Chapter 8: Understanding Xforms

In this chapter, we cover the newest technology in Web forms, XForms. And when we say new, we mean really new. XForms is currently a "work in progress" at the W3C. So, in this chapter, we just provide you with a very brief overview of XForms. For the latest and greatest, visit the W3C site at http://www.w3.org/MarkUp/Forms/.

The History of Web Forms

Forms provide a portal for electronic commerce (e-commerce) on the Internet. Unless you create scripts to handle user interaction, forms are one of the few ways XHTML Web pages can interact with users—and even then, you're most likely working with scripting to handle that data. Forms were introduced in 1993 as a formal part of HTML, and they haven't changed that much since then. What has changed, however, is their importance on the Web. HTML Web forms are now used by companies world-wide to drive their e-commerce applications. Basically, HTML Web forms are outdated and need a face-lift because they don't provide the functionality needed to fulfill the requirements of complex e-commerce sites.

However, before you dive into the newest proposals for Web forms, namely the W3C's XML Forms (XForms), take a second to visit where forms are now.

Why Use Forms at All?

Forms are simply a way to enable users to interact with (pass information to) a Web server. There are many reasons you might choose to use forms.

The simplest of forms are often created to solicit feedback from users. Feedback is a necessary and priceless resource for any Web page and should be used to drive current and future directions of that page—or possibly the entire Web site.

More complicated forms enable users to check their bank account balances, purchase airline tickets, and check their email. Form controls, combined with scripts, enable users to communicate with your server and exchange information. This process drives e-commerce. Think about the last time you purchased something over the Web—for example, an airline ticket. First, you must identify the flight you want to take. To do so, you fill out a form that helps you narrow your search. See Figure 8.1 for an example from Southwest Airlines (http://www.southwest.com/).

Figure 8.1
The reservations section of Southwest Airlines, where you enter information that narrows down the available flights for your specified preferences.

© 2000 Southwest Airlines Co. All content property of Southwest Airlines.

On the page displayed in Figure 8.1, you enter your departure city, arrival city, and desired departure and return dates. That information is sent to a Web server application that evaluates the data and sends you the schedule possibilities. See Figure 8.2 for the results we received when we requested information for flights between Austin, Texas and New Orleans, Louisiana for July 11, 2000.
Figure 8.2
The Web server application sends the user all flight possibilities.
© 2000 Southwest Airlines Co. All content property of Southwest Airlines.

After that, you’re presented with the option of purchasing a ticket. Again, you have to fill out a form. After you complete the form, that data is sent to a server and an appropriate response (hopefully, a purchase confirmation) is sent back to you, the user.

Using Forms Today

This chapter is a primer on XForms; however, it’s important that you understand and are able to use current form applications. Table 8.1 lists all XHTML elements and attributes used to create forms. For detailed descriptions of these elements and attributes, see Chapter 3, "Overview of Element Structure."

Table 8.1 XHTML Form Elements and Attributes

<table>
<thead>
<tr>
<th>Element Name</th>
<th>Empty</th>
<th>Description</th>
<th>Possible Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>button</td>
<td>No</td>
<td>Creates an input button</td>
<td>disabled, name, tabindex, type, value</td>
</tr>
<tr>
<td>fieldset</td>
<td>No</td>
<td>Groups related form controls</td>
<td>None</td>
</tr>
<tr>
<td>form</td>
<td>No</td>
<td>Contains form block</td>
<td>accept, accept-charset, action, enctype, method</td>
</tr>
<tr>
<td>input</td>
<td>Yes</td>
<td>Defines type and appearance for input objects</td>
<td>accept, align, alt, checked, disabled, maxlength, name, readonly, size, src, tabindex, type, usemap, value</td>
</tr>
<tr>
<td>isindex</td>
<td>Yes</td>
<td>Solicits a single line of input from users</td>
<td>prompt</td>
</tr>
<tr>
<td>label</td>
<td>No</td>
<td>Identifies form controls</td>
<td>accesskey, for</td>
</tr>
<tr>
<td>legend</td>
<td>No</td>
<td>Provides a caption to a set of related form controls</td>
<td>align, accesskey</td>
</tr>
<tr>
<td>option</td>
<td>No</td>
<td>Assigns a value to an input item</td>
<td>disabled, selected, value, label</td>
</tr>
<tr>
<td>optgroup</td>
<td>No</td>
<td>Groups selection choices logically</td>
<td>disabled, label</td>
</tr>
<tr>
<td>select</td>
<td>No</td>
<td>Creates a menu or</td>
<td>disabled, multiple,</td>
</tr>
</tbody>
</table>
Example 8.1 is a very simple form that shows some of these elements in action.

**Example 8.1 A Simple XHTML Form Enables Customer Data to Be Sent to a Server**

```xml
<?xml version="1.0"?>
<!DOCTYPE html
 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Title of the document.</title>
</head>
<body>
  <form action="URL" method="post">
    <p>This is a simple form.</p>
    <fieldset>
      <legend>Customer Information</legend><br />
      Last Name: <input name="lastname" type="text" tabindex="1" />
      <br />
      First Name: <input name="firstname" type="text" tabindex="2" />
      <br />
      E-mail Address: <input name="address" type="text" tabindex="3"/>
    </fieldset>
  </form>
</body>
</html>
```

The `input` elements collect data—in text form—from the user. That data is then instructed by the `form` element to be passed to a Uniform Resource Locator (URL) to be processed. That URL points to a server application ready to process the data. It's as easy as that.

To see how this example renders in a browser, see Figure 8.3.

**Figure 8.3**
A simple form displayed in a browser.

**Present-Day Limitations**

Currently, Web forms are used as a way to exchange bidirectional data over the Web. However, as e-commerce grows on the Web, so does the need to offer more complex ways to exchange data. For example, Company A might want to place a purchase order with Company B. However, Company A might want to place some conditions on that exchange. Without a common language, namely XML, this process gets difficult and requires serious programming. Even then, many limitations exist.

In addition to the limitations of data exchange, there are other concerns regarding present-day HTML
limitations: namely, the capability to separate data from presentation. Current form controls tie presentation closely to the involved data. Although this may not pose a serious problem with simple forms, such as the one in Example 8.1, it does create a mess when complex forms are needed to drive e-commerce for major companies.

After all, forms have not seen the editing block for over seven years!

**What Exactly Are XForms?**

The goal of XForms resembles the goal we’ve mentioned for XML throughout this book: to separate display from data. The hope for XForms is that it will separate the user interface from the data and logic, allowing the same form to be completed by users on a computer desktop, personal digital assistant (PDA), or mobile phone.

The following are the goals of XForms as stated by the W3C’s XForms Working Group (you can find this list and much more at [http://www.w3.org/MarkUp/Forms/](http://www.w3.org/MarkUp/Forms/)):

- Support for handhelds, television, desktop browsers, printers, and scanners
- Richer user interface to meet the needs of business, consumer, and device control applications
- Decoupled data, logic, and presentation
- Improved internationalization
- Support for structured form data
- Advanced forms logic
- Multiple forms per page, and pages per form
- Suspend and resume support
- Seamless integration with other XML tag sets

The first step to creating a new standard is to identify goals. Next, you need to create a standard to actually achieve those goals. At the time of this writing, the W3C is in the process of mapping out just how to achieve the goals outlined in the previous list.

It’s speculated that XForms will be three separate standards: one governing the data module, one governing the user interface, and one defining protocols. The first of the three has already made it to the Working Draft stage at the W3C and is appropriately titled "XForms 1.0: Data Model." (This information is based on the August 15, 2000 Working Draft.) The other two standards are not written yet.

According to the W3C, forms will be split into three layers (which is the reason for the three separate specifications): presentation, logic, and data:
- **Presentation.** The actual markup used with basic form controls. Each form control will be tied to a field in the data model.

- **Logic.** Enables you to define relationships between fields; for example, you might want to require additional fields to be filled in if a particular field is filled in.

- **Data.** Enables you to define a data model for your form. You will be able to use built-in data types or create your own. This layer will also make it easier to validate form information (check that it’s a date, a number, and so on) without enormous amounts of scripting.

---

**Now That’s Progress!**

Just to give you a taste of what’s coming, one of the goals of the XForms specification is to enable users to print a form, fill it in by hand, and scan it in!

---

**Who Should Learn XForms?**

Anyone interested in creating complex forms, as well as programmers who are used to creating the server-side scripting required to accept, validate, and process incoming data, should look at the next generation of Web forms. For those of us who create Web pages and Web forms, it makes sense to take the next logical step and learn XForms. Because of XForms’s e-commerce potential, most companies will expect their Web developers to soon adopt the new XForms capabilities.

Note that application vendors also make up a large piece of the target audience. Until application vendors adopt the XForms data model, Web developers will be unable to capitalize on the new functionality. One vendor is already working on implementing forms: Mozquito. See the "For More Information" section for more details.

As stated previously, programmers who have to deal with server-side scripting that processes the data collected from the Web form will be interested in the new capabilities XForms offers. The hope is that XForms will provide a consistent, XML-based format for incoming data, as well as take advantage of XML’s validation framework—both of which will make programmers' lives easier.

**Creating XForms**

Because XForms is still in its infancy, and the standard is not yet set, you can't start working with XForms just yet. However, you can visit an example cooked up by the W3C's XForms Working Group. Keep in mind that it is an experimental example used by the Working Group to explore different ideas for XForms.

To visit the XForms Working Group example, visit [http://www.w3.org/MarkUp/Forms/Sample/acme.html](http://www.w3.org/MarkUp/Forms/Sample/acme.html).

**For More Information**
Because XForms is still in its infancy, not a lot of information is available. However, the W3C, as usual, does have information on XForms on its Web site. Here are some URLs to guide you in your discovery of XForms:

- **The W3C XForms Working Group.** [http://www.w3.org/MarkUp/Forms/](http://www.w3.org/MarkUp/Forms/)

- **XForms 1.0: Data Model specification.** [http://www.w3.org/TR/xforms-datamodel/](http://www.w3.org/TR/xforms-datamodel/)

- **Mozquito.** [http://www.mozquito.com/](http://www.mozquito.com/) This is an XHTML editor vendor that not only has a significant role in the creation of XForms, but also has created its own version of Web forms—Forms Markup Language (FML). FML is a collection of new elements for XHTML Web pages. Much of the functionality of XHTML-FML was folded into the W3C's coming version of XForms.

- **W3C’s public mailing list subscription.** Send email to [www-forms-request@w3.org](mailto:www-forms-request@w3.org) with the word *subscribe* as the subject. If you decide you want to be removed from the list, send email to the same address with the word *unsubscribe* as the subject.

---

**Footnotes**

1. Denotes a deprecated attribute

© Copyright Pearson Education. All rights reserved.