

## Coldfusion 8's Catch 22 - Web 2.0 Widgets

Posted At : March 13, 2008 6:02 AM | Posted By : Mark Kruger

Related Categories: Coldfusion 8

Recently a colleague was lamenting the fact that the rich text editor that ships with CF8 seems to load slowly and have a few other problems that are causing consternation for his users. His exact comment was that he wished Adobe would "fix" or "patch" the FCK editor that underpins the rich text editor. I certainly sympathize with his desire to fix the behavior he is experiencing for his users, but I would hasten to point out that the underlying libraries for FCK editor are *not* from Adobe. Instead, along with copies other script libraries for Ajax, window behaviors and the like, they are open source libraries - many of them from Yahoo.

Of course this is one of the bugaboos of these new "client side" widgets that ship with Coldfusion 8 and in a way they put Adobe in a difficult position. In CF8, Adobe encapsulated a bunch of cool client side functionality into new tags like cfwindow, cfajax, the rich text editor etc. This is not new for Coldfusion - cfform, cfgrid, etc have been around for since the beginning. The biggest difference is that there is a lot more of them and they are exponentially more capable with the advent of Ajax, binding, and Cfajaxproxy. In fact, Ajax and Cfajaxproxy really continues the trend of blurring the line between the server and the browser - a trend already in full fury in the Flex world. Adobe is seemingly caught between the two worlds of client and server.

While they have a real "WOW" factor, the heavy reliance on JS in these client side implementations presents a problem. In my view Adobe will now be tasked with rolling code patches more quickly to keep up with browsers - a moving target to be sure. Server administrators who see a vendor issuing a copious number of patches will usually take it as something of a black mark and believe the product to be buggy. In the case of client side code "buggy" might mean IE is buggy or Firefox doesn't handle A B or C or Safari doesn't render X or Y. In other words, the bugs in question may have nothing to do with the server at all. Will Adobe development and QA, who are typically tasked with deploying carefully vetted Server patches for stuff like drivers and nuanced performance issues and the like, now need to roll patches every week to keep up with issues (albeit legitimate issues) of client side code?

Developers laud this blurring of the line between the server and the client. But I'm afraid they will fail to see issues with these implementations as part of the rough and tumble world of multiple browsers, security contexts, script settings and operating systems. Instead, the blame will fall to Adobe - in spite of their best efforts at using open source libraries etc (largely from Yahoo). I would note that this is why Yahoo has no phone number for support and why it is impossible to ever get more than a canned response for the email help system :)

I will say this by way of suggestion. Adobe should separate out builds for client side code from "server patches" and make it clear that some "hot fixes" are really new JS files from open source libraries. If it were me, I would not even call them "hotfixes" (which denotes a server bug in my mind). This will make server administrators breath easier. Since there is now so much of this code that ships with the server, it might be cool to see an update tool in the Admin - something like "update scripts directory from repository". Of course I suspect there are already folks out there who have hacked up that scripts directory to "fix" pet problems with these JS implementations - so that

might cause more problems than it solves.