Introduction to XForms

The new Web standard for forms

Chris Herborth

September 12, 2006

XForms is the next generation of Web-based data processing. It replaces traditional HTML forms with an XML data model and presentation elements. This article covers how XForms actually works, and shows you how to set up XForms with Firefox and Microsoft® Internet Explorer so that you can view your XForms samples.

Introduction

XForms is gaining momentum rapidly, with support available for common browsers using extensions or plugins, and through things like the IBM® Workplace Forms technology (see the Related topics section to find out more). Its flexibility and power make it attractive to Web developers, and its small footprint and client-side processing make it attractive to systems administrators. The W3C is currently reviewing XForms 1.1 as a Working Draft document (1.0 is an official Internet Recommendation, which puts it on par with things like XHTML, PNG, and CSS), and IBM is currently spearheading an effort to merge competing XML-based forms' standards with the features and abilities of XForms.

This article explains the evolution of forms from simple text searches to today's interactive masterpieces and explains the next step in Web forms, XForms. It explains what makes XForms different and gets the user ready for Part 2 by setting up his or her environment.

Read on to see how a simple HTML form can be replaced with a simple XForms form, and how you can use XForms' capabilities to save (and reload later) a form's data to a local file.

If you'd like to follow along with the XHTML and XForms documents discussed in this article, grab the archive linked in the Downloadable resources section. It contains the files you're going to look at, which you can use as a starting point for your own XForms experimentation.

You'll also be installing an XForms extension that lets you view XForms documents directly in current versions of Firefox, Internet Explorer, Seamonkey, or Mozilla. If you don't currently have one of these excellent, standards-compliant browsers installed, now's a good time to get one.
Why XForms?
Before jumping into XForms, let's get a little historical perspective on forms.

Early in the history of the World Wide Web, simple text entry forms were used to send requests to CGI programs that ran on the server, bringing some additional interactivity to those pioneers venturing out into the unknown. As the Web matured, the data entry forms presented by Web sites grew more complex.

These days, you can easily find extremely complex forms, especially on e-commerce sites or forums. These can be front ends to PHP or Java™ code running on the server, or JavaScript running in the browser, sending its results to the server using the Ajax techniques.

Unfortunately, HTML's forms are created from elements that intermingle presentation and content. With the move from sloppy presentation-based HTML mixing the document structure with formatting elements, to clean XHTML styled with CSS, you need a way of separating a form's data and behavior from its presentation.

Enter the browser
The XForms standard, which has been in discussion for several years now, is supported by quite a few applications (including Firefox, Mozilla, and Seamonkey if you install the XForms extension), and is intended to be an XML replacement for the chaos of HTML forms.

Other popular Web browsers support XForms through plugins (such as formsPlayer for Internet Explorer 6, available from Related topics). Any standards-supporting browser can also make use of FormFaces, a pure JavaScript XForms implementation that translates XForms to and from standard HTML controls on the client system. Additionally, major products such as IBM Workplace Forms implement XForms.

A form's data model (how it behaves when data is submitted, initial values, and more) is separated from its presentation. The same presentation can be interpreted differently based on a style sheet, or whatever is appropriate for the user's browser. A browser running on a cell phone might, for example, use different techniques to display the controls; menus and lists are easier for cell phone users than pop-up choice boxes.

How XForms work
Let's take a look at a simple HTML form that submits a search string to an imaginary query engine running on your local machine (see Listing 1). Note that this is actually XHTML rather than HTML 4.01; XForms requires valid XML documents (such as XHTML) and you'll be seeing an XForms version shortly. Also note that you may need to tweak your URLs to get them to work properly in your own environment.
Listing 1. A simple HTML form (simple.xhtml)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Search Form</title>
</head>
<body>
<h1>Search Form</h1>
<p>Enter a search string, then click the Search button.</p>
<form action="http://localhost/imaginary-search-engine" method="get">
  <p>Find: <input type="text" name="query"/>
  <input type="submit" value="Search"/>
  </p>
</form>
</body>
</html>
```

Listing 1 is standard, 100% valid XHTML Strict for your viewing pleasure. The `<form>` section has been highlighted in bold.

When rendered in a typical current Web browser, this XHTML looks quite familiar, if a bit primitive (Figure 1).

**Figure 1. A simple HTML form rendered by Firefox**

![Search Form](image)

When you enter some text in the field shown in Figure 1 and click the button, the data is encoded into the URL sent to the imaginary search engine: `http://localhost/imaginary-search-engine?query=whatever-you-entered`.

Listing 2 shows you the XForms version, with the differences highlighted.

Listing 2. A simple XForms form

```xml
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xf="http://www.w3.org/2002/xforms">
<head>
<title>Search Form v2</title>
</head>
<xf:model>
  <xf:submission action="http://localhost/imaginary-search-engine"
    method="get" id="submit-search"/>
</xf:model>
</html>
```
<xf:model>
</head>
<body>
<h1>Search Form</h1>
<p>Enter a search string, then click the Search button.</p>
<p><xf:input ref="query"><xf:label>Find:</xf:label></xf:input>
<xf:submit submission="submit-search"><xf:label>Search</xf:label></xf:submit>
</p>
</body>
</html>

There are three main changed areas here:

- The "xf:" namespace adds the elements from the XForms specification.
- An <xf:model> element has been added to the <head>, which declares an <xf:submission>. This binds the action (your imaginary search engine) to the get method and names it submit-search for later use.
- The form's presentation elements (<xf:input> for the text input field, and <xf:submit> for the submission button) are now just presentation elements, indicating where and how the form's fields should appear in the rendered page. They refer back to the model declared in the <xf:model> element.

And here’s how the XForms version looks (see Figure 2). Installation of the XForms extension for Firefox is covered later.

**Figure 2. A simple XForms form rendered in Firefox**

The forms look exactly the same! You lose absolutely nothing by using XForms instead of traditional HTML forms, and you've gained all the advantages of using an XML data model.

**Setting up to view XForms**

If you've downloaded the example archive (found in the Downloadable resources section), you might have noticed that your browser won't display the XForms version properly (see Figure 3).
Figure 3. Your simple XForms page on a browser without XForms support

Search Form

Enter a search string, then click the Search button.

Find: Search

Until your favorite Web browser offers built-in support for XForms, you'll have to use one of the XForms implementations (see the XForms Web site listed in Related topics). Let's take a quick look at installing one of the most convenient implementations, the Mozilla extension. Since this works with current versions of Mozilla, Firefox, and Seamonkey, you probably won't even have to install or switch browsers! You'll also look at installing the Internet Explorer 6 plugin.

Adding the XForms extension to Firefox

In Firefox, pull up the Mozilla XForms Project page (see Related topics). Click the download page link you'll find under the Latest Release heading.

The downloads page for the XForms Project will warn you about the plugin not being ready for end-users, and then give you a link to the actual download page. Click that, then click the Install Now link.

Figure 4. Firefox's standard warning about installing extensions

Firefox displays a warning dialog (see Figure 4); click the Install Now button to download and install the XForms extension. Restart Firefox to activate it (see Figure 5).
Figure 5. Restart to activate your new extensions

After restarting your browser, you can load the simple XForms example that was looked at earlier; this time you'll actually see the XForms elements rendered properly!

Adding the XForms plugin to Internet Explorer 6

On the formsPlayer Web page (see Related topics for a link), click the Download fP for free link to access the registration page. Enter your e-mail address and other data, then click the Install formsPlayer link to download and install the plugin.

When IE loads the plugin archive, it will display a warning under the URL about the ActiveX control. Click this warning, then select Install ActiveX Control. IE will download and install formsPlayer. When IE's Security warning pops up, click Install to go ahead with the installation.

Once formsPlayer is installed, you'll be redirected to a page with links to several examples. View any of them to verify your installation.

One form, several presentations

Because the data model and presentation are separated in an XForms document, the user's browser can render the form's elements using whatever makes the most sense and provides the most usability.

For example, if the user is asked to select one item from a list, a Web browser might present the list as a standard radio button group. In your cell phone's browser, this same list could be rendered as a pop-up menu, which would be easier to navigate using the phone's direction wheel.

This doesn't require any special CSS or even JavaScript to detect the user's browser (an error-prone mistake many Web sites make), it's handled automatically by the browser's XForms implementation. Adapting like this is difficult with plain HTML forms because the data and presentation are mixed together too tightly.

Because the XForms model is XML, you can also use technologies like XSLT or JavaScript (through the DOM) to present that model using any interface you'd like, from plain HTML or XHTML for users without XForms support, to proprietary GUIs.
Tight integration with XML
XForms documents must be valid XML (such as XHTML), and instances of the data model (used to provide default values, among other things) are XML documents. You can even load data model instances from external XML files loaded on the user's machine or the server.

This tight integration with XML makes it easy to integrate your XForms data with other technologies that support XML input and output, such as databases, without the need for additional scripting between the database and the form being presented to the user.

Save forms offline and submit them later
One of the interesting features that XForms brings to your browser is the ability to save a form's contents (any data in the fields, plus any "hidden" data stored in the data model) to a file. By providing additional <xf:submission> elements with a put method, you can store this data as an XML file in your local filesystem or on the server. You can then load it again later to continue where you left off, all without having to use additional scripting or database access.

This also shows you another of the XForms advantages; you can have any number of submission actions associated with a single form (see Listing 3). In HTML, you can only have one action per form.

Listing 3. Adding some data saving and loading to the form

```xml
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
     xmlns:xf="http://www.w3.org/2002/xforms">
<head>
<title>Search Form v3</title>
<xf:model>
    <xf:instance>
        <data xmlns="">
            <query>Enter your search string here.</query>
        </data>
    </xf:instance>
    <xf:bind noderset="query" required="true"/>
    <xf:submission action="http://localhost/imaginary-search-engine"
                    method="get" id="submit-search"/>
    <xf:submission action="file:///tmp/XFormsQuery.xml"
                    method="put" id="save-search" replace="none"/>
    <xf:submission action="file:///tmp/XFormsQuery.xml"
                    method="get" id="load-search" replace="instance"/>
</xf:model>
</head>
<body>
<h1>Search Form</h1>
<p>Enter a search string, then click the Search button. You can also save your search string for later.</p>
<p><xf:input ref="query"><xf:label>Find:</xf:label><xf:input>
<xf:submit submission="submit-search"><xf:label>Search</xf:label></xf:submit>
</p>
</xf:input>
<xf:submit submission="save-search"><xf:label>Save</xf:label></xf:submit>
```
Again, the changes from your previous XForms experiment have been highlighted in bold. Let's take a more detailed look at the changes.

First of all, you've added a data model instance (inside the `<xf:instance>` element). This contains a root for the data (the aptly-named `<data>` element; note that the empty xmlns attribute is a requirement here), which in turn contains the data you need for your form, the `<query>` you reference down in the form's `<xf:input>` element. The data found in this `<query>` element is the default query string, and will be displayed in the `<xf:input>` element when it gets rendered.

Next, there's an `<xf:bind>` element, which makes the `<query>` data required; if the `<query>` is empty, none of the `<xf:submit>` buttons will do anything.

Last, you've added two new `<xf:submission>` actions to the `<xf:model>` that let you save your data to a file (via the put method) and load it from a file without reloading the entire page (using the get method; the replace="instance" attribute reloads the data model instance only).

At the bottom of the page two more `<xf:submit>` elements have been added so you can save and load the data.

Figure 6 shows you what this now looks like in Firefox. Note how the Find field is filled in from the default that you've included in the data model instance and the new Save and Load buttons.

**Figure 6. The new and improved form**

If you click the **Save** button at this point, you'll get a file named XFormsQuery.xml in your /tmp directory containing the data model instance (see Listing 4). To test the Loading functionality, change the Find string, and then click **Load**; your Find string will be replaced with the original one, loaded from the XFormsQuery.xml file in /tmp.
Listing 4. The saved data model instance

chrish@dhcp1 [531]$ cat XFormsQuery.xml
<?xml version="1.0" encoding="UTF-8"?>
<data xmlns:xf="http://www.w3.org/2002/xforms">
  <query>Enter your search string here.</query>
</data>

There are some options for controlling the output, but you haven't used any of them here. In any case, the Save and Load functions are working, without any sort of scripting.

Summary

The XForms standard adds a powerful new abstraction between a form's data model and its presentation, as well as providing the ability to use a single form with several different submission actions. XForms elements can also validate data and the tight integration with XML lets you work directly with data from any other XML-aware application. Saving a form's data for use later is also easy with XForms.

Stay tuned for Part 2 to learn how to create a basic XForms model and form.
### Downloadable resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 1 sample code</td>
<td>xforms1.zip</td>
<td>2KB</td>
</tr>
</tbody>
</table>
Related topics

• Visit the XForms home at W3C.
• Get the XForms extension for Mozilla, Firefox, or Seamonkey.
• Get the Trial: Lotus Forms (formerly Workplace Forms product).
• Read about Web Forms 2.0, a similar but competing spec.
• Get ready for XForms (devleoperWorks, Septebmer 2002) provides a useful overview of this important technology.
• For a good starting point for XForms research, read Ten Favorite XForms Engines from XML.com.
• For an interactive, cross-platform tutorial on W3C XForms, visit the XForms institute.
• Get a historical perspective on the origins and purpose of XForms in this brief history of SGML.
• To get a good grounding in XML, read the Introduction to XML tutorial (developerWorks, August 2002).

© Copyright IBM Corporation 2006
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)