Perspective on ColdFusion's Big Question (TM)

Posted At : January 2, 2009 7:45 AM | Posted By : Steven Erat
Related Categories: Personal, Flex, Java, Adobe, Events, Quality Assurance, ColdFusion, Computer Technology, New England

Just wanted to share a reply I made on GetSatisfaction to provide a historical perspective to the question '**What really is the future of ColdFusion?**". Before you ask what the future holds, its good to look back to see where ColdFusion has been since **its inception** in 1995.

CFMX 7 (released Feb 2005) was the release where product adoption saw the first major boost since the "MX" overhaul. Since CFMX 6 (released June 2002, in a down economy) was a re-architecture in Java/J2EE from the earlier CF5 (released May 2001) written in C++, there were few new features introduced and there was an associated learning curve now that the product had a Java foundation.

Problems in the re-architecture surfaced, slowing new adoption of CFMX6, leading to the point release 6.1 (released July 2003) which for the most part corrected all the issues and restored the waning product reputation.

ColdFusion MX 7 was a feature rich release, which attracted many new developers, most of whom had begun to grok CFCs and Java integration. The post 9/11 economy had generally recovered as well, adding to an increase in technology spending.

With most product release cycles, there's a decline in sales or tail at the end, and ColdFusion 8 (released August 2007) saw another major boost in adoption over the tail as it too was a feature rich release that provided solutions to many contemporary problems in Web Dev.

Frankly, IMO, nearly all negative connotations (i.e. "Legacy Software") about the ColdFusion Web Application Server are due to anachronistic experiences with earlier versions of the product in the mid/late 90's. Those opinions seem to be expressed from developers that are less familiar with the revisions and enhancements found in recent ColdFusion versions. (**Case in point**)

[Added note: *The easy learning curve, weak typing, and case-insensitivity in the product are among some factors that may have been conducive to poor programming practices... i.e. give them enough rope to hang themselves, so to speak. Does anyone remember memory corruption from not locking shared scope variables? That whole conundrum went away with CFMX*]

Personally, I think ColdFusion is a fantastic product and I love using it. It has an extensive, contemporary tag library on a stable Java base and Web application development time can be short and sweet due to its perpetual focus on RAD.

ColdFusion 9 is well known to be underway and will further address solutions to where technology is going. Furthermore, risk due to proprietary software is mitigated by the release of third party CFML engines which can provide a core of language features if not the full, rich diversity of language found in Adobe's product.

To throw in a plug for myself, I'm currently seeking full time, permanent employment in the greater Boston area. See: **Adobe Expert Seeking ColdFusion / Flex Dev or QA**