

XFormsInCocoon

XForms in Cocoon

Although [XForms](#) is the approved way to do forms in Cocoon, it is still possible to use W3C standard [<http://www.w3.org/TR/xforms>]. This page is an attempt to gather information on the various ways to use XForms in Cocoon, and how to decide whether you want to do so.

Additions and corrections are welcome.

Background

- XMLForm - a way to support XForms in Cocoon. Superseded by JXForms and now deprecated and removed.
- JXForm - a cleaner rewrite and update of XMLForm. Superseded by CForms (formerly Woody) and now deprecated and removed.

Whether to use XForms

Let me start by pointing readers to Sylvain Wallez's meaty [meditation](#) on the pros and cons of XMLForms (an implementation of XForms in Cocoon) and Woody (known today as CForms). Sylvain points out some reasons why he feels XForms is not adequate for the forms requirements of certain kinds of server-based web applications. However, Sylvain's comments apply to XMLForms, which was superseded by JXForms, and was not based on the final specification for XForms. Can anybody comment on whether the limitations of XMLForms that Sylvain referred to are intrinsic to XForms (the final W3C Recommendation version)?

I started a response, entitled "XMLForms is dead, long live XForms" in my blog at http://dubinko.info/blog/2004/10.html#perm2004-10-13_xmlforms I'm interested in any comments. Eventually, I will fold that entry into the Wiki. – [MicahDubinko](#)

Technical considerations aside, one significant reason to prefer XForms over CForms is that the latter is a Cocoon-only approach. Thus your investment in CForms is tightly coupled to your decision /ability to continue using Cocoon.

Kurt Cagle has [listed seven reasons](#) why he believes XForms is the most likely forms technology to integrate into the web infrastructure, and will start gaining real traction in the next year or two.

How to use XForms

Potential approaches

As I looked for XForms implementations, my ideal was an open-source package that would allow client-side XForms functionality without users having to install additional software (such as a plugin). Some of the options are (see <http://www.w3.org/MarkUp/Forms/#implementations> for a list of implementations):

- XMLForms was an XForms package that was part of Cocoon, but has been removed. (See <http://marc.theaimsgroup.com/?l=xm1-cocoon-dev&m=108876021513344&w=2>)
- There seems to have been a project called ExFormula, which, according to <http://cocoon.apache.org/link/projects.html>, was to implement XForms for Cocoon. However the sourceforge project page seems to lack anything beyond the initial "what it's going to do" statement.
- (Side note: The Web Service Proxy in Cocoon 2.1 depends on XMLForms, but the samples for the proxy in 2.1.5.1 did not work for me. I don't know if that is because XMLForms was removed, or what. If this sample is no longer expected to work, it should probably be marked as such. Too bad; it sounds like a useful thing.)
- [Chiba](#) is a servlet webapp (i.e. like a sibling of Cocoon rather than its child) that implements XForms. There is a project called Chicoon whose purpose is to integrate Chiba into Cocoon. The latest release of Chicoon is 0.9.9 which has been updated to the respective Chiba version. There's also an online [demo](#) which is available via the Chiba homepage. Chicoon comes wrapped inside a Cocoon Action and is therefore available in any pipeline although most of the time a single, generic pipeline will do. Use of Chicoon should be very straightforward after studying the samples coming with the distribution. No flowscript, no multiple files, no nothing.
- [xslt2xforms](#) 'is a cross browser "W3C Xforms" processor. This xsl stylesheet add the W3C Xforms support in your web project using Xhtml, Javascript and Css.' The latest release at time of this writing is beta (0.7.6).
- [FormsPlayer](#): a commercial browser plug-in, free with [FormsPlayer](#) branding
- **NEW:** (2006-01) With [Firefox](#) 1.5 and later, (a subset of) XForms is supposed to be supported natively. This is the first time to have XForms support in a non-beta browser. Download the extension from <http://www.mozilla.org/projects/xforms>. If I were doing this project over again, this feature of Firefox would receive careful attention. Of course, you still have to consider whether most of your users will be using IE; and hope that IE 7.0 will support much of XForms.
- **NEW:** (2007-01) Some other good alternatives:
 - [FormFaces](#) "FormFaces(TM) is a pure [JavaScript](#) solution that utilizes AJAX techniques and can be seamlessly integrated with AJAX applications. This means that XForms+HTML can be sent directly to the browser where [JavaScript](#) transcodes the XForms controls to HTML form controls and processes the binding directly within the browser - requiring ZERO server-side processing and ZERO plug-ins." (Check [feature implementation coverage](#).)
 - [AjaxForms](#) "AJAXForms is a new tool that transforms XHTML/XForms documents (browsers can be process them) to HTML with Javascript documents. These pages manage the presentation and the user interface logic. And they only use the server to interchange data using AJAX."
 - [Concentré](#) "'Concentré' is a Xforms/XML framework based on 3 differents projects : xslt2xforms, libxforms and on a MVC php 5 framework. The objective is to provide a complete xforms/xml toolbox based only on w3c standards." (Current release as of 2007-01: 0.2 beta)

Experiences with Mozquito DENG

Looking through the list at <http://www.w3.org/MarkUp/Forms/#implementations> (which still lists Cocoon XMLForms, btw), I found "Mozquito DENG zero-install XForms". (Another helpful XForms implementation list is at <http://www.xml.com/pub/a/2003/09/10/xforms.html?page=1>.)

The DENG blurb says "Mozquito's latest XForms implementation is fully client-side and written in Actionscript to allow instant deployment for any flash-6-enabled browser on the web... Supported are subsets of CSS2, CSS3, XHTML, XForms, XFrames and SVG."

I tried DENG and have had good success with it, with a few caveats. Good points:

- It's free.
- Doesn't require an explicit software download (if SWF is already there).
- DENG is tiny - about 76KB.
- Looks good.
- Cool idea - using a Shockwave movie as an XForms renderer.
- Lets you deploy XForms to users right away, without having to develop, or decide on, stylesheets for rendering.
- The rendering is customizable via CSS.
- It's pretty easy to generate the necessary code from Cocoon. You just generate an HTML page that contains an <object> tag; the latter has URLs for (a) the deng_test.swf renderer file on your server, and (b) the XForms XML document you want to use. Make sure you also put pipelines in your sitemap to serve requests for (a) and (b).

Demos of DENG can be found at [this tutorial](#) See e.g. http://claus.packts.net/deng/deng_test.php?f=examples/xforms/textarea.html&debug=1 See also <http://xformsinstitute.com/lesson1.php> at the XForms Institute.

OK, now the caveats:

- It's not highly obvious how to get your web pages to use DENG. However I wrote a tiny how-to which is currently at <http://claus.packts.net/deng/forum/viewtopic.php?t=201> and with that, it should be pretty straightforward. (To do: add a section at the end of this page describing how to use DENG in Cocoon pages. See also the last bullet above under "Good Points".)
- Only a subset of XForms is supported. I'm not sure what's left out; a list of supported features is available at <http://claus.packts.net/deng/features.php>. This list looks fairly inclusive. One thing that *is* missing is a variety of submit methods. DENG currently only seems to support the POST method, with XML data in the request body (i.e. you can't do a GET with the data as request parameters). For me this meant I had to learn about the [StreamGenerator](#) to get the form data into a pipeline.
- Users do have to have a Flash plugin installed. However it seems to me that most will already have it, or can easily do so.
- It's a little slow in "rendering" the forms... it can take a few seconds. But if you have something for users to look at at the top of the page and the XForm rendering at the bottom, it's not noticeable.
- If the form has to scroll, instead of using browser scrollbars at the side of the window where you expect them, you have SWF scroll buttons across the top and bottom of the window, which are not as easy to use.
- The rendering of the XForms is customizable with CSS, but if you needed to do more flexible transforms, a la XSLT, between the XForms markup and the rendering step, you don't have that option.

The caveats don't seem to bad for the time being. Moreover, I feel comfortable with them because I'm not locked in. If I decide on a different solution, I should be able to repurpose the XForms output to some other renderer, such as Chicoon, without much trouble. I haven't done this yet, but I will most likely create an "insert DENG object" transformer that isolates any DENG-specific code from the rest.

I also would like to try [FormsPlayer](#), which claims to be fully XForms compliant, and compare it with DENG. The drawback here of course is that the user has to explicitly download/install the FormsPlayer plugin, and it only works in IE apparently.

Feedback and synergy

I would be interested in hearing from others about their experiences with Cocoon and XForms.