



Understanding the Differences Between Static and Dynamic PDF Forms

July 2008

Adobe® LiveCycle® Designer ES

Version 8.2

© 2008 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle® Designer ES (8.2) Understanding the Differences Between Static and Dynamic PDF Forms for Microsoft® Windows® Edition 1.0, July 2008

If this guide is distributed with software that includes an end user agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names, company logos and user names in sample material or sample forms included in this documentation and/or software are for demonstration purposes only and are not intended to refer to any actual organization or persons.

Adobe, the Adobe logo, Acrobat, LiveCycle, and Reader 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.

This product contains either BSAFE and/or TPEM software by RSA Security, Inc.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes software developed by the IronSmith Project (<http://www.ironsmith.org/>).

This product includes software developed by the OpenSymphony Group (<http://www.opensymphony.com/>).

This product includes software developed by the Indiana University Extreme! Lab (<http://www.extreme.indiana.edu/>).

This product includes copyrighted software developed by E. Wray Johnson for use and distribution by the Object Data Management Group (<http://www.odmg.org/>).

Portions © Eastman Kodak Company, 199- and used under license. All rights reserved. Kodak is a registered trademark and Photo CD is a trademark of Eastman Kodak Company.

Powered by Celequest. Copyright 2005-2008 Adobe Systems Incorporated. All rights reserved. Contains technology distributed under license from Celequest Corporation. Copyright 2005 Celequest Corporation. All rights reserved.

Single sign-on, extending Active Directory to Adobe LiveCycle ES provided by Quest Software “www.quest.com/identity-management” in a subsequent minor release that is not a bug fix (i.e., version 1.1 to 1.2 but not 1.1.1 to 1.1.2) of the Licensee Product that incorporates the Licensed Product.

The Spelling portion of this product is based on Proximity Linguistic Technology.

©Copyright 1989, 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 Merriam-Webster Inc. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2003 Franklin Electronic Publishers Inc. © Copyright 2003 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 Franklin Electronic Publishers, Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1991 Dr.Lluis de Yzaguirre I Maura © Copyright 1991 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 Munksgaard International Publishers Ltd. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1995 Van Dale Lexicografie bv © Copyright 1996 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 IDE a.s. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 Franklin Electronics Publishers, Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1992 Hachette/Franklin Electronic Publishers, Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 Bertelsmann Lexikon Verlag © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 MorphoLogic Inc. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 1990 Williams Collins Sons & Co. Ltd. © Copyright 1990 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA. © Copyright 1993-95 Russicon Company Ltd.

© Copyright 1995 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

© Copyright 2004 IDE a.s. © Copyright 2004 All Rights Reserved Proximity Technology A Division of Franklin Electronic Publishers, Inc. Burlington, New Jersey USA.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110, USA.

Notice to U.S. Government End Users. The Software and Documentation are "Commercial Items," as that term is defined at 48 C.F.R. §2.101, consisting of "Commercial Computer Software" and "Commercial Computer Software Documentation," as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Contents

1	About This Document.....	5
	Who should read this document?	5
	Additional information.....	5
2	Comparing Static and Dynamic PDF Forms	6
	Differences between static and dynamic PDF forms	6
	Considerations when creating static and dynamic PDF forms.....	7
	Applying formatting by using client-side scripts	7
	Saving object formatting.....	9
	Unavailable commands for dynamic forms in Adobe Reader	10
	Font behavior.....	10
	Objects and properties for static PDF forms.....	10

1

About This Document

This document describes the two different types of PDF forms that Adobe® LiveCycle® Designer ES (8.2) creates, and is intended to provide considerations for form authors when they are deciding which type of PDF form to create.

Who should read this document?

This document is intended for form authors who are responsible for creating PDF forms.

Additional information

The resources in this table can help you learn more about Adobe LiveCycle ES (Enterprise Suite) Update 1.

For information about	See
Creating forms using Designer ES	LiveCycle Designer ES Help
Other services and products that integrate with LiveCycle ES	www.adobe.com
Patch updates, technical notes, and additional information on this product version	LiveCycle Technical Support

2

Comparing Static and Dynamic PDF Forms

When creating forms in LiveCycle Designer ES, form authors can save their form designs in several different formats. Most often, the form author saves a form design as a PDF form so that form fillers can easily access the form using Adobe Reader®. When saving a form design as a PDF form in Designer ES, the form author can select between two different types of PDF forms: static and dynamic.

To understand the difference between static and dynamic PDF forms, it is important to understand what the term render means. Rendering a form is the process of creating the precise final layout and formatting from the form design. The form may or may not be merged with data, depending on the form design and data. Rendering can be done by LiveCycle ES on the server or by Adobe Reader on the client. A PDF form must be rendered before it can be displayed to the end user or printed. Therefore, the rendering is the final image.

Static PDF forms render once and are displayed on the client in the Adobe Acrobat® or Adobe Reader target version. They are not rerendered in response to user interaction. The form may have been designed with a flowable layout; however, when the static PDF form is created, its layout is fixed and the resultant PDF form will not rerender on the client.

Dynamic PDF forms render on the client in Adobe Reader and, depending on the end-user interactions, can rerender on the client several times. Changes to the appearance of an object are possible in Adobe Reader because Adobe Reader has enough information to rerender the final output. For example, objects can change color, pagination can change, and objects can appear or disappear. If the end user clicks a button that adds a new row to a table, the form is rerendered in Adobe Reader. Dynamic PDF forms were first introduced in Adobe Reader, Acrobat Professional and Acrobat Standard 7.0.

Both types of PDF forms can be interactive or non-interactive.

Forms can be designed to capture data directly from end users. End users fill the form and select options online, and return the form data according to some prescribed process. These forms are known as interactive forms. You can author interactive forms that the end user fills using Acrobat Professional, Acrobat Standard, or Adobe Reader, or you can author forms for LiveCycle ES that the user fills in a web browser.

Forms can be designed to present information to end users. The data can come from a variety of data sources, such as databases, web services, or enterprise content management systems. The end user views the form already prepopulated with data. The end user cannot modify the data in the form or add new data to the form. These types of forms are known as non-interactive forms. A typical scenario for these types of forms involves LiveCycle ES as part of the solution. LiveCycle ES merges the form design with data and renders the form, prepopulated with data, to the end user. A classic example of a non-interactive form is a credit card statement or telephone bill.

Differences between static and dynamic PDF forms

Static and dynamic PDF forms differ in three important areas:

File size In general, static PDF forms tend to be larger than the equivalent file that is saved as a dynamic PDF form.

Rendering location Static PDF forms render once and are displayed on the client in Adobe Reader. Because rendering is performed on the server, only small tweaks to the final output are possible on the client. For example, the value area may show a different color after the user clicks a button or exits a field.

Display speed In general, static PDF forms require more time to render on the server than dynamic PDF forms but are displayed faster in the client software on the end-user's computer. Static PDF forms require more time on the server because the server performs the page layout operations, data merging, and final rendering. In the case of dynamic PDF forms, the server performs only data merging. The client software on the end-user's computer performs the page layout operations and final rendering. As a result, the rendering time for dynamic PDF forms relies on the processing power of the end-user's computer, especially when the PDF file size is large.

Tip: As you design a form, you can see how the form behaves as a static or dynamic PDF form in the Preview PDF tab. For more information about setting preview options, see [LiveCycle Designer ES Help](#).

Considerations when creating static and dynamic PDF forms

You can create many kinds of forms as either static or dynamic with little difference to the end-user experience. However, some forms work as designed only if they are created as a static or dynamic PDF forms. In general, the choice of static over dynamic is determined by the following considerations:

- If the form works as either a static PDF form or a dynamic PDF form, use a dynamic PDF form to reduce server-side processing, which results in a greater number of transactions per second.
- If the form relies on client-side scripts to change the layout (for example, it uses scripts to add or remove rows from a table or to make text fields grow), use a dynamic PDF form.
- If end users will need to add annotations or comments to the PDF form, use a static PDF form.
- If the form must work with Acrobat installations earlier than version 7.0, use a static PDF form.

Additionally, there are a number of specific issues to consider when choosing to create a static or dynamic PDF form. These issues are differences in the behavior between the form types that may be critical in making your decision:

- ["Applying formatting by using client-side scripts" on page 7](#)
- ["Saving object formatting" on page 9](#)
- ["Unavailable commands for dynamic forms in Adobe Reader" on page 10](#)
- ["Font behavior" on page 10](#)
- ["Objects and properties for static PDF forms" on page 10](#)
- If your form includes digital signatures, there are limitations to the way that you can use dynamic PDF forms with the Signature service. For more information about these limitations, see [LiveCycle ES Services](#).

Applying formatting by using client-side scripts

In a static PDF form, only the value area of the field can be updated on the client. Everything else on the form is frozen or fixed. In a dynamic PDF form, the entire field object can be updated on the client. The examples that follow are provided to give you an idea of what happens.

Changing the field border color

If you create a client-side script to apply borders to objects, the results differ in static and dynamic PDF forms.

For example, you can write a script on the exit event of a field to turn the border color red:

```
TextField1.border.edge.color.value="255,0,0"
```

- In a static PDF form, the outline of only the value area turns red after the end user exits the field.



- In a dynamic PDF form, the outline of the entire field object turns red after the end user exits the field.



Applying field shading

If you create a client-side script to apply shading to objects, the results differ in static and dynamic PDF forms.

For example, you can write a script on the exit event of a field to turn the shading color in the field to red:

```
TextField1.fillColor = "255,0,0"
```

- In a static PDF form, only the value area turns red after the end user exits the field.



- In a dynamic PDF form, the entire object, including the value area, turns red after the end user exits the field.



Hiding objects

If you create a client-side script to hide objects, the results differ in static and dynamic PDF forms.

When you are designing a static or dynamic PDF form, you can hide objects on the form by setting the presence property to either "invisible" or "hidden". In both cases, the objects do not appear in the final output; however, when you set the presence value to "hidden", objects do not occupy any space in the layout. If the objects are inside flowed subform objects, the subform itself will shrink in response to the hidden objects.

The difference between static and dynamic PDF forms is that on a static PDF form you cannot change the presence value of an object by using an interactive scripting event, such as the click event of a button. Static PDF forms cannot rerender on the client; therefore, scripts executed on interactive events cannot change the visibility of form objects. Dynamic PDF forms do not have the same limitation because they can rerender on the client.

You can change the visibility of form objects on static PDF forms by using non-interactive events that trigger during form rendering, such as the initialize event. For example, on a dynamic PDF form, you can write a script on the click event of a button to hide a text field:

```
TextField1.presence = "invisible"
```

Alternatively, you can use this script to completely remove the text field from the layout:


```
TextField1.presence = "hidden"
```

In both cases, to achieve the same results on a static PDF form, you use the same scripts but write them on the initialize event of either the text field or the button.

Saving object formatting

Prior to Acrobat 8.0, in a static PDF form, the object's formatting is saved when the end user saves, closes, and reopens the form. In a dynamic PDF form, the object's formatting is not saved.

Caution: Although it is possible to save an object's formatting on a dynamic PDF form by using client-side scripting, it is not a recommended practice and should be avoided. If saving an object's formatting is a requirement, you need to use a static PDF form.

Beginning with Acrobat 8.0, the state of all the objects is saved and can be restored automatically or manually upon reopening. This is controlled by the `restoreState` scripting property on the root subform. For more information about the `restoreState` scripting property, see [LiveCycle Designer ES Help](#).

Object formatting, as well as the layout of a dynamic PDF form, can change when any of the following form actions occur:

- The user opens the form
- The user imports new data
- The user adds new data to the form
- A client-side script is run

The following table compares the end-user experience using the examples in the section [“Applying formatting by using client-side scripts” on page 7](#) when saving as either a static PDF form or a dynamic PDF form in Adobe Reader.

Form content	Static PDF form	Dynamic PDF form
A client-side script to change the border color of a field. (See “Changing the field border color” on page 8.)	The outline of the value area remains red when the form is saved, closed, and reopened.	The outline of the object does not remain red when the form is saved, closed, and reopened.
A client-side script to apply shading. (See “Applying field shading” on page 8.)	The fillable value area remains red when the form is saved, closed, and reopened.	The object, including the value area, does not remain red when the form is saved, closed, and reopened.
A client-side script to hide objects. (See “Hiding objects” on page 8.)	The value in the text field remains invisible when the form is saved, closed, and reopened.	The text field does not remain hidden when the form is saved, closed, and reopened.

Note: If `restoreState` is set to `auto`, all of these changes are maintained if the document is saved and then reopened in Acrobat 8.0.

Unavailable commands for dynamic forms in Adobe Reader

When a PDF form is opened in Adobe Reader, some Adobe Reader commands that are available for static PDF forms are not available for dynamic PDF forms. The reason is because the final layout of the form can change the position of objects, which in turn affects the number of form pages.

The following Adobe Reader commands are available for static PDF forms but not for dynamic PDF forms:

- Comment & Markup tools
- Import Comments and Export Comments
- Insert Pages, Extract Pages, Replace Pages, Delete Pages, Crop Pages, and Rotate Pages

Font behavior

When you create PDF forms, it is possible to include only those characters of a font that are actually used in the form. This technique is called font subsetting. You can also embed fonts into the form so that end users have all of the fonts they need to use the form. In this case, the size of the PDF file is larger.

Note: The manufacturer of the font can specify the level of embedding that is allowed.

Fonts can be subset in static PDF forms

In static PDF forms, fonts can be subset for text and field captions. Subsetting reduces the size of the PDF file. To subset fonts into the form, the font you select must be set to the Print & Preview Embedding Allowed level.

Note: The fonts used in the fillable area of fields cannot be subset on static PDF forms because the user input can contain any character from the font.

Fonts are embedded in dynamic PDF forms

In dynamic PDF forms, fonts are fully embedded or linked. Embedded fonts are never subset. When a form that has embedded fonts is opened, even if the fonts are already on the system, Adobe Reader uses the embedded fonts. To embed fonts into the form, the embedding settings in the fonts you select must allow embedding for edit.

Objects and properties for static PDF forms

Here is the list of scripting objects that form authors can use to make changes to fields in static PDF forms in Adobe Reader. For more information about scripting objects, see the [LiveCycle Designer ES Scripting Reference](#).

Object	Description
border	Describes the border surrounding an object
color	Describes a unique color on an object
fill	Applies a color and optional rendered designs to the region enclosed by an object

Here is the list of scripting properties that form authors can change for fields in static PDF forms in Adobe Reader. For more information about scripting properties, see the [LiveCycle Designer ES Scripting Reference](#)

Property	Description	Syntax
access	Controls user access to the contents of a container	<code>SOMexpression.access = "open readOnly protected"</code>
fontColor	The font color value for the text	<code>SOMexpression.fontColor = "[0-255] , [0-255] , [0-255]"</code>
hAlign	Specifies the horizontal text alignment	<code>SOMexpression.hAlign = "left center right"</code>
maxChars	Specifies the maximum number of characters that this text value can enclose	<code>SOMexpression.maxChars = "0 integer"</code>
maxLength	Specifies the maximum (inclusive) permitted length of the content or -1 to indicate that no maximum length is imposed	<code>SOMexpression.maxLength = "0 integer"</code>
multiLine	Specifies whether the text may span multiple lines	<code>SOMexpression.multiLine = "0 1"</code>
presence	Specifies the visibility of an object's value	<code>SOMexpression.presence = "visible invisible hidden"</code>
textEntry	Determines whether a user can type a value into a drop-down list	<code>SOMexpression.textEntry = "0 1"</code>

Index

A

Adobe Reader
unavailable commands for dynamic PDF forms 10

D

dynamic PDF forms
about creating 7
differences with static 6
font embedding 10
formatting using client-side scripts 7
saving object formatting 9
unavailable commands in Adobe Reader 10

E

embedding
fonts into PDF forms 10

F

fonts
embedding into PDF forms 10

P

PDF forms
differences between static and dynamic 6
dynamic 7
font subsetting and embedding 10
formatting using client-side scripts 7
formatting via client-side scripts 7
objects and properties 10
saving object formatting 9
static 7

S

static PDF forms
about creating 7
differences with dynamic 6
font subsetting 10
formatting using client-side scripts 7
objects 10
properties 10
saving object formatting 9
subsetting fonts in PDF forms 10