Scripting Support for HTML Forms and Form Guides

Adobe® LiveCycle® Designer ES

Version 8.2

February 2009

© 2009 Adobe Systems Incorporated. All rights reserved.

Adobe® LiveCycle® Designer ES 8.2 Scripting Support for HTML Forms and Form Guides for Microsoft® Windows®
Edition 1.1, February 2009

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, Flash, LiveCycle, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apple is a trademark of Apple Inc., registered in the United States and other countries.

Microsoft and Windows are either registered trademarks or 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 TIPEM 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.comlidentity-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.

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.

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

About This Document **5**
Who should read this document? 5
How to use this document 5
Additional information 6
1 Overview **7**
Objects that support scripting 8
2 Creating Scripts for HTML Forms and Form Guides **10**
Choosing a scripting language 10
Executing JavaScript in a specific client application 10
Considerations when creating JavaScript for form guides 11
Limitations when working with data sources 12
3 Object-Level Scripting Support **13**
xfa 13
data 13
datasets 14
eventpseudomodel 14
hostpseudomodel 14
form 15
field 16
subform 17
4 Scripting Events **20**
5 Scripting Properties **22**
6 Scripting Methods **32**

About This Document

This document outlines the XML Form Object Model scripting events, objects, properties, and methods that are available for forms rendered in HTML and as form guides. For more information about the availability of other features, see *Target Version Reference* and *Transformation Reference*.

Who should read this document?

This document is intended for form developers interested in either creating scripts for forms that will be rendered into HTML or as form guides, or in maintaining existing scripting capabilities across several form output types. Knowledge of JavaScript™, the XML Form Object Model, as well as data binding using Adobe® LiveCycle® Designer ES 8.2 is expected.

How to use this document

This document is divided into sections to present similar kinds of information in different ways. The intent is to allow users to locate what they need as quickly as possible.

Chapter	Provides information on
Overview	The scope of the support for scripting in HTML forms and form guides
Creating Scripts for HTML Forms and Form Guides	Considerations for creating scripts that may need to run on multiple client applications, as well as techniques for adapting your existing scripts
Object-Level Scripting Support	What scripting objects are supported for HTML forms and form guides. For each object, any supported properties and methods are also listed. Use this chapter if you want to know, generally, what properties and methods are supported for a particular scripting object.
Scripting Events	A complete, alphabetical list of all scripting events and their availability for HTML forms and form guides
Scripting Properties	A complete, alphabetical list of all scripting properties and their availability for HTML forms and form guides. Use this chapter if you want to know the availability for a specific scripting property.
Scripting Methods	A complete, alphabetical list of all scripting methods and their availability for HTML forms and form guides. Use this chapter if you want to know the availability for a specific scripting method.

Additional information

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

For information about	See
Detailed information about creating and editing form guides using Guide Builder	*LiveCycle Designer ES Help*
Getting started creating scripts in Designer ES	*LiveCycle Designer ES Scripting Basics*
Complete reference information detailing the scripting objects, properties, methods, and events available in Designer ES	*LiveCycle Designer ES Scripting Reference*
An overview of what from guides are, how you can create them, and how to integrate them into processes using Adobe LiveCycle Workspace ES	*Getting Started with Form Guides*
LiveCycle ES terminology	*LiveCycle ES Glossary*
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

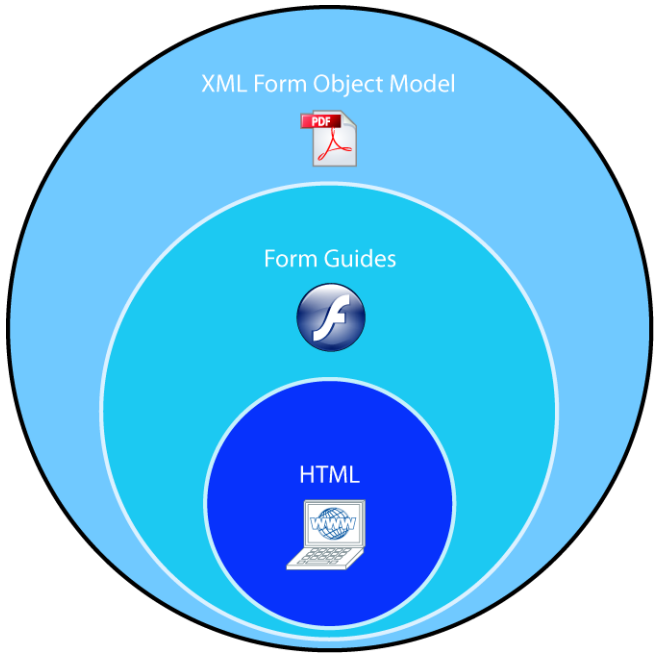
1 Overview

When you create forms using LiveCycle Designer ES, you have the option of rendering those forms in either PDF or HTML, or creating form guides based on Adobe Flash® technology. End users view each of the formats (PDF, HTML, and form guides) by using an associated client application. The following table provides an overview of the client applications required to view PDF forms, HTML forms, and form guides.

Format	Client application
PDF	Adobe Acrobat® Professional, Acrobat Standard, Adobe Reader®
HTML	Microsoft® Internet Explorer, Mozilla Firefox, Netscape Navigator, Apple® Safari
Form guide	Adobe Flash Player

Note: While end users may view the PDF form, HTML form, or form guide in an application such as LiveCycle Workspace ES, it is important to remember that it is the underlying client technology that is responsible for displaying the form and its data.

Not all form features available in LiveCycle Designer ES are available across all output formats. Forms rendered in PDF format, by default, have access to the full XML Form Object Model, including all scripting events, objects, properties, and methods. Forms rendered in HTML format, or those rendered as form guides, have access to only a portion, or *subset*, of the XML Form Object Model. The following diagram illustrates the relationship between the full XML Form Object Model and the subsets available for HTML forms and form guides. Note that the support for form guides is a subset of the full XML Form Object Model, and that the support for HTML forms is a subset of what is supported for form guides.



The reduced scope of the XML Form Object Model for HTML forms and form guides means that if you are creating form designs that may be rendered in either of those formats, you will need to consider how to

structure your form scripting to avoid scripting on your form that references objects, properties, or methods that are not available, to avoid unexpected results.

Objects that support scripting

LiveCycle Designer ES uses two separate definitions of objects with respect to scripting:

- Objects that you add to your form design from the Object Library palette
- Objects exposed from the underlying XML architecture of a form design

Architecturally, objects that exist within the Object Library palette are more meaningful representations of objects exposed in the underlying XML. In fact, all objects available in the Object Library palette are derived from only four XML objects: contentArea, draw, field, and subform.

The Designer ES scripting model allows you to reference objects directly using the name of the object. For example, if you have a text field on your form, and you want to set its value, you could use the following JavaScript:

```
TextField1.rawValue = "Hello World";
```

From an architectural perspective, this is logically equivalent to the following:

```
field.rawValue = "Hello World";
```

The following table outlines how the Object Library objects map to their underlying XML equivalents.

Object on the Standard tab of the Object Library	Derived from
Content Area	contentArea
Circle, Image, Line, Rectangle, Text	draw
Barcodes (all), Button, Check Box, Date/Time Field, Decimal Field, Drop-Down List, Email Submit Button, HTTP Submit Button, Image Field, List Box, Numeric Field, Paper Forms Barcode, Password Field, Print Button, Radio Button, Reset Button, Signature Field, Text Field	field
Subform, Table (each body row, header row, and footer row is a distinct subform object)	subform subform.instanceManager

Note: You cannot script against Designer ES objects derived from contentArea or draw objects. To find what properties and methods are available for your Object Library objects, consult either the field or subform sections of Object-Level Scripting Support.

In addition to objects available from the Object Library in Designer ES, objects that are only available through scripting are supported by HTML forms and form guides. The table below outlines how you reference the supported scripting model objects using script:

Supported scripting model objects	Syntax
data data.datasets	`xfa.data.` `xfa.data.datasets.`
eventpseudomodel	`xfa.event.`

Supported scripting model objects	Syntax
hostpseudomodel	`xfa.host.`
form	`xfa.form.`
xfa	`xfa.`

For more information on the entire Adobe XML Form Object Model, see *LiveCycle Designer ES Scripting Reference*.

2 Creating Scripts for HTML Forms and Form Guides

When you add scripts to a form you create in LiveCycle Designer ES, you must decide where those scripts will execute: either on the client application, on the server, or on both the client and the server. If you choose to execute your scripts on the server in any way, your form must be deployed in conjunction with Adobe LiveCycle Forms ES so that the form and its data can return to the server when the script executes. After the script executes, the form is re-rendered and posted back to the end user. Executing scripts on the server provides you with access to the full XML Form Object Model objects, scripting properties, and scripting methods.

If you choose to run your scripts on the client application, there are restrictions to the scope of the XML Form Object Model you can access, and even what scripting language you can use. This section discusses the considerations you need to keep in mind when creating scripts for HTML forms or form guides.

Choosing a scripting language

If you are creating scripts that execute on the server, you are free to create scripts in either JavaScript or FormCalc. However, if you are creating scripts that will execute on the client application, you can only use JavaScript. This restriction applies because FormCalc is not natively available in client applications, such as internet browsers or Flash Player.

If you have an existing PDF form that you want to make available in HTML format, or as a form guide, one strategy is to set your existing FormCalc scripts to execute on the server. This technique may not always be appropriate for your specific use case, for example, if you don't want the form to make a round trip to the server as part of the form filling experience. However, this technique may be useful if you do not want to rewrite your existing FormCalc script in JavaScript.

Note: Not all scripting events can execute on the server. For example, interactive events are specific to user interactions with the form in the client application, so they cannot be set to execute on the server.

Executing JavaScript in a specific client application

If you want to customize your scripting depending on the particular client application in which the form or form guide is being viewed, you can use the `name` property of the `host` model to conditionalize your scripts. For example, the following JavaScript executes only if the form is rendered as a form guide:

```
if (xfa.host.name == "Flash")
{
   xfa.host.messageBox("This is executing in Flash Player!");
}
```

Similarly, the following JavaScript executes only if the form is rendered in HTML format (and the form is viewed in Internet Explorer):

```
if (xfa.host.name == "IE")
{
   xfa.host.messageBox("This is executing in Internet Explorer!");
}
```

Form output format	Client application	xfa.host.name
PDF	Acrobat Standard, Acrobat Professional, Acrobat Pro Extended, and Adobe Reader	Acrobat
Form guide (SWF)	Adobe Flash Player	Flash
HTML	Internet Explorer	Microsoft Internet Explorer
HTML	Mozilla Firefox	Netscape
HTML	Apple Safari	Netscape

If you want to distinguish between Firefox and Safari, you can use the standard HTML object model to access the `window.navigator.userAgent` property. The userAgent property returns a string that should contain the name of the application. For example, for Firefox 2.0.0.13 the userAgent property contains the following value:

```
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311
Firefox/2.0.0.13
```

The following JavaScript uses the `String.search()` method to determine if the value of userAgent contains either `Firefox` or `Safari`, and displays a customized message box for each case.

```
var sVar = window.navigator.userAgent;

if (xfa.host.name == "Netscape")
{
   if (sVar.search(/FireFox/) != -1)
   {
      xfa.host.messageBox("This is executing in Mozilla Firefox!");
   }
   if (sVar.search(/Safari/) != -1)
   {
      xfa.host.messageBox("This is executing in Apple Safari!");
   }
}
```

Note: Individual applications can configure the userAgent property to display any string value. You should verify the value of userAgent for specific versions of the HTML client application on which your users will view the rendered form.

Considerations when creating JavaScript for form guides

When you create a form guide based on a form design, you should consider the following:

- Scripting on master pages is ignored.

  Form guides do not make use of page-based layout. Instead, form guides use the concept of sections and panels which the form guide author creates in Guide Builder. Therefore, no scripting on master pages defined in LiveCycle Designer ES is preserved when the form guide is rendered.

- Field level dependencies are ignored, with the exception of the `rawValue` property.

  Field data values, controlled using the `rawValue` property, that are dependent on the values of another field or fields will continue to operate as expected. All other field properties that are dependent on the values of other fields will not persist on a form guide. For example, if the color of a

field's border is set to change when the border color of another field changes, this behavior will not persist on a form guide.

- Form guides cannot correctly use the `this` JavaScript shortcut reference as part of a script object.

  For example, the following function will not evaluate correctly on a form guide.

  ```
  function addToValue( nNumber )
  {
     this.rawValue = this.rawValue + nNumber;
  }
  ```

  In this case, you must modify the function to either explicitly pass the `this` shortcut reference as a variable, or pass a variable that represents a pointer to the current object as a formal parameter. For example:

  ```
  addToValue( this, 10 );
  - or -
  function addToValue( obj, nNumber)
  {
     obj.rawValue = obj.rawValue + nNumber;
  }
  ```

Limitations when working with data sources

If your form is bound to a data source, you should consider the following restrictions:

- Only simple binding against a data value is supported. Complex binding is not supported.
- If you use data binding, you must use explicit binding. For example, if a subform is using normal binding, the name of the subform must appear in the data source.
- Subform objects that can grow in response to the amount of data must be explicitly bound to elements in the data.
- The concept of form state is not supported for HTML forms or form guides. Only states represented in the data are preserved.

For more information on data binding and form state, see *LiveCycle Designer ES Help*.

3 Object-Level Scripting Support

This section lists only those scripting objects that are available for HTML forms and form guides, along with any scripting properties and methods that each object supports for those output formats. For a complete, alphabetic list of scripting properties and methods, and their availability in HTML forms and form guides, see Scripting Properties and Scripting Methods.

Note: Scripting properties and methods listed as *Not applicable* do not have any logical meaning within the specified output format. For example, form guides do not contain a concept of pages, so properties or methods related to information about form pages do not apply. These properties and methods will fail gracefully and will not generate errors or return exceptions.

xfa

The xfa object corresponds to the following Designer ES scripting syntax:

```
xfa.*
```

Properties

Scripting property	Supported in HTML	Supported in form guides
className	no	yes

Methods

Scripting method	Supported in HTML	Supported in form guides
resolveNode	yes	yes
resolveNodes	yes	yes

data

The data object corresponds to the following Designer ES scripting syntax:

```
xfa.data.*
```

Properties

None

Methods

Scripting method	Supported in HTML	Supported in form guides
resolveNode	no	yes
resolveNodes	no	yes

datasets

The data object corresponds to the following Designer ES scripting syntax:

```
xfa.data.datasets.*
```

Properties

None

Methods

Scripting method	Supported in HTML	Supported in form guides
loadXML	no	yes
resolveNode	no	yes
resolveNodes	no	yes
saveXML	no	yes

eventpseudomodel

The eventpseudomodel object corresponds to the following Designer ES scripting syntax:

```
xfa.event.*
```

Properties

Scripting property	Supported in HTML	Supported in form guides
className	no	yes
name	no	yes
newText	no	yes

Methods

None

hostpseudomodel

The hostpseudomodel object corresponds to the following Designer ES scripting syntax:

```
xfa.host.*
```

Properties

Scripting property	Supported in HTML	Supported in form guides
calculationsEnabled	yes	yes
className	no	yes
currentPage	yes	Not applicable

Scripting property	Supported in HTML	Supported in form guides
name	yes	yes
numPages	yes	Not applicable
validationsEnabled	yes	yes

Methods

Scripting method	Supported in HTML	Supported in form guides
gotoURL	yes	no
messageBox	yes	yes
pageDown	yes	no
pageUp	yes	no
print	no	no
resetData	yes	yes
setFocus	yes	no

form

The form object corresponds to the following Designer ES scripting syntax:

```
xfa.form.*
```

Supported child objects

- field
- subform

Properties

Scripting property	Supported in HTML	Supported in form guides
className	no	yes
nodes	no	yes

Methods

Scripting method	Supported in HTML	Supported in form guides
execCalculate	yes	yes
execInitialize	yes	yes
execValidate	yes	yes
recalculate	no	yes
remerge	no	yes
resolveNode	yes	yes
resolveNodes	yes	yes

field

The field object corresponds to the following Designer ES scripting syntax:

```
field_name.*
```

Properties

Scripting property	Supported in HTML	Supported in form guides
access	yes	yes
all	no	yes
borderColor	yes	Not applicable
borderWidth	yes	Not applicable
classesAll	no	yes
className	no	yes
fillColor	yes	Not applicable
fontColor	yes	Not applicable
formattedValue	yes	yes
h	yes	Not applicable
index	yes	yes
isContainer	no	yes
mandatory	yes	yes
name	yes	yes
nodes	no	yes
parent	yes	yes
parentSubform	no	yes

Scripting property	Supported in HTML	Supported in form guides
presence	yes	yes
somExpression	no	yes
validationMessage	yes	yes
w	yes	Not applicable
x	yes	Not applicable
y	yes	Not applicable

Methods

Scripting method	Supported in HTML	Supported in form guides
addItem	yes	Drop-down lists and
boundItem	no	Drop-down lists only
clearItems	yes	Drop-down lists only
deleteItem	no	Drop-down lists only
execCalculate	yes	yes
execEvent	yes	yes
execInitialize	yes	yes
execValidate	yes	yes
getDisplayItem	no	Drop-down lists only
getItemState	no	Drop-down lists only
getSaveItem	no	Drop-down lists only
resolveNode	yes	yes
resolveNodes	yes	yes
setItemState	no	Drop-down lists only

subform

The subform object corresponds to the following Designer ES scripting syntax:

```
subform_name.*
```

Supported child objects

- instanceManager
- occur

Properties

Scripting property	Supported in HTML	Supported in form guides
all	no	yes
borderColor	yes	Not applicable
borderWidth	yes	Not applicable
classesAll	no	yes
className	no	yes
fillColor	yes	Not applicable
index	no	yes
instanceIndex	no	yes
isContainer	no	yes
name	yes	yes
nodes	no	yes
parent	yes	yes
presence	yes	yes
somExpression	no	yes
validationMessage	yes	yes

Methods

Scripting method	Supported in HTML	Supported in form guides
execCalculate	yes	yes
execEvent	yes	yes
execInitialize	yes	yes
execValidate	yes	yes
resolveNode	yes	yes
resolveNodes	yes	yes

instanceManager

The instanceManager object corresponds to the following Designer ES scripting syntax:

```
subform_name.instanceManager.*
```

Properties

Scripting property	Supported in HTML	Supported in form guides
className	no	yes
count	yes	yes
isContainer	no	yes
max	yes	yes
min	yes	yes
name	yes	yes
somExpression	no	yes

Methods

Scripting method	Supported in HTML	Supported in form guides
addInstance	yes	yes
insertInstance	yes	yes
instanceCount	yes	yes
moveInstance	yes	yes
removeInstance	yes	yes
setInstances	yes	yes

occur

The occur object corresponds to the following Designer ES scripting syntax:

```
subform_name.occur.*
```

Properties

Scripting property	Supported in HTML	Supported in form guides
initial	no	yes
max	yes	yes
min	yes	yes

Methods

None

4 Scripting Events

This section lists support for XML Form Object Model scripting events in HTML forms and form guides.

For more information on scripting events and event timing, see *LiveCycle Designer ES Scripting Basics*.

Scripting event	Supported in HTML	Supported in form guides
calculate	yes	yes
change	Drop-down lists only	yes
click	yes	yes
docClose	no	no
docReady	no	no
enter	yes	yes
exit	yes	yes
form:ready	no	yes
full	no	no
indexChange	no	no
initialize	yes	yes
layout:ready	no	no
mouseDown	yes	yes
mouseEnter	no	yes
mouseExit	no	yes
mouseUp	yes	yes
postOpen	no	no
postPrint	no	no
postSave	no	no
postSign	no	no
postSubmit	no	no
preOpen	no	Drop-down lists only
prePrint	yes	no
preSave	no	no
preSign	no	no

Scripting event	Supported in HTML	Supported in form guides
preSubmit	Submit buttons only	
validate	yes	yes

5 Scripting Properties

This section lists support for XML Form Object Model scripting properties in HTML forms and form guides.

For more information on scripting properties, see *LiveCycle Designer ES Scripting Reference*.

Note: Items marked as not applicable will not generate an error or cause a script to fail; however, they do not produce any effect on the rendered form or form guide.

Scripting property	Supported in HTML	Supported in form guides
#text	no	no
{default}	no	no
access	yes	yes
accessKey	no	no
action	no	no
activity	no	no
addRevocationInfo	no	no
aliasNode	no	no
all	no	yes
allowMacro	no	no
allowNeutral	no	no
allowRichText	no	no
anchorType	no	no
appType	no	no
archive	no	no
aspect	no	no
baselineShift	no	no
bind	no	no
binding	no	no
blankOrNotBlank	no	no
bofAction	no	no
borderColor	yes	Not applicable
borderWidth	yes	Not applicable
bottomInset	no	no

Scripting property	Supported in HTML	Supported in form guides
break	no	no
calculationsEnabled	yes	yes
cancelAction	no	no
cap	no	no
change	no	no
charEncoding	no	no
checksum	no	no
circular	no	no
classesAll	no	yes
classId	no	no
classIndex	no	no
className	no	yes
codeBase	no	no
codeType	no	no
colSpan	no	no
columnWidths	no	no
commandType	no	no
commitKey	no	no
commitOn	no	no
connection	no	no
contains	no	no
content	no	no
contentType	no	no
count	yes	yes
credentialServerPolicy	no	no
crlSign	no	no
cSpace	no	no
currentPage	yes	Not applicable
currentRecordNumber	no	no
currentValue	no	no
cursorLocation	no	no

Scripting property	Supported in HTML	Supported in form guides
cursorType	no	no
data	no	no
dataColumnCount	no	no
dataDescription	no	no
dataEncipherment	no	no
dataLength	no	no
dataPrep	no	no
dataRowCount	no	no
db	no	no
decipherOnly	no	no
delayedOpen	no	no
delimiter	no	no
digitalSignature	no	no
disable	no	no
editValue	no	no
embedPDF	no	no
encipherOnly	no	no
endChar	no	no
eofAction	no	no
errorCorrectionLevel	no	no
excludeAllCaps	no	no
excludeInitialCap	no	no
executeType	no	no
fillColor	yes	Not applicable
fontColor	yes	Not applicable
fontHorizontalScale	no	no
fontVerticalScale	no	no
format	no	no
formatMessage	no	no
formattedValue	yes	yes
formatTest	no	no

Scripting property	Supported in HTML	Supported in form guides
fracDigits	no	no
from	no	no
fullText	no	no
h	yes	Not applicable
hAlign	no	no
hand	no	no
highlight	no	no
href	no	no
hScrollPolicy	no	no
hyphenate	no	no
id	no	no
imagingBBox	no	no
index	yes	yes
initial	no	yes
initialNumber	no	no
input	no	no
instanceIndex	no	yes
intact	no	no
inverted	no	no
isContainer	no	yes
isDefined	no	no
isNull	no	no
join	no	no
kerningMode	no	no
keyAgreement	no	no
keyCertSign	no	no
keyDown	no	no
keyEncipherment	no	no
labelRef	no	no
ladderCount	no	no
language	no	no

Scripting property	Supported in HTML	Supported in form guides
layout	no	no
leadDigits	no	no
leader	no	no
leftInset	no	no
length	no	no
letterSpacing	no	no
lineHeight	no	no
lineThrough	no	no
lineThroughPeriod	no	no
locale	no	no
lockType	no	no
long	no	no
mandatory	yes	yes
mandatoryMessage	no	no
marginLeft	no	no
marginRight	no	no
mark	no	no
match	no	no
max	yes	yes
maxChars	no	no
maxH	no	no
maxLength	no	no
maxW	no	no
min	yes	yes
minH	no	no
minW	no	no
model	no	no
modifier	no	no
moduleHeight	no	no
moduleWidth	no	no
multiLine	no	no

Scripting property	Supported in HTML	Supported in form guides
name	yes	yes
newContentType	no	no
newText	no	yes
next	no	no
nodes	no	yes
nonRepudiation	no	no
ns	no	no
nullTest	no	no
numbered	no	no
numberOfCells	no	no
numPages	yes	Not applicable
oddOrEven	no	no
oneOfChild	no	no
open	no	no
operation	no	no
orientation	no	no
output	no	no
override	no	no
pagePosition	no	no
parent	yes	yes
parentSubform	no	yes
passwordChar	no	no
permissions	no	no
placement	no	no
platform	no	no
posture	no	no
presence	yes	yes
preserve	no	no
prevContentType	no	no
previous	no	no
prevText	no	no

Scripting property	Supported in HTML	Supported in form guides
printCheckDigit	no	no
priority	no	no
pushCharacterCount	no	no
radius	no	no
radixOffset	no	no
rate	no	no
rawValue	yes	yes
ready	no	no
recordsAfter	no	no
recordsBefore	no	no
reenter	no	no
ref	no	no
relation	no	no
relevant	no	no
remainCharacterCount	no	no
reserve	no	no
restoreState	no	no
rightInset	no	no
role	no	no
rotate	no	no
rowColumnRatio	no	no
runAt	no	no
save	no	no
savedValue	no	no
scope	no	no
scriptTest	no	no
selectedIndex	no	no
selEnd	no	no
selStart	no	no
server	no	no
shape	no	no

Scripting property	Supported in HTML	Supported in form guides
shift	no	no
short	no	no
signatureType	no	no
size	no	no
slope	no	no
soapFaultCode	no	no
soapFaultString	no	no
somExpression	no	yes
spaceAbove	no	no
spaceBelow	no	no
startAngle	no	no
startChar	no	no
startNew	no	no
stateless	no	no
stock	no	no
stroke	no	no
sweepAngle	no	no
tabDefault	no	no
tabStops	no	no
target	no	no
targetType	no	no
textEncoding	no	no
textEntry	no	no
textIndent	no	no
textLocation	no	no
thickness	no	no
this	no	no
timeout	no	no
timeStamp	no	no
title	no	no
topInset	no	no

Scripting property	Supported in HTML	Supported in form guides
trailer	no	no
transferEncoding	no	no
transient	no	no
truncate	no	no
type	no	no
typeface	no	no
underline	no	no
underlinePeriod	no	no
upsMode	no	no
url	no	no
urlPolicy	no	no
usage	no	no
use	no	no
usehref	no	no
uuid	no	no
validationMessage	yes	Not applicable
validationsEnabled	yes	yes
vAlign	no	no
value	no	no
valueRef	no	no
variation	no	no
version	no	no
vScrollPolicy	no	no
w	yes	Not applicable
weight	no	no
wideNarrowRatio	no	no
wordCharacterCount	no	no
wordSpacingMaximum	no	no
wordSpacingMinimum	no	no
wordSpacingOptimum	no	no
x	yes	Not applicable

Scripting property	Supported in HTML	Supported in form guides
xdpContent	no	no
y	yes	Not applicable

6 Scripting Methods

This section lists support for XML Form Object Model scripting methods in HTML forms and form guides.

For more information on scripting methods, see *LiveCycle Designer ES Scripting Reference*.

Note: Items marked as not applicable will not generate an error or cause a script to fail; however, they do not produce any effect on the rendered form or form guide.

Scripting method	Supported in HTML	Supported in form guides
absPage	no	no
absPageCount	no	no
absPageCountInBatch	no	no
absPageInBatch	no	no
absPageSpan	no	no
addInstance	yes	yes
addItem	yes	Drop-down lists only
addNew	no	no
append	no	no
applyXSL	no	no
assignNode	no	no
beep	no	no
boundItem	no	Drop-down lists only
cancel	no	no
cancelBatch	no	no
clear	no	no
clearErrorList	no	no
clearItems	yes	Drop-down lists only
clone	no	no
close	no	no
createNode	no	no
delete	no	no
deleteItem	no	Drop-down lists only
documentCountInBatch	no	no

Scripting method	Supported in HTML	Supported in form guides
documentInBatch	no	no
emit	no	no
enumerate	no	no
evaluate	no	no
execCalculate	yes	yes
execEvent	yes	yes
execInitialize	yes	yes
execute	no	no
execValidate	yes	yes
exportData	no	no
first	no	no
formNodes	no	no
getAttribute	no	no
getDelta	no	no
getDeltas	no	no
getDisplayItem	no	Drop-down lists only
getElement	no	no
getFocus	no	no
getItemState	no	Drop-down lists only
getSaveItem	no	Drop-down lists only
gotoRecord	no	no
gotoURL	yes	no
hasDataChanged	no	no
importData	no	no
insert	no	no
insertInstance	yes	yes
instanceCount	yes	yes
isBOF	no	no
isCompatibleNS	no	no
isEOF	no	no
isPropertySpecified	no	no

Scripting method	Supported in HTML	Supported in form guides
isRecordGroup	no	no
item	no	no
last	no	no
loadXML	no	yes
messageBox	yes	yes
metadata	no	no
moveCurrentRecord	no	no
moveInstance	yes	yes
namedItem	no	no
next	no	no
open	no	no
openList	no	no
page	no	no
pageContent	no	no
pageCount	no	no
pageDown	yes	no
pageSpan	no	no
pageUp	yes	no
previous	no	no
print	no	no
recalculate	no	yes
record	no	no
relayout	no	no
relayoutPageArea	no	no
remerge	no	yes
remove	no	no
removeAttribute	no	no
removeInstance	yes	yes
requery	no	no
reset	no	no
resetData	yes	no

Scripting method	Supported in HTML	Supported in form guides
resolveNode	no	yes
resolveNodes	no	yes
response	no	no
restore	no	no
resync	no	no
saveFilteredXML	no	no
saveXML	no	yes
selectedMember	no	no
setAttribute	no	no
setElement	no	no
setFocus	yes	no
setInstances	yes	yes
setItems	no	no
setItemState	no	Drop-down lists only
sheet	no	no
sheetCount	no	no
sheetCountInBatch	no	no
sheetInBatch	no	no
sign	no	no
update	no	no
updateBatch	no	no
verify	no	no