
160	ODBC Timestamp	Date appears as ODBC timestamp in yyyy-mm-dd hh:nn:ss format
161	ODBC Date Binary	Date appears as ODBC date in binary format
162	ODBC Time Binary	Date appears as ODBC time in binary format
163	ODBC Timestamp Binary	Date appears as ODBC timestamp in binary format
164	DDpMONTHYYYY	Date appears as DD.monthYYYY (for example, 06.April 2001)
165	DpMONTHYYYY	Date appears as dpMONTHYYYY (for example, 6.April 2001)
166	DDpMMpYYYY	Date appears as dd.mm.yyyy (for example, 06.04.2001)
167	DpMpYYYY	Date appears as d.m.yyyy (for example, 6.4.2001)
168	MpDpYYYY	Date appears as m.d.yyyy (for example, 4.6.2001)
169	MMpDDpYYYY	Date appears as mm.dd.yyyy (for example, 04.06.2001)
170	MpDpYY	Date appears as m.d.yy (for example, 4.6.01)

Formatting codes, continued

Code	Abbreviation	Format
171	DpMpYY	Date appears as d.m.yy (for example, 6.4.01)
199	SOAPDate	Date appears as SOAP date in yyyy-mm-dd format
200	SOAPTime	Date appears as SOAP time in hh:nn:ss format
201	SOAPTimestamp	Date appears as SOAP Timestamp in yyyy-mm-dd hh:nn:ss format
202	YYYYpMMpDD	Date appears as YYYY.MM.DD (for example, 2001.04.06)
203	YYYYpMpD	Date appears as YYYY.M.D (for example, 2001.4.6).
204	YYYYpMONTHDD	Date appears as YYYY.MonthDD (for example, 2001.April 06)

Unicode formatting codes

Code	Abbreviation	Format
172	GengoYYsMMsDD	Date appears in Unicode format as Hyy/mm/dd
173	GengoYYdMMdDD	Date appears in Unicode format as Hyy-dd-dd
174	GengoYsMsD	Date appears in Unicode format as Hy/m/d
175	GegnoYdMdD	Date appears in Unicode format as Hy-m-d
176	GengoxxYYxMMxDDx	Date appears as 平成 yy 年 mm 月 dd 日
177	GengoxxYxMxD	Date appears as 平成 y 年 m 月 d 日
178	GengoWidexxYYxMMxDDx	Do not use.
179	GengoWidexxYxMxD	Do not use.
180	GengoSimplexxYxMxD	Date appears as 平成九年九月九日
181	GengoCommonxxYxMxD	Date appears as 平成十年十月十日
186	LastGengo = GengoCommonxxYxMxD + 5	Do not use.
187	JapYYYYxMMxDDx	Date appears as yyyy 年 mm 月 dd 日
188	JapYYYYxMxDx	Date appears as yyyy 年 m 月 d 日

Unicode formatting codes, continued

Code	Abbreviation	Format
189	JapMMxDDx	Date appears as mm 月 dd 日
190	JapMxDx	Date appears as m 月 d 日
191	JapWideYYYYxMMxDDx	Do not use.
192	JapWideYYYYxMxDx	Do not use.
193	JapWideMMxDDx	Do not use.
194	JapWideMxDx	Do not use.
195	ChineseYsYsYsYxMxsDx	Date appears as 九九九九年九月九日
196	ChineseYsYsYsYxcMxcDx	Date appears as 九九九九年十月十日
197	ChineseMxsDx	Date appears as 九月九日
198	ChineseMxcDx	Date appears as 十月十日

Chapter 7: Importing Quark designs

The Quark Converter lets you import QuarkXPress files into the Exstream Design and Production environment. The Quark Converter analyzes a specified Quark file and creates a DXF file, which you can then import into Design and Production.

This section discusses the Quark Converter version 1.1.

Most Quark design objects are supported in the Design and Production environment, and when they are converted, become editable objects. For example, Quark line styles, colors, and shades are supported fully. Therefore, when a page is converted and imported into Design and Production, you can see the properties of these lines and change them just as you would for pages created in Design and Production.

Some features used in Quark files are not supported as Design and Production objects. Designs that use unsupported features can be converted as view-only bitmaps. For example, bezier lines are not supported by the Quark Converter format. Therefore, during the conversion, you can select to convert the lines as a bitmap. When the converted page is imported into the design environment, the bezier line will be visible, but you can interact with it only as an image (you can change its location, etc. but not its appearance). If you choose not to convert unsupported objects to bitmap format, those objects do not appear on the converted page. You can also choose to convert entire pages to bitmap format.

Before using the Quark Converter, make sure your system meets the following requirements:

System	Requirements
Windows	• Windows 7 • QuarkXPress 6.1, 6.5, or 7.2 • Ghostscript 9.0 or later **Note:** You must install Ghostscript separately from Exstream or the Quark Converter and then add the location of the Ghostscript executable file to your system path in Windows. Ghostscript is a commercially available PostScript and PDF conversion and rendering tool. For information about installing Ghostscript, see *Installation and Upgrade Information* in the Exstream Design and Production documentation. • Apache Xerces 2.7.0
Macintosh	• Mac OS X 10.3 or later • QuarkXPress 6.1, 6.5, or 7.0-7.2 • Ghostscript 9.0 or later (included in the Macintosh installer) • Apache Xerces 2.7.0 (included in the Macintosh installer)

This section discusses the following topics:

7.1 Supported Quark features

This section describes the Quark features supported by the Design and Production environment. If your design uses a feature not listed here, you can use one of the following solutions to make the conversion process simpler:

- Adjust the design so it uses only supported features.
- Convert the area or page as a bitmap. The object or feature will then not be editable in the Design and Production environment.

Any features that are not supported and cannot be converted to bitmap do not appear on the converted page.

Before converting a Quark file, make sure you familiarize yourself with the following information about how objects in the Quark file will appear after conversion. The following tables list Quark features and objects that are supported by the converter. The tables are organized by the following functions:

- Box features
- Color types
- Justification settings
- Line features
- Shape features
- Text features
- Miscellaneous features

Features that are identified as "fully supported" are imported as editable objects in the Design and Production environment.

Supported box features

Feature	Notes about conversion behavior
Anchored boxes	Design and Production does not support anchored boxes. However, the Quark Converter can convert them to bitmaps (if you select the **Anchored Boxes** check box on the **Convert to Bitmap if checked** area during the conversion).
Angle	Fully supported
Background color	Fully supported
Background shade	Fully supported
Bezier picture boxes	Appears on the converted page only if you convert the entire page as bitmap
Border gap color	Supported if you select to convert the feature as bitmap (by selecting the **Border gap color** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Border style	Supported if you select to convert the feature as bitmap (by selecting the **Border style** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Frame back color	Appears on the converted page only if you convert the entire page as bitmap
Frame back shade	Appears on the converted page only if you convert the entire page as bitmap
Frame color	Fully supported
Frame shade	Fully supported
Frame style	Fully supported
Frame width	Fully supported
Shape	Supported if you select to convert the feature as bitmap (by selecting the **Shape** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Text angle	Supported if you select to convert the feature as bitmap (by selecting the **Text Angle** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Text skew	Supported if you select to convert the feature as bitmap (by selecting the **Text skew** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.

Supported color types

Feature	Notes about conversion behavior
DIC	Converted to RGB colors
Focoltone	Converted to RGB colors
Hexachrome	Converted to RGB colors

Supported color types, continued

Feature	Notes about conversion behavior
HSB	Converted to RGB colors
LAB	Converted to RGB colors
Multi-Ink	Converted to RGB colors
PANTONE	Converted to CMYK colors
TOYO	Converted to RGB colors
TRUMATCH	Converted to RGB colors

Supported justification settings

Feature	Notes about conversion behavior
Bottom	Fully supported
Centered	Fully supported
Justified	Fully supported
Top	Fully supported

Supported line features

Feature	Notes about conversion behavior
Arrowheads	Appears on the converted page only if you convert the entire page as bitmap
Bezier lines	Appears on the converted page only if you convert the entire page as bitmap
Clipping	Appears on the converted page only if you convert the entire page as bitmap
Freehand	Fully supported
Line	Fully supported
Line back color	Appears on the converted page only if you convert the entire page as bitmap
Line back shade	Appears on the converted page only if you convert the entire page as bitmap
Line color	Fully supported
Line shade	Fully supported
Line style	Fully supported
Line width	Fully supported

Supported line features, continued

Feature	Notes about conversion behavior
Orthogonal	Fully supported

Supported shape features

Feature	Notes about conversion behavior
Beveled rectangle	Fully supported
Concave rectangle	Fully supported
Convex rectangle	Fully supported
Corner radius or rectangle	Fully supported
Oval	Fully supported
Rectangle	Fully supported
Runaround both sides	Appears on the converted page only if you convert the entire page as bitmap
Shape	Fully supported
Skew	Appears on the converted page only if you convert the entire page as bitmap
Spline	Appears on the converted page only if you convert the entire page as bitmap
Text slant (with object)	Appears on the converted page only if you convert the entire page as bitmap
Text wrap	Appears on the converted page only if you convert the entire page as bitmap
Transparency	Appears on the converted page only if you convert the entire page as bitmap

You can convert shapes not listed here to bitmap format by selecting the **Shapes** check box on the **Export to OpenText Exstream** dialog box.

Supported text features

Feature	Notes about conversion behavior
All caps	Supported if you select to convert the feature as bitmap (by selecting the **All Caps** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Baseline shift	Supported if you select to convert the feature as bitmap (by selecting the **Baseline shift** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Drop caps	Supported if you select to convert the feature as bitmap (by selecting the **Drop Caps** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.

Supported text features, continued

Feature	Notes about conversion behavior
Em dash (breaking and nonbreaking dash)	Fully supported
En dash	Supported if you select to convert the feature as bitmap (by selecting the **Special Dashes** check box on the **Convert to Bitmap if checked** area during the conversion).
Hyphen	Fully supported
Indent here	Honored on the converted page only if you convert the entire page as bitmap
Kerning	Supported if you select to convert the feature as bitmap (by selecting the **Kerning** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Outline	Supported if you select to convert the feature as bitmap (by selecting the **Outline** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Paragraph alignment settings	The following paragraph alignment settings are supported: • Baseline shifts • Centered • Indents • Justified • Keep lines together • Keep with next paragraph • Leading (line spacing) • Left • Right The forced justify paragraph alignment setting is supported if you select to convert it as a bitmap.
Paragraph lines	Supported if you select to convert the feature as bitmap (by selecting the **Paragraph Lines** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Paragraph rules	Converted to bitmap
Scale	Supported if you select to convert the feature as bitmap (by selecting the **Scale** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Shadow	Supported if you select to convert the feature as bitmap (by selecting the **Shadow** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Small caps	Supported if you select to convert the feature as bitmap (by selecting the **Small Caps** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.
Soft hyphen	Fully supported

Supported text features, continued

Feature	Notes about conversion behavior
Spaces (em)	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box on the **Convert to Bitmap if checked** area during the conversion).
Spaces (en)	Supported if you select to convert the feature as bitmap (by selecting the **Special Spaces** check box on the **Convert to Bitmap if checked** area during the conversion).
Spaces (punctuation)	Fully supported
Special characters	Supported if you select to convert them as bitmap (by selecting the **Special Characters** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, they are ignored.
Tab setting	The following tab alignment settings are supported: • Align on • Center • Comma • Decimal • Fill character • Left • Position (from left) • Right **Note:** Designer supports a single tab fill character while Quark supports up to two tab fill characters. Therefore, the Quark Converter can support the conversion of a single table fill character only. If you want to convert a design with two tab fill characters, select the **Tab Fill** check box on the **Export to OpenText Exstream** dialog box The following features are supported if you select to convert them as bitmap: • Align on tabs • Tab fill
Unicode characters	Supported if you select to convert them as bitmap (by selecting the **Special Characters** check box in the **Convert to Bitmap if checked** area during the conversion). Otherwise, they are ignored.
Variables	Fully supported. To include variables in a converted document, surround the variable name in Quark with brackets (for example, `<CustomerName>`).
Word underline	Supported if you select to convert the feature as bitmap (by selecting the **Word underline** check box on the **Convert to Bitmap if checked** area during the conversion). Otherwise, it is ignored.

Miscellaneous features

Feature	Notes about conversion behavior
Embedded objects	These types of objects might not convert correctly and are often separated during conversion.

Miscellaneous features, continued

Feature	Notes about conversion behavior
EPS Images with overlapping text	After the conversion, the overlapping text is treated as an uneditable part of the image.
Grouped objects	Objects grouped in Quark convert to individual ungrouped objects. Each object from the original grouped design is subject to individual conversion support and feature restrictions. However, you might choose to convert all the objects in the group to bitmap format, if some of the grouped objects will not convert correctly. To convert all the objects to a single bitmap image, select the **Grouped** check box on the **Export to OpenText Exstream** dialog box.
Objects that extend past the design area	These types of objects might not convert correctly.

7.2 Converting the Quark design to a DXF File

You use the Quark Converter to convert files from Quark format into DXF format. Before you run the Quark Converter, you must resolve any issues that appear with a Missing or Modified status on the Usage dialog box in Quark. Objects with either a Missing or Modified status can cause errors in the conversion and will not appear when the DXF file is imported into Design and Production. In addition, you must install any fonts used in the Quark file on the computer where the DXF file will be imported.

There are three ways you can convert files from Quark format. The method you choose to use depends on how you want to use the design.

- **Convert the entire file into a bitmap**—The entire Quark page is converted into a TIFF image, which you can then import into Designer. All Quark objects appear in the TIFF; however, you cannot make any changes to the converted file. You might choose to use this method if you do not need to make any changes to the design after it is imported and if you want the converted file to appear exactly the same as the Quark file.
- **Convert specific features to bitmap**—You specify which features are converted to bitmap. This method gives you the most control over the conversion process, and will allow you to make changes to most design objects in Designer. You might choose to use this method if you want to retain as much of the original appearance of the design as possible, while still having the ability to make changes to the design in Designer. If you choose to use this method, make sure you are familiar with the supported objects and how they are treated by the Quark converter.
- **Convert all supported features and ignore unsupported features**—The converter converts all supported features. Any features that are not natively supported in the Design and Production format or that must be converted to bitmap format in order to appear in the converted file are ignored. You might choose to use this method if your Quark design does

not contain many unsupported objects, if it will be easy to recreate the ignored objects in Designer, or if you need to make extensive changes to the page in Designer in the future.

To convert the file from Quark Format:

1. In Quark, from the **File** menu, select **Export to Exstream**.

   The **Export to OpenText Exstream** dialog box opens.

   > **Tip:** If you use a configuration file to automatically load pre-defined conversion settings, click **Load**, select the configuration file, and skip to step 3.

2. In the **Convert to Bitmap Settings** area, specify how you want the Quark Converter to treat objects that are not natively supported by Design and Production.

To	Do this
Convert the entire file into a bitmap	a. Select **All**. b. In the **Convert to Bitmap Device** area, select the settings for the TIFF that will be created. **Tip:** Generally, the resolution you select should match the dots per inch (dpi) of the target output device. Also, keep in mind that high dpi resolution can result in very large image file sizes.
Convert specific features to bitmap format	a. Select **Some**. b. In the **Convert to Bitmap if checked** area, select the objects that you want the converter to convert to bitmap format. c. At the bottom of the **Convert to Bitmap if checked** area, select the **Embed Images in DXF** check box if you want images in the InDesign page to be embedded in the converted file. You must select this check box if you run Quark and the design environment on different computers. However, if Quark and Design and Production are located on the same computer, you can clear this check box to improve the speed of the conversion process. After the conversion, the converted file must be used on the computer that contains Quark and the image files in order for the images to appear.
Convert all supported features and ignore unsupported features	a. Select **None**. b. At the bottom of the **Convert to Bitmap if checked** area, select the **Embed Images in DXF** check box if you want images in the Quark page to be embedded in the converted file. You must select this check box if you run InDesign and the design environment on different computers. However, if Quark and Design and Production are located on the same computer, you can clear this check box to improve the speed of the conversion process. After the conversion, the converted file must be used on the computer that contains Quark and the image files in order for the images to appear.

> **Tip:** If you want to save the conversion settings you just defined, click **Save** in the **Convert to Bitmap Settings** area. The conversion settings are saved as an INI file that you can use in future conversions to automatically load settings for the converter. This feature allows you to quickly load pre-defined settings for each conversion you must complete.

3. In the **File name** box, click  to specify the location and file name of the DXF file the InDesign Converter will create. If you use a Macintosh, you cannot enter a name manually in the **File name** box; you must click  to specify the location and file name. You will select this file when you import the DXF into Design and Production.
4. Click **Export**.

   The conversion process begins. The **Log** dialog box opens and provides details about the conversion. You can use the information it provides for troubleshooting.

When the conversion is complete, one DXF file for each Quark file will be located in the directory you specified. You can then import the DXF file or files into Exstream Design and Production.

For more information on DXF files, see "Understanding the DXF format" on page 7.

7.3 Importing the generated DXF file

After you run the Quark Converter, import the DXF file into Design and Production.

To import the DXF file:

1. In Design Manager, right-click the **Pages** heading and select **Import from DXF**.

   The **Import DXF** dialog box opens.
2. Navigate to the DXF file you want to import. You can select multiple files to import by holding down CTRL and selecting the files.
3. In the **Import** box, specify the Library folder into which you want to import the pages.

   a. Click  .

      The **Folders** dialog box opens.

   b. Select the folder and click **OK**.

      The **Folders** dialog box closes and the folder you selected appears in the **Import** box.
4. In the **Page Type** list, select the Exstream Design and Production page settings you want to apply to the imported page. You can change this setting later, but you must specify a

setting option for the initial import.

a. Depending on the way your design standards are enforced, select either **Use specified Paper Type** or **Use specified Page Template**.

b. In the list below the **Page Type** list, select either the paper type or page template you want to apply to the imported pages.

5. In the **Variable Import** list, select how variables in the converted document are handled.

To	Do this
Ignore any variables in the original document	Select **Ignore variables**.
Honor only original variables in the file that exist in the Exstream Design and Production Library	a. Select **Map to existing variables only**. b. In the **Variable Folder** box, specify the Design Manager folder that contains the variables. i. Click  . The **Folders** dialog box opens. ii. Select the folder and click **OK**. The **Folders** dialog box closes and the folder you selected appears in the **Variable Folder** box.
Honor all variables from the original file and create new Exstream Design and Production variables for those that do not currently exist in the Library	a. Select **Create missing variables**. b. In the **Variable Folder** box, specify the Design Manager folder that contains the variables. i. Click  . The **Folders** dialog box opens. ii. Select the folder and click **OK**. The **Folders** dialog box closes and the folder you selected appears in the **Variable Folder** box.

6. Click **Open**.

The **Log** dialog box displays informational messages about the import process. The files then appear in the Design Manager Library. If the DXF file contains multiple pages, Exstream Design and Production names the imported pages according to their order in the DXF file. For example, if you import a DXF file named `Test` that contains three pages, the file names are `TEST.DXF.page.1`, `TEST.DXF.page.2`, and `TEST.DXF.page.3`. You can rename the pages in Design Manager after they have been imported.

After you import the files, make sure to visually check the pages and make adjustments to objects as needed.