

Why move to CFMX

Posted At : December 25, 2003 11:43 AM | Posted By : Mark Kruger

Related Categories: Coldfusion Upgrading

Picture this - a good customer comes to you with a complex application running nicely on Coldfusion 5.0. It has a few performance issues, but much of it is related to the code not the platform. He or she poses this question, "Why should I move to Coldfusion MX"? Hmmm... .

If you read the literature on CFMX you would think that the most compelling reason that someone would upgrade to CFMX is Java - more specifically J2EE. In his excellent presentation titled **Java and CFMX - Power and Productivity** the ubiquitous Ben Forta makes much of this point. And it *is* a salient point to many people. Companies looking to cut development time and expense who have *already* invested heavily in J2EE or are planning to do so may well salivate at the prospect of using CFMX on the presentation layer. I have found, however, that making this pitch to existing CF customers or small to medium businesses is a tough sell. If they are not already moving toward J2EE standardization, CFMX is not enough to push them in that direction.

That being said, there is a strong case to be made for migrating that has little or nothing to do with Java. Some might say it could be made in spite of Java - which (let's face it) adds complexity to the installation, configuration and troubleshooting of a MS web shop. On Macromedia's site there is a little article or snippet that goes something like "the Top ten reasons to migrate". It's a great starting point to make your pitch. Here's my approach for each category.

Superior XML support

CFMX has much *much* better XML support than any previous version of CF. Maybe you never tried to use XML in CF 5 or 4.5. I use to use a COM object to do it and the DOM model. Just getting at the data required way too much code and it was slow and balky - and COM on CF was always quirky for me.

XML in CFMX on the other hand is a breeze - especially if "grew up" with the previous versions of CF. An XML object is an array of structures - or a structure of arrays - or an array of structures of arrays that are arrays of struct.... well you get the idea. Sure there are new tags and methods that allow you to traverse the XML nodes and unpack them. But if you like doing things very "CF-like" you can just walk up and down using the rich structure/array function set that you are use to. The more "CFish" it is to learn - the quicker it is to pick up and use. With customers XML is still a buzz-word. Many things so "XMLish" that saying "XML support is waaay better" is a good argument.

Easy to consume and publish web services

You client may or may not be familiar with this concept. There are many ways to demonstrate- stock quotes to an Excel spreadsheet are particularly effective, as is the popular "babelfish" web service. If the client has any plans to integrate with other businesses or provide customers with non-web-page data about orders or products or services, its a good bet that this topic will make their ears perk up. If you are planning on pitching Flash remoting, cover this first - at least in part.

Better email support

2 items here seem to make clients giddy - the vast bulk email benchmarks (MM claims 1,000,000 an hour on a dual proc machine with adequate SMTP server support) and Multi-part messaging. Multi-part is important because it can greatly enhance the number of eye-balls that see your message without forcing you to resort to all plain text. Be warned however, SPAM is universally frowned upon by everyone these days - for good reason. If I read all spam I receive everyday I'd be unable to think about anything other than the size of my... mortgage payment (ha). Bulk emailing hasn't peaked as a marketing tool just yet, but it is definitely going to change and go the way of telemarketing (please Lord!).

OO Design features (CFCs)

This point is undersold to current CF 5.0 users. I make a good living out of charging into an underperforming customer site and re-organizing spaghetti code. CFML can really lend itself to messy coding practices. Your aps will work, but they will be a great cause for baldness later on. It *is* possible to build well organized and modular code in cf 5 - but it doesn't naturally lend itself to that task.

CFMX however has compelling reasons for building your code with:

- **Abstraction layers** - application logic and data access completely separate from presentation logic and templates.
- **Component objects** - Your events and data can be modeled as very accessible objects that can serve to support your web application or your flash movie or an XML feed or a public or private web service or the package it resides in or private methods of other objects in the application. In other words, separating the presentation layer - getting all of that messy tagish HTML out of the way - frees you up to create a Service Oriented Architecture (SOA).
- **Documentation** - All programmers hate documentation. They hate creating documents, adding comments and explaining in any way what they do. You might as well question God (ha). CFMX cfc's have a great feature where a programmer can add hints to just about everything in a CFC, then they can view a summary of methods and properties by opening the CFC directly in the browser. It works great and it's a godsend if you have over 15 or so methods or properties.
- **Automatic WSDL generation** - Any CFC that is web service enabled (a trivial task) can generate its own WSDL file with no tweaking and no understanding from the programmer.

If the client is a techy or has brought a techy with him, this is a great pitch.

Multiple Instances

If you have scalability problems that result from long running or weak processes in your web application (file dumps, FTP tasks, data migration, batch mail etc), the ability to have *Multiple instances* of CF server is a great selling point. With this ability you can isolate aps that cause the rest of your server to hang or become sluggish. Indeed, that "sluggish server" issue is one of the chief weaknesses of the CF 5 platform.

Faster Runtime

Yes it's true - CFMX is faster than any previous version of CF. In fact, CFMX 6.1 is significantly faster than CFMX 6.0 because the compiler no longer produces Java source code as an intermediate step in the compiling process. Keep in mind however, that customers struggling with performance will not find CFMX a panacea - especially if

their issue is poorly written code, bad database design or performance, or 3rd party dependencies beyond the control of the web server.

Better Access To Java

CFMX has faster and better access to Java subsystems, EJB's classes and the like. You can take advantage of that whole world of Java resources out there.

Flash Remoting

You can build an entire site using Flash as the UI design tool and CFMX as the integration and data tool. CFMX becomes (in this case) a data feed or RPC handler for whatever the client is doing. An amazing array of public web sites and intranet tools is growing around this approach. Flash is fast, the file size is very small (all things being equal) and the user experience can be quite good. It can also be bad - so look out. The Macromedia gallery is a great example of a *bad* RIA. You search for tags and components, then click on something to download it, then you have to go back to your search which is re-run (painfully slowly) again. If you are like me and you like to download a bunch of items all at once, you start to hate that little wheel of torture that rotates taunting you.

***Much* Better Charting**

The charting engine has improved dramatically. Particularly the ability to add more than one series (how did *that* get left out before?).

Double byte encoding and internationalization

CFMX now supports more language sets for better international support.

IDE Integration

CF 5.0 also had great integration with MM authoring tools. CFMX, however, includes WSDL introspection integration in Dream Weaver. That's a nice touch! You can make methods and properties accessible to your CF code through the typical properties panels you've come to expect from DW.

All of these points make excellent arguments when you are trying to sway someone to choose CFMX as a chosen platform.