



Designing Forms for LiveCycle® Output ES2

November 2009

Adobe® LiveCycle® Designer ES2

Version 9

© 2009 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle® Designer ES2 (9.0) Designing Forms for LiveCycle Output ES for Microsoft® Windows®
Edition 2.0, November 2009

This reference document is licensed for use under the terms of the Creative Commons Attribution Non-Commercial 3.0 License. This License allows users to copy, distribute, and transmit the document for noncommercial purposes only so long as (1) proper attribution to Adobe is given as the owner of the document; and (2) any reuse or distribution of the document contains a notice that use of the document is governed by these terms. The best way to provide notice is to include the following link. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>.

Adobe, the Adobe logo, Acrobat, Adobe Reader, LiveCycle, Minion, Myriad, and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Microsoft and Windows are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

Contents

About This Document

Who should read this document?	1
Additional information	1

Setting Up Forms for Printing

Specify the master page placement in a page set	3
Use page sets to control single-sided and double-sided printing in a form	5
Specify which side of the paper the master page is applied to	6
Insert a blank page after the first page in a form intended for double-sided printing	7
Preview and test printing forms	8
Specify the device configuration file	9
About Output-IVS	9

Font Handling

Specifying printer-resident fonts in Designer ES2	10
Font mapping	11

Paper Tray Handling

Designing for Label Printers

Supported label printer features	13
Testing the label printer's standard fonts	14
Supported barcodes	14
Using RFID barcode objects	15

Considerations for Batch Data Usage

Incremental loading of multiple records in a batch	16
Designing for multiple records	17
Scripting methods for use when processing batch data sets	18

About This Document

This document explains what to consider when using Adobe® LiveCycle® Designer ES2 to design forms for Adobe LiveCycle Output ES2. It includes the following information:

- Setting up forms for printing
- Choosing fonts
- Paper tray handling
- Configuring forms for supported able printers
- Configuring output based on variable data

Who should read this document?

This document is intended for form authors and form developers who create forms that are rendered by using LiveCycle Output ES2. Together with the information from your Adobe LiveCycle ES2 and network printer administrators, this document describes things to keep in mind when planning your forms.

Additional information

The resources in this table can help you learn more about LiveCycle Designer ES2.

For information about	See
Creating forms using Designer ES2	LiveCycle Designer ES2 Help
Output ES2	LiveCycle ES2 Services
Editing XDC files	LiveCycle ES2 XDC Editor Help
Using a service in a process map	LiveCycle Workbench ES2 Help

Setting Up Forms for Printing

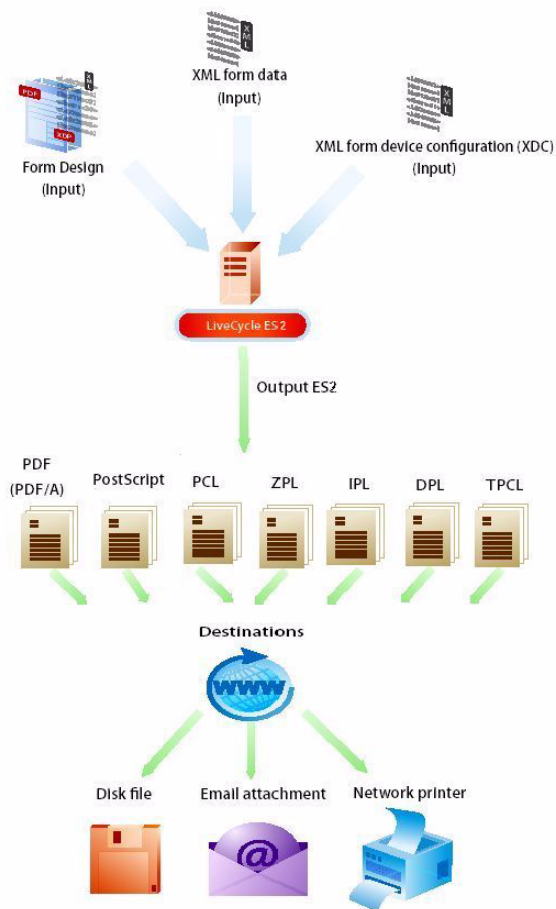
LiveCycle Output ES2 merges XML form data with a form design created in LiveCycle Designer ES2 to produce a formatted document. Documents can be sent to any of the following devices:

- Disk file
- Network printer

When used as part of a process, the document can be sent to an email recipient as a file attachment

As shown in the following illustration, Output ES2 lets you output documents to the following formats and printers:

- Adobe PDF (including PDF/A)
- Adobe PostScript®
- Printer Control Language (PCL)
- Zebra Printer Language (ZPL) for Zebra label printers
- Intermec Printer Language (IPL) for Intermec label printers
- Datamex Printer Language (DPL) for Datamex label printers
- TEC Printer Command Language (TPCL) for Toshiba TEC label printers



Specify the master page placement in a page set

Before LiveCycle Designer ES 8.0, you managed page sets by controlling the number of occurrences of each page in the form. This process meant you controlled repetitive page rendering by specifying one of these options in the Object palette for the selected page set or master page:

- Restrict Page Occurrence
- Min
- Count
- Max

For compatibility reasons, this functionality is still available.

In Designer ES 8.0 or later, use master pages in a page set to control page placement and the number of page occurrences in a form.

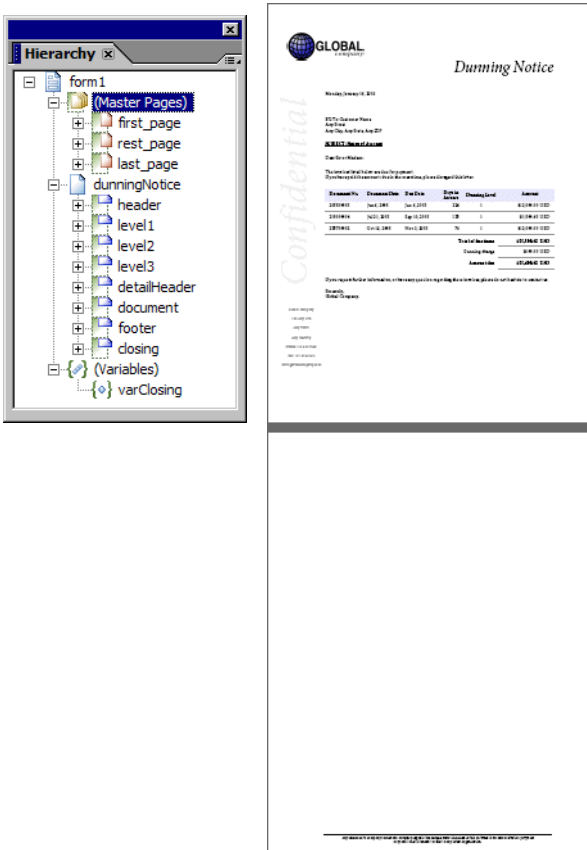
Form designs contain at least one page set. For forms that are intended for double-sided printing, specify where the master page is used in the page set. For example, specify in the first printed page, the last printed page, or the printed pages in between.

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

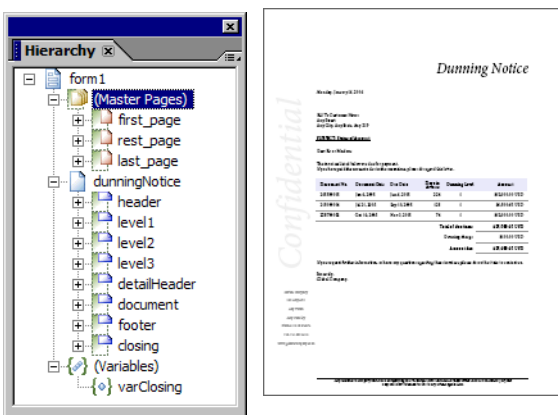
For form designs that contain more than one page set, specify a break in a subform for printing to transition from one page set to the next.

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

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



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



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

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

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

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

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

In MasterPage2, the paper that the preprint form is printed on contains the company logo and contact information. This master page leaves room for that information outside the content area. Following next is a small content area to receive employee-specific information. The content area occurs once for the preprint form and therefore is assigned as the first page in the second page set. The second page set requires double-sided printing.

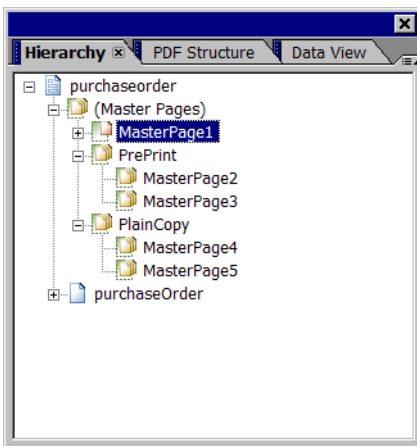
MasterPage3 has a larger content area, possibly covering the whole page, to receive more information. It is used for the second and subsequent pages of the preprint form. MasterPage3 is assigned the Rest of Pages placement option in the second page set. MasterPage3 also includes a page break in a subform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

MasterPage4 has the company logo and contact information first, outside the content area. Following next is a small content area to receive employee-specific information. It occurs once for the plain copy form and therefore is assigned as the first page in the third page set. The third page set also requires double-sided printing.

MasterPage5 has a larger content area, possibly covering the whole page, to receive more information. MasterPage5 is used for the second and subsequent pages of the plain copy form. It occurs from 0 to infinite times and is assigned the Rest of Pages placement option in the second page set.

The master pages are grouped into page sets:

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



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

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

Specify which side of the paper the master page is applied to

For forms intended for double-sided printing, you can specify which side of the paper the master page is applied to:

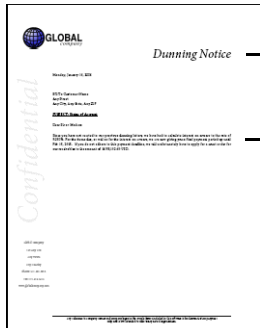
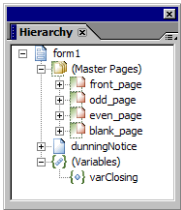
- odd (front)
- even (back)
- no odd/even restrictions
- blank

Use master pages in a form intended for double-sided printing

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

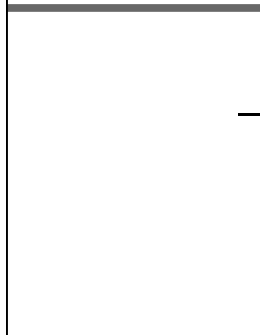
Insert a blank page after the first page in a form intended for double-sided printing

To insert a blank page after the first page in a form for double-sided printing, the page set must include master pages for the first, odd, and even printed pages. For example, to print a blank page on the back side of the cover letter, specify that the area following the cover letter subform begins on an odd page. A blank page is inserted as the first even page when the form is printed.

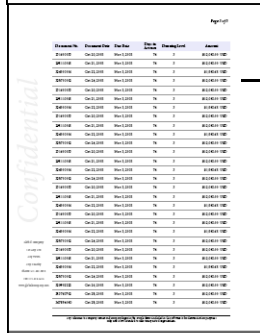


This page uses the front_page master page.

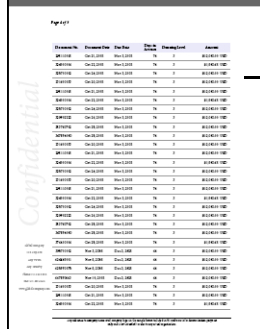
The area after this subform is set to Go To Next Odd Page



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



This page uses the odd_page master page



This page uses the even_page master page

Select a blank page after the first page

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

Preview and test printing forms

Testing your form and its data bindings in Designer ES2 avoids the need to deploy the forms to the server and generate a printed test output.

As you work in Designer ES2, you can preview the form in the Preview PDF tab. (You must have Adobe Acrobat® Professional, Acrobat Standard or Adobe Reader installed to use the Preview tab.) Use the Preview PDF tab to view and test the operation of the form as if it were a PDF file. To set options for previewing interactive forms or printable forms in PDF, use the Form Properties dialog box (Preview tab).

By previewing the form with sample data, you can determine whether the layout, formatting, content, and behavior of a form responds as expected. You can preview the form with data from an existing XML data file. You can also generate sample data by clicking the Generate Preview Data button on the Preview tab in the Form Properties dialog box.

Also, you can print the form design with sample data to a network printer by selecting Print Form With Data in the Print dialog box. When you print the form, the data values from the sample XML file appear in the respective objects. If you are setting up the form for double-sided printing, you can use sample data to test printing. You can also review how the rendered form looks when the printed page ends on an odd page or an even page. You can also review how it looks when it fits on one printed page. For double-sided printing, use the Two-Sided Print option in the Print dialog box.

For more information, see [LiveCycle Designer ES2 Help](#).

Note: When preparing sample data to test printing a form in Output ES2, the sample data that you use must contain only one data record. This issue is a Designer ES2 restriction; Output ES2 can merge XML form data that contains multiple records.

For example, the following XML data shows an example of a data file that contains three data records.

```
<?xml version="1.0" encoding="UTF-8"?>
<batch>
<LoanRecord>
  <mortgageAmount>500000</mortgageAmount>
  <lastName>Blue</lastName>
  <firstName>Tony</firstName>
  <SSN>555666777</SSN>
  <Description>Buy a home</Description>
</LoanRecord>
<LoanRecord>
  <mortgageAmount>300000</mortgageAmount>
  <lastName>White</lastName>
  <firstName>Sam</firstName>
  <SSN>555666222</SSN>
  <Description>Buy a home</Description>
</LoanRecord>
<LoanRecord>
  <mortgageAmount>700000</mortgageAmount>
  <lastName>Green</lastName>
  <firstName>Steve</firstName>
```

Setting Up Forms for Printing

```
<SSN>55566688</SSN>  
<Description>Buy a home</Description>  
</LoanRecord>  
</batch>
```

To test printing a form with data in Output ES2, the XML data must contain only one data record.

```
<?xml version="1.0" encoding="UTF-8"?>  
<LoanRecord>  
  <mortgageAmount>500000</mortgageAmount>  
  <lastName>Blue</lastName>  
  <firstName>Tony</firstName>  
  <SSN>555666777</SSN>  
  <Description>Buy a home</Description>  
</LoanRecord>
```

For information about creating sample data files in Designer ES2, see the topics “To create sample data to print with your form” and “To automatically generate sample data to print with your form” in [LiveCycle Designer ES2 Help](#).

Specify the device configuration file

Designer ES2 includes device configuration files (XDC) for PCL, PostScript, and the supported label formats you can select from when test printing your forms.

The device configuration files are installed in the Designer ES2 installation folder and appear as options in the Configuration File list in the Print dialog box. Custom device configuration files that you place in the Designer ES2 installation folder also appear as options in the Configuration File list. You can use the custom device configuration files to test printing your forms using the same XDC file that Output ES2 uses to render your forms.

The Configuration File list shows the name of the XDC file in brackets, such as PCL - Black & White [hppcl5e].

About Output-IVS

As part of your testing, you may want your network printer administrator to test rendering your forms in Output ES2. Output Installation and Verification Sample (Output IVS) is a sample application that ships with Output ES2. It provides network printer administrators with an interface that they can use to test the rendering of your forms.

Font Handling

Here are some guidelines for font handling:

- The set of fonts in LiveCycle Designer ES2 is determined from the fonts on the computer and fonts that are enumerated in the Designer.xdc file.
- It is important that the fonts used to design the form in Designer ES2 are also available when the form is output from LiveCycle Output ES2. For example, when Output ES2 renders a PDF file that uses unavailable fonts, the missing fonts are substituted. The font substitution can cause the layout of the form to change from the original design in Designer ES2. Similarly, rendering documents in PostScript, Printer Control Language (PCL), or one of the supported label formats produces unexpected results when missing fonts are substituted.
- Output ES2 can embed or reference additional fonts in the generated output.
- For PDF forms, you can select the Embed Fonts option in the Form Properties (Save As tab) dialog box to embed the fonts. (See [LiveCycle Designer ES2 Help](#).)
- When output is generated for either PCL or PostScript, preference is given to fonts known to be standard fonts included on the printer. These fonts are known as *printer-resident fonts*. Printer-resident fonts are stored in a printer's memory or on a hard drive that is connected to the printer.
- Consult with your network printer administrator and test printing a few sample forms from Output ES2 to determine which fonts are available.
- Forms that result in a smaller file size are rendered more quickly and transferred to the printer more efficiently. To minimize the file size, use printer-resident fonts and keep the number of embedded fonts used to a minimum.
- To ensure that a form design looks the same across PDF, PCL, and PostScript output devices, use these fonts, which are resident in most computers:
 - Courier
 - Arial
 - Times New Roman
- Alternatively, you can choose to embed a font. The embedded font looks the same across all devices as well, but a larger output file is produced.
- For form designs that are rendered only as PDF forms, use any of the following fonts, which are included with Adobe Reader to optimize performance:
 - Courier Std, Courier Std Bold, Courier Std Bold Oblique, Courier Std Oblique
 - Minion[®] Pro Bold, Minion Pro Bold It, Minion Pro It, Minion Pro Regular
 - Myriad[®] Pro Bold, Myriad Pro Bold It, Myriad Pro It, Myriad Pro Regular
 - Symbol (Type 1)

Specifying printer-resident fonts in Designer ES2

The fonts you use in your form design do not need to be present on the computer hosting Designer ES2. Usually, you select a font from the Font list in the Font palette; however, Designer ES2 also permits you to enter a font name. The font name you enter must be identical to the network printer's printer-resident font name. (See [LiveCycle Workbench ES2 Help](#).)

Font mapping

Designer ES2 and Output ES2 provide font mapping to accommodate the unique font requirements of printers. In the Designer ES2 installation directory, the Designer.xci file contains the default set of font mappings. This configuration file can be extended to accommodate additional font mappings. However, this file affects only Designer ES2. The font mappings used for processing the form, in concert with a particular XDC device profile, must also be in the target XDC file. For information about adding a new mapping to the font-mapping table, see [LiveCycle Designer ES2 Help](#)

The following excerpt is from the Designer.xci file. It shows the XML markup describing font-mapping statements that map requests for various Helvetica fonts to similar Arial fonts.

```
<equate from="Helvetica Black_*_*" to="Arial Black_*_*" force="0"/>
<equate from="HelveticaBlack_*_*" to="Arial Black_*_*" force="0"/>
<equate from="Helvetica-Black_*_*" to="Arial Black_*_*" force="0"/>
<equate from="Helvetica_*_*" to="Arial_*_*" force="0"/>
<equate from="Helv_*_*" to="Arial_*_*" force="0"/>
```

The font-mapping capabilities provide for strict control over mapping. You can map whole typefaces or an individual typeface with a particular weight and posture. The `force` attribute denotes whether a font is always mapped or only when the requested font is not available.

The equate font-mapping statements are evaluated in order until a matching statement is encountered.

(See “Adding a new mapping to the font-mapping table in LiveCycle Designer ES2” in [LiveCycle Designer ES2 Help](#).)

Paper Tray Handling

In LiveCycle Designer ES2, you can select paper from different input trays. You associate a master page with a paper type (media type) in the Paper Type list on the Master Pages tab in the Object palette. Because a form design can have more than one master page, the form design can contain multiple paper types.

The paper types that are available in Designer ES2 are derived from the Designer.xdc file. Using the Designer.xdc file, printers can select the paper trays that they pull basic paper types from, such as letter and legal.

Single paper type for a paper size: If the printer has one paper tray with plain letter-sized paper and the form's master pages use letter-sized paper, the printer prints on the letter-sized paper. This process is also true for legal and other sizes that are installed on the printer.

Four or fewer paper types for a paper size: The Designer.xdc file defines a set of special paper types. These paper types can be used for most cases. However, if necessary, you can create new ones and hand-edit the XDC file that Designer ES2 uses (Designer.xdc) (See [LiveCycle ES XDC Editor Help](#)). For letter size, the following paper types are defined:

- Letter Plain
- Letter Letterhead
- Letter Color
- Letter Special

An equivalent set of paper types exists for the A4 and B4 JIS paper sizes.

For example, an application can use letter-sized paper from three different trays without modifying the Designer.xdc file. This example assumes that the correct paper types are installed in the input trays that are associated with each paper type. The associations are defined in the XDC file for the printer.

Here is another example of a job that can be printed using the Designer.xdc file without modification. Consider an insurance document package that uses four different letter-size paper types:

- First page uses letterhead. (Select Letter Letterhead in the Paper Type list in the Master Page tab.)
- Second page through third-to-last page use plain white. (Select Letter Plain in the Paper Type list in the Master Page tab.)
- Second-to-last page uses blue paper with some notice/information. (Select Letter Color in the Paper Type list in the Master Page tab.)
- Final page uses preprinted and prescored insurance cards at the bottom. (Select Letter Special in the Paper Type list in the Master Page tab.)

This example assumes that the correct paper types are installed in the input trays that are associated with each paper type. The associations are defined in the XDC file for the printer.

You can also add unique paper types to the Designer.xdc file. Work closely with your network printer administrator to ensure that the new paper type information is included in the Designer.xdc file and the XDC file. These files are used by the LiveCycle Output ES2 server network printers. (See [LiveCycle ES XDC Editor Help](#).)

Designing for Label Printers

By default, LiveCycle Designer ES2 uses the Designer.xdc file. Designer ES2 also includes a separate XDC file called *Designer.xdc.label* that is created specifically for output to the following label printers:

- Zebra
- Intermec
- Datamax
- TEC

When designing for output to one of these label printers, move or rename the default Designer.xdc file. Then rename Designer.xdc.label to Designer.xdc. This file adds label media sizes to Designer ES2 and added fonts available on the label printers.

For more information about the Designer.xdc file, see [LiveCycle ES XDC Editor Help](#).

Supported label printer features

The following table lists the available support for several Designer ES2 and label printer features.

Feature	Zebra	Intermec	Datamax	TEC
Circle	Yes	No	Yes, but thickness for circles not supported.	Yes
DPI supported	203, 300, 600	203, 300, 400	203, 300, 406, 600	203, 305, 600
Ellipse	Yes	No	No	No
flipLabel finishing option	Yes	Yes	No, but this feature can be done by using the front panel of the printer.	Yes
Images	Yes	Yes	Yes	Yes
Language Support	Latin1, Latin2, HATV, Asian (CJK)	Latin1, Latin2, HATV	Latin1, Latin2	Latin1, Latin2, HATV
Line styles	No	No	No	No
Reverse Printing	Yes	Yes	Yes	Yes, but only the region behind the text is highlighted.
RFID	Yes	No	No	No
Rounded Corners for rectangles	Yes	No	No	Yes
Shading for Objects	Solid Black	Solid Black	Yes	No
Slanted Lines	Yes, but thickness for slanted lines is not supported.	No	Yes, but thickness for slanted lines is not supported.	Yes
Table Object	yes	Yes	No	Yes

Testing the label printer's standard fonts

The label printer's standard fonts that appear in Designer ES2 are representative of the fonts that are printed. They may not accurately reflect how the font appears on the printed label. Therefore, it is recommended that you test print your labels to see a true representation of the font you chose.

Supported barcodes


The following table lists the barcodes that are supported for the label printers. Some barcodes are software rendered (that is, they are supported as an image). Some barcodes are hardware (that is, they are supported on the printer).

Barcodes	Zebra	Intermec	Datamax	TEC
Aztec	Hardware	Not Supported	Hardware	Not Supported
codabar	Hardware	Hardware	Hardware	Hardware
code11	Hardware	Hardware	Hardware	Not Supported
code128	Hardware	Hardware	Hardware	Hardware
code128A	Hardware	Hardware	Hardware	Hardware
code128B	Hardware	Hardware	Hardware	Hardware
code128C	Hardware	Hardware	Hardware	Hardware
code128SSCC	Software Rendered	Hardware	Not Supported	Not Supported
code2Of5Industrial	Hardware	Not Supported	Not Supported	Hardware
code2Of5Interleaved	Hardware	Hardware	Hardware	Hardware
code2Of5Matrix	Software rendered	Not Supported	Not Supported	Not Supported
code2Of5Standard	Hardware	Hardware	Not Supported	Not Supported
code3Of9	Hardware	Hardware	Hardware	Hardware
code49	Hardware	Hardware	Not Supported	Not Supported
code93	Hardware	Hardware	Hardware	Hardware
DataMatrix	Hardware	Hardware	Hardware	Hardware
ean13	Hardware	Hardware	Hardware	Hardware
ean8	Hardware	Hardware	Hardware	Hardware
logmars	Hardware	Not Supported	Not Supported	Not Supported
msi	Hardware	Not Supported	Not Supported	Hardware
pdf417	Hardware	Hardware	Hardware	Hardware
planetCode	Hardware	Hardware	Not Supported	Not Supported
plessey	Hardware	Not Supported	Hardware	No
postAUSCust2	Software Rendered	Software Rendered	Hardware	Software Rendered

Barcodes	Zebra	Intermec	Datamax	TEC
postAUSCust3	Software Rendered	Software Rendered	Hardware	Software Rendered
postAUSReplyPaid	Software Rendered	Software Rendered	Hardware	Software Rendered
postAUSStandard	Software Rendered	Software Rendered	Hardware	Software Rendered
postUKRM4SCC	Software Rendered	Software Rendered	Software Rendered	Hardware
postUS5Zip	Hardware	Hardware	Hardware	Hardware
postUSDPBC	Hardware	Hardware	Hardware	Hardware
postUSStandard	Hardware	Hardware	Hardware	Hardware
QR Code	Hardware	Hardware	Hardware	Hardware
rss14	Hardware	Hardware	Hardware	Hardware
rss14Stacked	Hardware	Hardware	Hardware	Hardware
rss14StackedOmni	Hardware	Hardware	Hardware	Hardware
rss14Truncated	Hardware	Hardware	Hardware	Not Supported
rssExpanded	Hardware	Hardware	Hardware	Hardware
rssLimited	Hardware	Hardware	Hardware	Hardware
upcA	Hardware	Hardware	Hardware	Hardware
upcE	Hardware	Hardware	Hardware	Hardware
upcean2	Hardware	Not Supported	Hardware	Hardware
upcean5	Hardware	No	Hardware	Hardware
UPS Maxicode	Hardware	Hardware	Hardware	Hardware
USPS Intelligent Mail	Software Rendered	Software Rendered	Hardware	Software Rendered

Using RFID barcode objects

Radio Frequency Identification (RFID) printers/encoders simultaneously encode an integrated circuit chip that is embedded in the label. RFID printers/encoders also print barcodes and text on the label. The encoding is accomplished by using radio frequency waves. RFID does not require a visible tag or label to read its stored data.

The RFID barcode object ( RFID) is for printers that support ZPL only. The RFID barcode object is listed in the Barcodes category of the Object library palette and is used to program the RFID. Designer ES2 uses this object as a placeholder to represent the RFID in the form design.

For more information about specifying the RFID parameters, see [LiveCycle ES XDC Editor Help](#).

Considerations for Batch Data Usage

LiveCycle Output ES2 can process XML files that contain one record or many records. These XML files are known as *batch data files*. When processing a batch XML data file, Output ES2 can generate a single output file that contains many documents (one output file for each record). Output ES2 can also generate a separate file for each document that is generated (one output file for each record)

The following XML data shows an example of a data file that contains three data records.

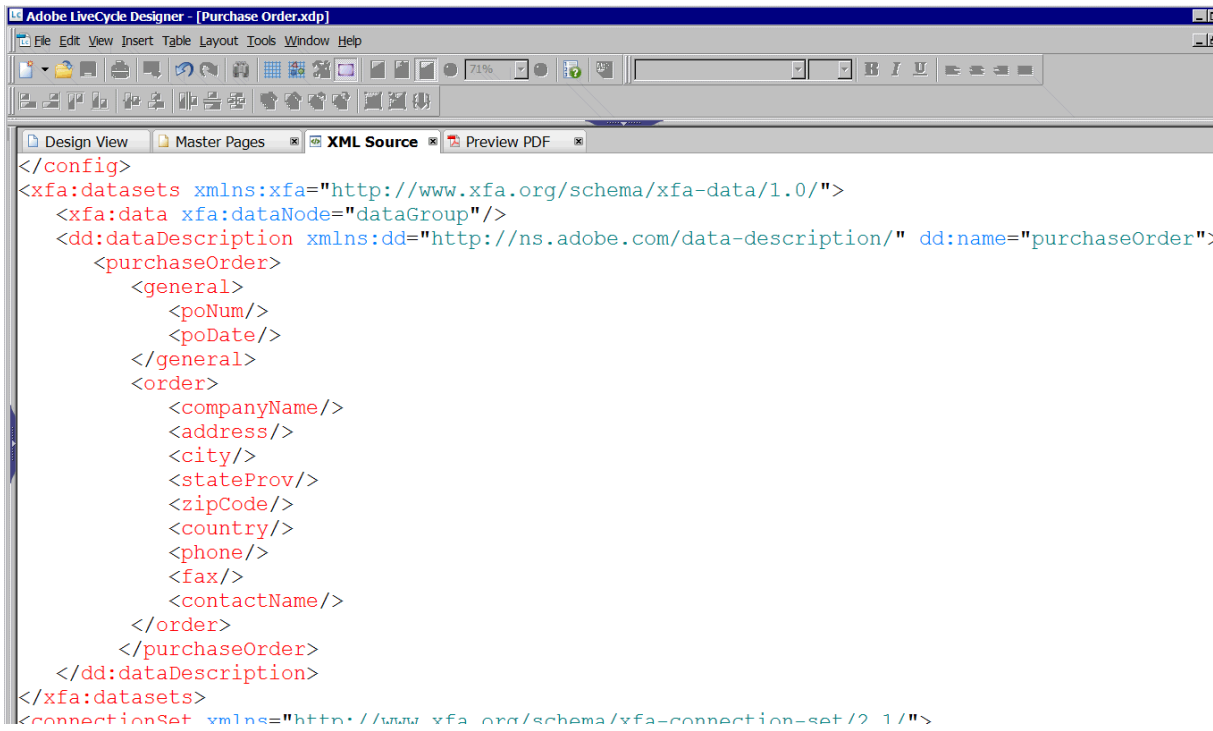
```
<?xml version="1.0" encoding="UTF-8"?>
<batch>
  <LoanRecord>
    <mortgageAmount>500000</mortgageAmount>
    <lastName>Blue</lastName>
    <firstName>Tony</firstName>
    <SSN>555666777</SSN>
    <Description>Buy a home</Description>
  </LoanRecord>
  <LoanRecord>
    <mortgageAmount>300000</mortgageAmount>
    <lastName>White</lastName>
    <firstName>Sam</firstName>
    <SSN>555666222</SSN>
    <Description>Buy a home</Description>
  </LoanRecord>
  <LoanRecord>
    <mortgageAmount>700000</mortgageAmount>
    <lastName>Green</lastName>
    <firstName>Steve</firstName>
    <SSN>555666888</SSN>
    <Description>Buy a home</Description>
  </LoanRecord>
</batch>
```

Note: Notice that the XML element that starts and ends each data record is named *LoanRecord*. The application logic that generates each document references this XML element.

Incremental loading of multiple records in a batch

Output ES2 incrementally loads each record in a batch into memory, processes the record, and releases it before moving on to read the next record. Incremental loading is useful when merging a form design with a batch data set. The result is an improvement in the volume of records that can be processed. The reason is because not all the records are loaded and merged at once into memory.

To enable incremental loading, remove the `<xfa:datasets>` node from the form design in LiveCycle Designer ES2. This restriction applies to the current version of Output ES2. You can view and remove the `<xfa:datasets>` node in the XML Source tab.



The network printer administrator can confirm that incremental loading is enabled after processing the form using Output ES2. The network printer administrator can check the server log file for the following message:

```
Unable to perform an incremental load. Performing a full data load.
```

This message is to inform you that Output ES2 examined the form design and discovered the presence of a data schema. It determines this fact by searching for the `<dataset>` node. If it discovers this node, it abandons the attempt to load the data incrementally (one record at a time). Instead, it merges the entire data stream with the form design.

Designing for multiple records

The sample batch XML above contains multiple instances of `LoanRecord`; however, the record starts at the `Batch` node. The network printer administrator uses the following run-time options:

- **Output ES2 Record Level** Indicates that the data file has multiple records
- **Record Name indicates where the record begins** The Record Level option specifies the level where Output ES2 creates a document for each node. For example, if you look at the sample XML, `Batch` is the root; therefore it is at level 1. The `LoanRecord` node is next; therefore it is level 2. In this example, the network printer administrator would enter 2 in the Record Level field to indicate that each record begins at the `LoanRecord` node.

The Record Level option is used when all the records in the XML data are of the same type. However, if the XML data has a mix of various record types, the network printer administrator can use the Record Name option. The Record Name option specifies the record type to use for generating output. For more information about these options, see [LiveCycle Workbench ES2 Help](#).

Remember the Record Level and Record Name Output ES2 run-time options as you design your form and data binding in Designer ES2. These options allow you to take advantage of incremental loading,

To ensure that the records are properly designed and the data binding functions properly, start with a schema that represents one record. Match the record to record level 2 or 3 in the Record Level Output ES2 run-time option. If the XML data contains various record types, create a representation for each record, matching each to the Record Name option.

Avoid using [*] at the record node because it causes Output ES2 to look to repeating the record rather than loading a new record.

After you test the form and are satisfied, you can base the remaining records on the ones used for testing.

Scripting methods for use when processing batch data sets

Several scripting methods that you can use in the Designer ES2 scripting environment are available.

Method	Description
documentCountInBatch	Determines the number of documents in the current batch
absPageCountInBatch	Determines the page count of the current batch
documentInBatch	Determines the ordinal number of the current document within the batch
sheetCountInBatch	Determines the sheet count of the current batch
absPageInBatch	Determines which page within the batch contains the form object
cancelBatch	Cancel a pending batch update
sheetInBatch	Determines which sheet within the batch contains the form object
updateBatch	Writes all pending batch updates to the data source

For the most up-to-date information about scripting objects, properties, and methods, see [LiveCycle Designer ES2 Scripting Reference](#).