

# Migrating from Adobe Acrobat forms to XML forms

**TABLE OF CONTENTS**

- 1 Description
- 1 Scripting methods and properties
- 1 Looking at Acrobat
- 2 Mapping examples

**Description**

Support for form fields was first introduced in Adobe® PDF version 1.3 and Adobe Acrobat® 4.0 software. AcroForms is the name of the dictionary in the PDF specification that contains references to the fields in a PDF document.

The form field objects are displayed in a “top layer” in the PDF document, separate from the PDF content layer (containing the images, fonts, words, and so forth). With the introduction of the PDF 1.5 specification and Acrobat 6.0, a new forms dictionary was introduced. In the PDF specification, this is referred to as the XFA dictionary, but the term XML forms is used here to differentiate between the older forms model and the new one. This new model can be accessed in Acrobat 6.x and 7.x and implemented with Adobe LiveCycle™ Designer software.

This tip discusses converting JavaScript and actions created in Acrobat (using AcroForms) to work in Adobe LiveCycle Designer (using XML forms).

**Scripting methods and properties**

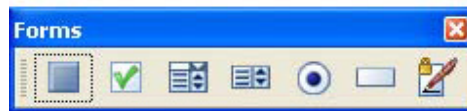
Adobe Acrobat and Adobe Reader® software have a built-in JavaScript engine. The JavaScript methods and properties available in Acrobat as AcroForms are described in the [JavaScript Scripting Reference guide](#). The JavaScript methods in Adobe LiveCycle Designer are described in the online Help accompanying that software and in the [Adobe XML Form Object Model Reference](#) and the [Calculations and Scripts document](#). Although many of the JavaScript methods and properties map one to one between LiveCycle Designer and Acrobat, some methods do not map directly. Let's take a look at the following examples.

**Looking at Acrobat**

In Acrobat 6 and 7, there are seven form field types (figure 1):

- ▶ Button
- ▶ Check box
- ▶ Combo box
- ▶ List box
- ▶ Radio button
- ▶ Text field
- ▶ Digital signature field

Figure 1. Forms toolbar in Acrobat Professional



Each of these objects can contain JavaScript commands and/or actions. LiveCycle Designer will open any PDF form created with Acrobat products version 3.5 and higher. LiveCycle Designer replicates all form fields and the page content. JavaScript and actions are saved as comments.

In this example, a button with a MouseUp action was created. In one instance, the action was set to Execute Menu Item > View > Go To > Next Page (Figure 2a). In another instance, the action was set to Run a JavaScript: this.pageNum++; (Figure 2b).

Figure 2a. MouseUp action: Execute Menu Item to go to next page

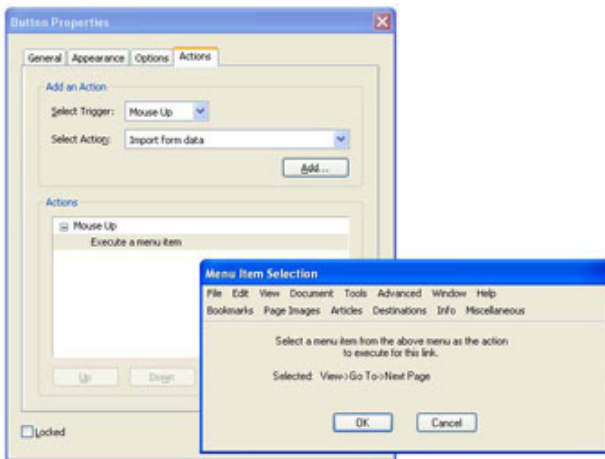
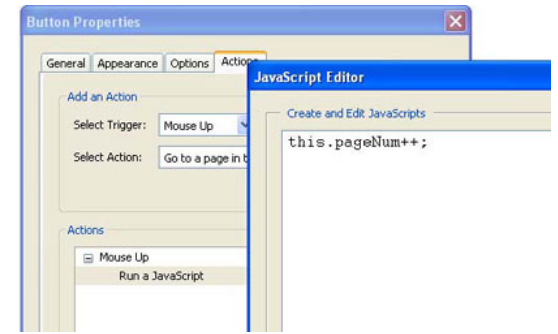


Figure 2b. MouseUp Action: JavaScript to go to next page



When the PDF file is opened and converted in LiveCycle Designer, the following code appears in the Script Editor in the mouseUp action:

For Execute Menu Item (Figure 2a):

```
comment();
Named: Flags=0 Url=NextPage
```

For JavaScript (Figure 2b):

```
comment();
this.pageNum++;
```

You have to convert these comments to JavaScript commands that can be interpreted by the XML Form model. The good news is that since every Adobe LiveCycle Designer form is an XML file, you can use simple search/replace methods to quickly update your scripts. Just save the form as XDP (XML Data Package) and view it in any text editor. The entire form, including the scripting, is represented as XML.

For this example, the XDP looks like this:

```
<event activity="mouseUp">
  <script contentType="application/x-javascript">
    comment();
    this.pageNum++;
  </script>
</event>
```

When a simple search/replace is run on the file, you get:

```
<event activity="mouseUp">
  <script contentType="application/x-javascript">
    xfa.host.pageDown();
  </script>
</event>
```

And when saved as a PDF file, the button action matches the original Acrobat created form.

## Mapping examples

Here are some examples of mapping from AcroForms to the XML Forms model.

ACTION	ACROFORM	XML FORM
Navigation to the Next Page	<code>this.pageNum = this.numPages++;</code>	<code>xfa.host.pageDown();</code>
Reset Form Fields	<code>this.resetForm();</code>	<code>xfa.host.resetData();</code>

Stay tuned for a more complete list and additional information.

### FOR MORE INFORMATION

If you'd like to provide feedback on this tip or if you have questions, send e-mail to [Lori DeFurio](mailto:Lori.DeFurio@adobe.com), Developer Evangelist.

Adobe Systems Incorporated • 345 Park Avenue, San Jose, CA 95110-2704 USA • [www.adobe.com](http://www.adobe.com)

Adobe, the Adobe logo, Acrobat, Adobe Portable Document Format, Reader, LiveCycle, Adobe Intelligent Document Platform are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. All other trademarks are the property of their respective owners.

© 2005 Adobe Systems Incorporated. All rights reserved.5/05