Essentials of ColdFusion

Contents

Articles

References

Article Licenses

ColdFusion Markup Language

Paradigm	imperative, object-oriented
Appeared in	1995
Designed by	Jeremy Allaire
Developer	Adobe Systems, Railo, New Atlanta
Major implementations	Adobe ColdFusion, Railo, BlueDragon
OS	Windows, Linux, UNIX, Macintosh
License	Depends on the implementation
Website	CFML Advisory Committee [1], Adobe ColdFusion [2], Railo [3], BlueDragon [4]

ColdFusion Markup Language, more commonly known as **CFML**, is the scripting language used by Adobe ColdFusion, BlueDragon and Railo, as well as other CFML server engines. The CFML language is guided by the CFML Advisory Committee [1].

Synopsis

CFML generally augments standard HTML files with database commands, conditional operators, high-level formatting functions, and other elements to produce web applications.[5]

The pages in a ColdFusion application include the server-side CFML tags in addition to HTML tags. When a web browser requests a page in a ColdFusion application, it is automatically pre-processed by the ColdFusion Application Server.[6]

CFML can also be used to generate other languages, aside from HTML, such as XML, JavaScript, CSS, and so on.

Despite the name, CFML is *not* a markup language. It is also not SGML, since certain core CFML features prevent it from complying.

ColdFusion tags tell the ColdFusion server that it must process the tagged information. The ColdFusion server only processes ColdFusion tag contents; it returns text outside of ColdFusion tags to the web server unchanged.[7]

History

The program was originally made by Allaire systems, based in Cambridge, Mass. The server-side technology was bought by Macromedia and became Macromedia Cold Fusion.

On June 18 2009, Adobe announced at the CFUnited conference that it had formed a CFML Advisory Committee [1][8] that would be responsible for guiding and reviewing changes to the CFML language.

Syntax

CFML tags have a similar format to HTML tags. They are enclosed in angle brackets (< and >) and generally have zero or more named attributes, though some tags (e.g. cfset, cfif) contain an expression rather than attributes. Many CFML tags have bodies; that is, they have beginning and end tags with text to be processed between them. For example:

```
<cfoutput>
   #value# Bob!
</cfoutput>
```

Other tags, such as cfset and cfftp, never have bodies; all the required information goes between the beginning (<) character and the ending (>) character, as in the example below. If it is legal for tags not to have a body, it is syntactically acceptable to leave them unclosed.

```
<cfset value = "Hello">
<cfset value = "Hello" />
```

Sometimes, although the tag can have a body, you find that you do not need to put anything in it because the attributes specify all the required information. In these cases (as with all HTML), you can choose to omit the end tag (and hence, the body) and just put a forward slash character before the closing (>) character, as in the following example:[9]

```
<cfexecute name="C:\winNT\System32\netstat.exe" arguments="-e"
outputfile="C:\Temp\out.txt" timeout="1" />
```

Various tags offer the ability to type-check input parameters (e.g. cffunction, cfparam, cfqueryparam) if the programmer declares their type specifically. This functionality is used with cfqueryparam to secure web applications and databases from hackers and malicious web requests.

Built-in tags

Over 80 built-in tags make up the heart of ColdFusion. The following lists CFML tags by their function or purpose.[10]

- Application framework
- Communications
- Control
  - Flow-control
  - Database manipulation
  - Exception handling
- Data output
- Debugging
- Display management
- Extensibility
- File management
- form
- Internet protocol
- Page processing
- Security
- Variable manipulation
- Other tags (cfimage, cfregistry etc)

Custom tags

CFML allows language extensions in the form of custom tags. In other words, CFML allows tags that are not built-in ColdFusion tags. Custom tags are normal files which are intended to be invoked as tags, although it is possible to treat a template as both a custom tag and a regular template. Custom tags written in CFML may be prefixed with *cf_*, although there are other ways to invoke them.

If a template is invoked as a custom tag, the attributes used to invoke that tag are available in a special structure *attributes* and the variables on the calling page are accessible via the *caller* struct. For example, if writing an *add* tag which takes two attributes and adds them together, the sum.cfm page would look like this:

```
<cfset caller.sum = attributes.first + attributes.second / >
```

Assuming the template and tag are in the same directory, the tag can be invoked thus:

```
<cf_sum first="1" second="2">
```

CFX tags are custom tags which are developed using Java language or C++, and are prefixed with cfx_ just like cf_. Tags are added to the ColdFusion runtime environment using the ColdFusion administrator, where JAR or DLL files are registered as custom tags.

JSP tags can also be included in CFML pages using the <cfimport> tag.

Functions

ColdFusion Markup Language includes a set of functions that you use to perform logical and arithmetic operations and manipulate data.

function	code
Array	[11] (ArraySort, ArrayAppend, ArrayDeleteAt...)
Conversion	[12] (URLEncodedFormat, ToString...)
Date and time	[13] (LsTimeFormat, DateAdd, DateDiff...)
Decision	[14] (IsDefined, IIF...)
Display and formatting	[15] (CJustify, NumberFormat...)
Dynamic evaluation	[16] (DE, Evaluate...)
Extensibility	[17] (CreateObject, ToScript...)
Image	[18] (ImageRotate, ImageAddBorder...)
International functions	[19] (SetLocale, GetTimeZoneInfo...)
List	[20] (FindOneOf, ListSetAt...)
Mathematical	[21] (Randomize, Sqr...)
Other functions	[22] (WriteOutput, GetBaseTemplatePath...)
Query	[23] (QueryAddColumn, QuerySetCell...)
Security	[24] (Encrypt, Decrypt...)
String	[25] (Reverse, HTMLCodeFormat...)
Structure	[26] (StructKeyExists, StructDelete...)

System	[27] (GetTickCount, GetTempFile...)
XML	[28] (XMLParse, GetSOAPResponse...)

ColdFusion Components (CFCs)

CFCs provide some (not all) of the typical features and functionality that are provided by object-oriented (OOP) languages. To create a CFC:

Create a file with a.CFC extension (this distinguishes CFCs from ColdFusion templates, which have a.CFM extension).

Use four tags to create the components, define their functions and arguments, and return a value.

<cfcomponent>: Defines a CFC

<cffunction>: Defines the functions (methods) within a CFC

<cfargument>: Defines the arguments (parameters) that a function accepts

<cfreturn>: Returns a value or result from a function

CFCs are plain CFML. Within a CFC you can use any tag, function, custom tag, component, and more. After creating your CFC, save it with.cfc extension.

To use your CFC, use <cfinvoke> tag to call your component methods from a.cfm file. <cfinvoke> takes the name of the component (minus the.cfc extension) and the method to execute. To access any returned data, the RETURNVARIABLE attribute provides the name of a variable to contain whatever the function returns. CFCs are created using four tags, saved as.CFC files, and invoked using the <cfinvoke> tag.[29]

In the example below, component temperature.cfc has a method FtoC which converts temperature from Fahrenheit to Celsius. The test.cfm template invokes the method and converts 212 degrees Fahrenheit and outputs the result.

```
<!--- temperature.cfc --->
<cfcomponent>
  <cffunction name="FtoC" access="public" returntype="numeric">
    <cfargument name="fahrenheit" required="yes" type="numeric"
/>
    <cfset answer= (fahrenheit - 32)*100/180 />
    <cfreturn answer />
  </cffunction>
</cfcomponent>
<!--- test.cfm --->
<cfset fDegrees = 212 />
<cfinvoke component="temperature" method="FtoC"
returnvariable="result">
  <cfinvokeargument name="fahrenheit" value="#fDegrees#" />
</cfinvoke>
<cfoutput>#fDegrees#&deg;F =
#result#&deg;C</cfoutput> <br />
```

External links

- CFQuickDocs—ColdFusion tags and functions reference [30]
- Adobe Dreamweaver CS4 - CFML IDE (Integrated Development Environment) [31]
- EasyCFM—ColdFusion Reference, Tutorials and Community Outreach site. Learn ColdFusion! [32]
- LearnCF - Learn ColdFusion with ColdFusion tutorials. Every tutorial has a working demo, code view and code download. [33]
- Free ColdFusion Hosting—the perfect way to learn ColdFusion, get free coldfusion [34]
- ColdFusion.TV— Free ColdFusion Video Tutorials! [35]

References

[1] http://www.opencfml.org/
[2] http://www.adobe.com/products/coldfusion
[3] http://railo.ch
[4] http://www.newatlanta.com/products/bluedragon/product_info/overview.cfm
[5] ColdFusion Markup Language (http://livedocs.adobe.com/coldfusion/6.1/htmldocs/introb8.htm)
[6] Michael Smith. "What is ColdFusion?" (http://www.fusionauthority.com/cfintro.cfm)
[7] Tags. (http://livedocs.adobe.com/coldfusion/6.1/htmldocs/element4.htm)
[8] http://corfield.org/entry/CFML_Advisory_Committee
[9] Tag syntax (http://livedocs.adobe.com/coldfusion/6.1/htmldocs/element5.htm)
[10] Tags by function. (http://livedocs.adobe.com/coldfusion/8/cf8_cfml_ref.pdf)
[11] Array functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_03.html#3473387)
[12] Conversion functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_04.html#1098761)
[13] Date and time functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_05.html#1098968)
[14] Decision functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_06.html#3485787)
[15] Display and formmatting functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_07.html#1099219)
[16] Dynamic evaluation functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_08.html#1099242)
[17] Extensibility (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_09.html#3490127)
[18] Image functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_11.html#1099325)
[19] International functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_12.html#3614227)
[20] List functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_13.html#1099435)
[21] Mathematical functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_14.html#1099613)
[22] Other functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_15.html#3493621)
[23] Query functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_16.html#1099653)
[24] Security functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_17.html#3542210)
[25] String functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_18.html#1099887)
[26] Structure functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_19.html#1099964)
[27] System functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_20.html#1100017)
[28] XML functions (http://livedocs.adobe.com/coldfusion/8/htmldocs/functions-pt0_21.html#3468770)
[29] Ben Forta, "Using ColdFusion components" (http://www.adobe.com/devnet/coldfusion/articles/intro_cfcs.html)
[30] http://www.cfquickdocs.com/
[31] http://www.adobe.com/products/dreamweaver/
[32] http://www.easycfm.com/
[33] http://www.learncf.com/
[34] http://www.freecoldfusionhosting.com/
[35] http://www.coldfusion.tv/

BlogCFC

BlogCFC is a popular open source weblog software for CFML, maintained by Raymond Camden. The latest version, v5.9.3, was released on 1 April 2009 and supports all the major blog features.

External links

- Official BlogCFC Website [1]

References

[1] http://blogcfc.com

BlueDragon

Developer(s)	New Atlanta Communications, LLC
Stable release	7.1 / June 23, 2009
Operating system	Cross-platform
License	Proprietary
Website	[1]

BlueDragon is a ColdFusion Markup Language (CFML) engine comparable to Adobe Systems's ColdFusion. It is licensed and distributed by New Atlanta from TagServlet Ltd based in Scotland. BlueDragon is also distributed and supported by BEA Systems on their Oracle WebLogic Server server platform.

BlueDragon applications run on a variety of platforms, including Microsoft Windows, Linux, and Mac OS X. It is mostly compatible with ColdFusion MX 7.

In March, 2008 New Atlanta announced that the future versions of BlueDragon (Java EE editions) will be released as open source products.[2]

Editions

BlueDragon is available in six editions:

- BlueDragon Server
- BlueDragon Server JX (similar to ColdFusion standalone editions)
- BlueDragon for J2EE Application Servers (BD J2EE)
- BlueDragon for the Microsoft .NET Framework (BD .NET)
- BlueDragon, BEA WebLogic Edition (sold by BEA as a BEA product)
- Open BlueDragon, an open source version of BlueDragon for J2EE.

The first two editions are standalone servers which run on Windows, Linux, and OS X. With BD J2EE, CFML applications can be deployed on any J2EE server, enabling integration of CFML and J2EE as enabled in ColdFusion MX. BD, BEA WebLogic Edition, is a special edition based on BD J2EE for use with BEA's WebLogic Server. BD.NET extends the Microsoft .NET framework and IIS to permit deployment of CFML applications as native MS .NET web applications, offering integration between CFML and ASP.NET that's not possible with ColdFusion.

The Server JX, J2EE, MS .NET, and BEA WebLogic editions of BlueDragon are commercial products which are available as 30 day unlimited trials which convert to a single-IP developer edition with no expiration.

The Server edition, on the other hand, is free for deployment -- though not for hosting, redeployment, or (as of the 6.2 release) commercial use. There are no differences in CFML tags supported in the free Server edition, but it supports only ODBC drivers on MS Windows (and only MySQL or PostgreSQL on Linux and Mac OS X), it supports only IIS on MS Windows or Apache on Linux or Mac OS X, and does not support secured (SSL) connections.

None of the limitations of the free Server edition exist in the commercial Server JX, J2EE, .NET, or BEA WebLogic editions.

The .NET edition of BlueDragon runs on Microsoft's .NET platform, BD.NET enables CFML applications to leverage the .NET platform and allows for integration between CFML and ASP.NET as well as .NET objects.

Open BlueDragon is an open source version of BlueDragon is released under the GNU General Public License version 3 (GPLv3). The chief differences between the open source and the J2EE version are the removal of

commercial libraries (e.g., for PDF generation), The JTurbo JDBC driver for Microsoft SQL Server, and the BlueDragon Administrator application. It runs on any standard J2EE application server, such as Tomcat, JBoss or Jetty.

The original version of BlueDragon was released in 2002.

Corporate adoption

MySpace, one of the most heavily visited sites on the Internet,[3] uses the .NET version of BlueDragon to power some of its online applications.[4] [5]

Compatibility

Though BlueDragon 7.0 was designed to be compatible with Adobe ColdFusion MX 7.0.2,[6] there are differences in the two CFML implementations. BlueDragon offers several advantages (tags, functions, and other functionality) not found in ColdFusion. Similarly, there are a few tags and functions found in ColdFusion that are not supported currently in BlueDragon. New Atlanta maintains a complete list of incompatibilities with Adobe ColdFusion MX in the documentation.

Notable differences compared to ColdFusion 8

- No support for on-demand presentations
- No built-in AJAX support
- No built-in support for exchange
- BlueDragon's .NET edition can create .NET objects
- No support for PDF Documents and Forms
- No built-in Server Monitoring and Alerts
- No support for AMF (Flash Remoting protocol)
- No support for event gateways
- No built-in stepthrough debugger
- No RDS Connectivity to server
- No Report builder or cfreport tag
- BlueDragon includes a tag for IMAP protocol
- BlueDragon cannot generate Flash movies, neither via *CFDOCUMENT* nor *CFFORM*. Version 7 will generate raster documents using *CFDOCUMENT*.

Framework compatibility

A number of popular ColdFusion frameworks are fully supported on BlueDragon:

- ColdSpring
- Fusebox
- Model-Glue
- Mach-II
- FarCry Framework

See also

For a list of useful resources for developers, see the ColdFusion Development aids section.

External links

- BlueDragon Product Documentation [7]
- New Atlanta BlueDragon website [8]
- Press release announcing BlueDragon version 7 [9]
- Interview with the creator [10]

References

[1] http://www.newatlanta.com/products/bluedragon/

[2] "New Atlanta to Open-Source Java Version of BlueDragon" (http://www.newatlanta.com/corporate/news/bluedragon_opensource_announce.jsp). New Atlanta. . Retrieved 2008-04-14.

[3] "Alexa Web Search - Top 500" (http://www.alexa.com/site/ds/top_sites?ts_mode=global&lang=none). Alexa.com. . Retrieved 2007-10-12.

[4] New Atlanta Communications, LLC (2005-06-28). "BlueDragon Powers The #1 CFML Website!" (http://www.newatlanta.com/corporate/news/myspace_announce.jsp). Press release. . Retrieved 2007-10-12.

[5] Dinowitz, Judith (2005-07-05). "BlueDragon.NET and MySpace.Com: An Interview with New Atlanta" (http://www.fusionauthority.com/community/4477-bluedragon-net-and-myspace-com-an-interview-with-new-atlanta.htm). Fusion Authority. . Retrieved 2007-10-12.

[6] (PDF) *BlueDragon 7.0 CFML Compatibility Guide* (http://www.newatlanta.com/products/bluedragon/self_help/docs/7_0/BlueDragon_70_CFML_Compatibility_Guide.pdf), New Atlanta Communications, LLC, 2007-01-26, pp. 3, , retrieved 2007-10-13

[7] http://www.newatlanta.com/support/bluedragon/docs/index.jsp

[8] http://www.newatlanta.com/bluedragon

[9] http://www.newatlanta.com/corporate/news/bluedragon_7_0_release.jsp

[10] http://alan.blog-city.com/interview_alanwilliamson.htm

CFEclipse

CFEclipse is a CFML plugin for the Eclipse platform. It includes many of the features common to modern IDEs such as code assist, context help, syntax highlighting, snippets, and more.

The plugin is released under the terms of the Eclipse Public License which is very similar to the Common Public License.

Versions

The current stable version of CFEclipse is 1.3 and is targeted at Eclipse 3.2. It is available from the Official CFEclipse website [1].

People

Mark Drew is the lead developer of the CFEclipse project

Technical mailing lists

- Users [2] This is the main place to ask questions and provide support to people using CFEclipse.
- Developers [3] This is for developers that are writing or want to write code for CFEclipse itself.
- Testers [4] This is a closed group for people that want to regularly test features of CFEclipse. Joining this group means you want to actively be a tester, so no support questions, but submissions of test results etc.
- SnipEx [5] This is a group for developers of SnipEx services. If you like SnipEx and want to talk about how to code it or integrate it, here is the place to go to.

External links

- Mark Drew's CF etc.. [6]
- Home page [7]
- What is CFEclipse ? [8]

References

[1] http://www.cfeclipse.org
[2] http://groups.google.com/group/cfeclipse-users
[3] http://groups.google.com/group/cfeclipse-dev
[4] http://groups.google.com/group/cfeclipse-testers
[5] http://groups.google.com/group/cfeclipse-snipex
[6] http://www.markdrew.co.uk
[7] http://cfeclipse.org/
[8] http://www.adobe.com/devnet/coldfusion/articles/cfeclipse.html

CFUnit

CFUnit is a unit testing framework for ColdFusion (CFML), modelled after the popular JUnit framework. CFUnit is an Open Source project hosted on SourceForge. "CFUnit v0.1 Alpha" was the project's first publicly distributed version, published in August of 2005. The latest version, v2.0 Beta 1, was released on 16 September 2006.

Several articles had been written in the House of Fusion [1], where there is also a mailing list dedicated to CFUnit [2].

See also

cfcUnit - an alternate unit testing framework for CFML.

External links

- CFUnit home page [3]
- Sourceforge site [4]
- JUnit home page [5]

References

[1] http://www.houseoffusion.com/
[2] http://houseoffusion.com/groups/cfunit/
[3] http://cfunit.sourceforge.net
[4] http://sourceforge.net/projects/cfunit
[5] http://junit.org/

cfcUnit

cfcUnit is a unit testing framework for ColdFusion (CFML), part of the xUnit family, and modelled on the API of the JUnit framework.

See also

CFUnit - an alternate unit testing framework for CFML.

External links

- cfcUnit home page [1]
- JUnit home page [5]

References

[1] http://www.cfcunit.org/cfcunit/

ColdFusion on Wheels

Developer(s)	Wheels Core Team [1]
Initial release	November 27, 2009
Stable release	1.0.4 / April 21, 2010
Written in	CFML
Operating system	Cross-platform
Development status	Active
Type	Web application framework
License	Apache License
Website	cfwheels.org [2]

ColdFusion on Wheels is an open source web application framework designed for applications written in ColdFusion Markup Language. Its name is often shortened to **CFWheels** or **Wheels**.

Wheels was designed to bring many concepts from Ruby on Rails to ColdFusion. Its developers aim for it to be simple to use, allow for rapid development, and make use of the Model-view-controller architectural pattern.

External links

- Official site [2]
- "Putting Wheels on ColdFusion" [3] by David Utter, *WebProNews*

References

[1] http://cfwheels.org/community/core-team
[2] http://cfwheels.org/
[3] http://www.webpronews.com/expertarticles/expertarticles/wpn-62-20060802PuttingWheelsOnColdFusion.html

ColdFusion

Original author(s)	Jeremy and JJ Allaire
Developer(s)	Adobe Systems Incorporated
Initial release	1995
Stable release	Adobe ColdFusion 9
Operating system	Windows, Linux, UNIX, Macintosh
Available in	English
Type	Application server
License	Proprietary
Website	ColdFusion Homepage [2]

ColdFusion is a commercial rapid application development platform invented by Jeremy and JJ Allaire in 1995. Originally designed to make it easier to connect simple HTML pages to a database, by version 2 it had become a full platform that included an IDE in addition to a full Scripting Language. Current versions of ColdFusion, sold by Adobe Systems, include advanced features for enterprise integration and development of rich internet applications. ColdFusion primarily competes with PHP and ASP.

Overview

One of the distinguishing features of ColdFusion is its associated scripting language, ColdFusion Markup Language (CFML), which compares to the scripting components of ASP, JSP, and PHP in purpose and features, but more closely resembles HTML in syntax. "ColdFusion" is often used synonymously with "CFML", but there are additional CFML application servers besides ColdFusion, and ColdFusion supports programming languages other than CFML, such as server-side Actionscript and embedded scripts that can be written in a JavaScript-like language known as CFScript.

Originally a product of Allaire and released in July 1995, ColdFusion was developed by brothers Joseph JJ and Jeremy Allaire. In 2001 Allaire was acquired by Macromedia, who in turn were acquired by Adobe Systems Inc in 2005.

ColdFusion is most often used for data-driven web sites or intranets, but can also be used to generate remote services such as SOAP web services or Flash remoting. It is especially well-suited as the server-side technology to the client-side Flex.

ColdFusion can also handle asynchronous events such as SMS and instant messaging via its gateway interface, available in ColdFusion MX 7 Enterprise Edition.

Main features

ColdFusion provides a number of additional features out of the box. Among them:

- Simplified database access
- Client and server cache management
- Client-side code generation, especially for form widgets and validation
- Conversion from HTML to PDF and FlashPaper
- Data retrieval from common enterprise systems such as Active Directory, LDAP, SMTP, POP, HTTP, FTP, Microsoft Exchange Server and common data formats such as RSS and Atom
- File indexing and searching service based on Verity K2
- GUI administration
- Server, application, client, session, and request scopes
- XML parsing, querying (XPath), validation and transformation (XSLT)
- Server clustering
- Task scheduling
- Graphing and reporting
- Simplified file manipulation including raster graphics (and CAPTCHA) and zip archives (introduction of video manipulation is planned in a future release)
- Simplified web service implementation (with automated WSDL generation / transparent SOAP handling for both creating and consuming services - as an example, ASP.NET[1] has no native equivalent for <CFINVOKE WEBSERVICE=" UNIQ-nowiki-0-9adabca3ea8f40f0-QINU " METHOD="Celsius2Fahrenheit" TEMP="#tempc#" RETURNVARIABLE="tempf">[2])

Other implementations of CFML offer similar or enhanced functionality, such as running in a .NET environment or image manipulation.

The engine was written in C and featured, among other things, a built-in scripting language (CFScript), plugin modules written in Java, and a syntax very similar to HTML. The equivalent to an HTML element, a ColdFusion tag begins with the letters "CF" followed by a name that is inidicative of what the tag is interpreted to, in HTML. E.g. <cfoutput> to begin the output of variables or other content.

In addition to CFScript and plugins (as described), CFStudio provided a design platform with a WYSIWYG display. In addition to ColdFusion, CFSTudio also supports syntax in other languages popular for backend programming, such as Perl. In addition to making backend functionality easily available to the non-programmer, (version 4.0 and forward in particular) integrated easily with the Apache Web Server and with Internet Information Server.

Other features

The first version of ColdFusion (then called Cold Fusion) was released on July 10, 1995. This first version was written almost entirely by one person, Joseph JJ Allaire. Primitive by modern standards, early versions of ColdFusion did little more than database access.[3]

All versions of ColdFusion prior to 6.0 were written using Microsoft Visual C++. This meant that ColdFusion was largely limited to running on Microsoft Windows, although Allaire did successfully port ColdFusion to Sun Solaris starting with version 3.1.

For reasons that may have been tied to lackluster sales the company was sold to Macromedia, then to Adobe. Earlier versions were not as robust as the versions available from version 4.0 forward.

With the release of ColdFusion MX 6.0, the engine had been re-written in Java and supported its own runtime environment, which was easily replaced through its configuration options with the runtime environment from Sun. Version 6.1 included the ability to code and debug Shockwave Flash.

History

- **1995** : Allaire Cold Fusion version 1.0
- **1996** : Allaire Cold Fusion version 1.5
- **1996** : Allaire Cold Fusion version 2.0
- **1997-June** : Allaire Cold Fusion version 3.0
- **1998-January** : Allaire Cold Fusion version 3.1
- **1998-November** : Allaire ColdFusion version 4.0 (space eliminated between Cold and Fusion to make it ColdFusion)
- **1999-November** : Allaire ColdFusion version 4.5
- **2001-June** : Macromedia ColdFusion version 5.0
- **2002-May** : Macromedia ColdFusion MX version 6.0 (build 6,0,0,48097), Updater 1 (build 6,0,0,52311), Updater 2 (build 6,0,0,55693), Updater 3 (build 6,0,0,58500)
- **2003-July** : Macromedia ColdFusion MX version 6.1 (build 6,1,0,63958), Updater 1 (build 6,1,0,83762)
- **2005** : Macromedia ColdFusion MX 7 (build 7,0,0,91690), 7.0.1 (build 7,0,1,116466), 7.0.2 (build 7,0,2,142559)
- **2007-July-30** : Adobe ColdFusion 8 (build 8,0,0,176276)
- **2009-April-04** : Adobe ColdFusion 8.0.1 (build 8,0,1,195765)
- **2009-October-05** : Adobe ColdFusion 9 (build 9,0,0,251028)

Versions

Cold Fusion 3.1

Version 3.1 brought about a port to the Sun Solaris operating system. Cold Fusion studio gained a live page preview and HTML syntax checker.

ColdFusion 4

"Cold Fusion" moniker renamed simply as "ColdFusion" - possibly to distinguish it from Cold fusion theory.

ColdFusion 4.5

Version 4.5 brought the ability to natively invoke Java objects, execute system commands, and talk directly to a Java EE server.

ColdFusion 5

First release from Macromedia after Allaire acquisition. The last to be legacy coded for a specific platform.

ColdFusion 6 aka MX

Prior to 2000, Allaire began a project codenamed "Neo". This project was later revealed as a ColdFusion Server re-written completely using Java. This made portability easier and provided a layer of security on the server, because it ran inside a Java Runtime Environment. Senior software engineer Damon Cooper, still with Adobe on the LiveCycle team, was the major initiator of the Java move.

On January 16, 2001, Allaire announced a pending merger with Macromedia. Macromedia continued its development and released the product under the name ColdFusion 5.0. It retained the name "ColdFusion" through the remainder of version 5 releases. In June 2002 Macromedia released the product under a slightly different name, allowing the product to be associated with the Macromedia brand, as well as the brand that the Allaire brothers had given it, originally: ColdFusion MX (6.0). ColdFusion MX was completely rebuilt from the ground up and was based on the Java EE platform. ColdFusion MX was also designed to integrate well with Macromedia Flash using

Flash Remoting.

With the release of ColdFusion MX, the CFML language API was released with an OOP interface.

ColdFusion MX 7

With the release of ColdFusion 7.0 on February 7, 2005, the naming convention was amended, rendering the product name "Macromedia ColdFusion MX 7". CFMX 7 added Flash-based, and XForms-based, web forms and a report builder that output in Adobe PDF as well as FlashPaper, RTF and Excel. The Adobe PDF output is also available as a wrapper to any HTML page, converting that page to a quality printable document. The enterprise edition also added Gateways. These provide interaction with non-HTTP request services such as IM Services, SMS, Directory Watchers, and an asynchronous execution. XML support was boosted in this version to include native schema checking.

ColdFusion MX 7.0.2, codenamed "Mystic" includes advanced features for working with Adobe Flex 2.

Adobe ColdFusion 8

On July 30, 2007, Adobe Systems released **ColdFusion 8**, dropping "MX" from its name. During beta testing the codename used was "Scorpio" (the eighth sign of the zodiac and the eighth iteration of ColdFusion as a commercial product). More than 14,000 developers worldwide were active in the beta process - many more testers than the 5,000 Adobe Systems originally expected. The ColdFusion development team consisted of developers based in Newton/Boston, Massachusetts and offshore in Bangalore, India.

Some of the new features are the CFPDFFORM tag, which enables integration with Adobe Acrobat forms, some image manipulation functions, Microsoft .NET integration, and the CFPRESENTATION tag, which allows the creation of dynamic presentations using Adobe Acrobat Connect, the Web-based collaboration solution formerly known as Macromedia Breeze. In addition, the ColdFusion Administrator for the Enterprise version ships with built-in server monitoring. ColdFusion 8 is available on several operating systems including Linux, Mac OS X and Windows Server 2003.

Other additions to ColdFusion 8 are built-in AJAX widgets, file archive manipulation (CFZIP), Microsoft Exchange server integration (CFEXCHANGE), image manipulation including automatic captcha generation (CFIMAGE), multi-threading, per-application settings, Atom and RSS feeds, reporting enhancements, stronger encryption libraries, array and structure improvements, improved database interaction, extensive performance improvements, PDF manipulation and merging capabilities (CFPDF), interactive debugging, embedded database support with Apache Derby, and a more ECMAScript compliant CFSCRIPT.

For development of ColdFusion applications, several tools are available: primarily Adobe Dreamweaver CS4, Macromedia HomeSite 5.x, CFEclipse, Eclipse and others. "Tag updaters" are available for these applications to update their support for the new ColdFusion 8 features.

Adobe ColdFusion 9

ColdFusion 9 (Codenamed: Centaur) was released on October 5, 2009. New features for CF9 include:

- Ability to code User Defined Functions (UDFs) and ColdFusion Components (CFCs) entirely in CFScript.
- An explicit "local" scope that does not require local variables to be declared at the top of the function.
- Implicit getters/setters for CFC.
- Implicit constructors via method called "init" or method with same name as CFC.
- New CFFinally tag for Exception handling syntax and CFContinue tag for Control flow.
- Object-relational mapping (ORM) Database integration through Hibernate (Java).
- Server.cfc file with onServerStart and onServerEnd methods.
- Tighter integration with Adobe Flex and Adobe AIR.

- Integration with key Microsoft products including Word, Excel, Sharepoint, Exchange and Powerpoint.
- In Memory Management - or Virtual File System - an ability to treat content in memory as opposed to using the HDD.
- Exposed as Services - an ability to access, securely, functions of the server externally.

Adobe ColdFusion Builder

Adobe ColdFusion Builder (codenamed "Bolt") is the name for Adobe's new Eclipse based development IDE that can be used to build applications for ColdFusion. The codename Bolt is a reference to the original lightning icon for the product from the Allaire days. ColdFusion Builder became available on 22nd March, 2010 along with Flash Builder 4. [4]

Features include:

- Object Relational Mapping auto-configuration
- Application Code Generation
- Server management
- Easily extensible through the Eclipse framework
- CFML, HTML, Javascript, and CSS Syntax Highlighting
- Code assist for tags, functions, variables, and components
- Code folding
- Snippet creation and management
- Outline viewing
- RDS Explorer for files and databases
- Line-level Debugging
- Refactoring

Features

Rich forms

ColdFusion Server includes a subset of its Macromedia Flex 1.5 technology. Its stated purpose is to allow for rich forms in HTML pages using CFML to generate Flash movies. These Flash forms can be used to implement rich internet applications, but with limited efficiency due to the ActionScript restrictions in place on Flash forms by Macromedia.

Flash forms also provide additional widgets for data input, such as date pickers and data grids.

In previous versions of ColdFusion, some form validation and additional widgets were available using a combination of Java applets and JavaScript. This option persists for those who do not wish to use Flash, however not all features are supported.

An example:

```
<cfform format="flash" method="post" width="400" height="400">
      <cfinput type="text" name="username" label="Username"
required="yes" >
      <cfinput type="password" name="password" label="Password"
required="yes" >
      <cfinput type="submit" name="submit" value="Sign In" >
</cfform>
```

ColdFusion also includes some XForms capability, and the ability to "skin" forms using XSLT.

PDF and FlashPaper generation

ColdFusion can generate PDF or FlashPaper documents using standard HTML (i.e. no additional coding is needed to generate documents for print). CFML authors simply place HTML and CSS within a pair of cfdocument tags and specify the desired format (FlashPaper or PDF). The generated document can then either be saved to disk or sent to the client's browser. ColdFusion 8 has now introduced the cfpdf tag which allows for unprecedented control over PDF documents including PDF forms, and merging of PDFs. These tags however do not use Adobe's PDF engine but a free and open source java library called iText.

ColdFusion Components (Objects)

ColdFusion was originally not an object-oriented programming language, and even today lacks some OO features. ColdFusion falls into the category of OO languages that do not support multiple inheritance (along with Java, Smalltalk etc.)[5] . With the MX release (6+), ColdFusion introduced the *component* language construct which resembles classes in OO languages. Each *component* may contain any number of properties and methods. One component may also extend another (Inheritance). Components only support single inheritance. With the release of ColdFusion 8, Java-style interfaces are supported. ColdFusion components use the file extension *cfc* to differentiate them from ColdFusion templates (.cfm).

Remoting

Component methods may be made available as web services with no additional coding and configuration. All that is required is for a method's access to be declared 'remote'. ColdFusion automatically generates a WSDL at the URL for the component in this manner: http://path/to/components/Component.cfc?wsdl. Aside from SOAP, the services are offered in Flash Remoting binary format.

Methods which are declared remote may also be invoked via an HTTP GET or POST request. Consider the GET request as shown.

```
http://path/to/components/Component.cfc?method=search&query=your+query&mode=strict
```

This will invoke the component's search function, passing "your query" and "strict" as arguments.

This type of invocation is well-suited for AJAX-enabled applications. ColdFusion 8 introduced the ability to serialize ColdFusion data structures to JSON for consumption on the client.

The ColdFusion server will automatically generate documentation for a component if you navigate to its URL and insert the appropriate code within the component's declarations. This is an application of component introspection, available to developers of ColdFusion components. Access to a component's documentation requires a password. A developer can view the documentation for all components known to the ColdFusion server by navigating to the ColdFusion URL. This interface resembles the Javadoc HTML documentation for Java classes.

Custom tags

ColdFusion provides several ways to implement custom markup language tags, i.e. those not included in the core ColdFusion language. These are especially useful for providing a familiar interface for web designers and content authors familiar with HMTL but not imperative programming.

The traditional and most common way is using CFML. A standard CFML page can be interpreted as a tag, with the tag name corresponding to the file name prefixed with "cf_". For example, the file IMAP.cfm can be used as the tag "cf_imap". Attributes used within the tag are available in the ATTRIBUTES scope of the tag implementation page. CFML pages are accessible in the same directory as the calling page, via a special directory in the ColdFusion web application, or via a CFIMPORT tag in the calling page. The latter method does not necessarily require the "cf_" prefix for the tag name.

A second way is the development of CFX tags using Java or C++. CFX tags are prefixed with "cfx_", for example "cfx_imap". Tags are added to the ColdFusion runtime environment using the ColdFusion administrator, where JAR or DLL files are registered as custom tags.

Finally, ColdFusion supports JSP tag libraries from the JSP 2.0 language specification. JSP tags are included in CFML pages using the CFIMPORT tag.

Alternative server environments

ColdFusion originated as proprietary technology based on Web technology industry standards. However, it is becoming a less closed technology through the availability of competing products. Products include Railo, BlueDragon, IgniteFusion, SmithProject and Coral Web Builder.

The argument can be made that ColdFusion is even less platform-bound than raw Java EE or .NET, simply because ColdFusion will run on top of a .NET app server (New Atlanta), or on top of any servlet container or Java EE application server (JRun, WebSphere, JBoss, Geronimo, Tomcat, Resin Server, Jetty (web server), etc.). In theory, a ColdFusion application could be moved unchanged from a Java EE application server to a .NET application server.

Currently, alternative server platforms generally support ColdFusion MX 6.1 functionality, with minor changes or feature enhancements.

Interactions with other programming languages

ColdFusion and Java

The standard ColdFusion installation allows the deployment of ColdFusion as a WAR file or EAR file for deployment to standalone application servers, such as Macromedia JRun, and IBM WebSphere. ColdFusion can also be deployed to servlet containers such as Apache Tomcat and Mortbay Jetty, but because these platforms do not officially support ColdFusion, they leave many of its features inaccessible.

Because ColdFusion is a Java EE application, ColdFusion code can be mixed with Java classes to create a variety of applications and use existing Java libraries. ColdFusion has access to all underlying Java classes, supports JSP custom tag libraries, and can access JSP functions after retrieving the JSP page context (*GetPageContext()*).

Prior to ColdFusion 7.0.1, ColdFusion components could only be used by Java or .NET by declaring them as web services. However, beginning in ColdFusion MX 7.0.1, ColdFusion components can now be used directly within Java classes using the CFCProxy class.[6]

Recently, there has been much interest in Java development using alternate languages such as Jython, Groovy and JRuby. ColdFusion was one of the first scripting platforms to allow this style of Java development.

ColdFusion and .NET

ColdFusion 8 natively supports .NET within the CFML syntax. ColdFusion developers can simply call any .NET assembly without needing to recompile or alter the assemblies in any way. Data types are automatically translated between ColdFusion and .NET (example: .NET DataTable → ColdFusion Query).

A unique feature for a Java EE vendor, ColdFusion 8 offers the ability to access .NET assemblies remotely through proxy (without the use of .NET Remoting). This allows ColdFusion users to leverage .NET without having to be installed on a Windows operating system.

The move to include .NET support in addition to the existing support for Java, CORBA and COM is a continuation of Adobe ColdFusion's agnostic approach to the technology stack. ColdFusion can not only bring together disparate technologies within the enterprise, but can make those technologies available to a number of clients beyond the web browser including, but not limited to, the Flash Player, Adobe Integrated Runtime (AIR), Mobile devices (SMS), Acrobat Reader (PDF) and IM gateways.

Acronyms

The acronym for the ColdFusion Markup Language is ***CFML***. When ColdFusion templates are saved to disk, they are traditionally given the extension .cfm or .cfml. The .cfc extension is used for ColdFusion Components. The original extension was DBM or DBML, which stood for Database Markup Language. When talking about ColdFusion, most users use the acronym CF and this is used for numerous ColdFusion resources such as user groups (CFUGs) and sites.

CFMX is the common abbreviation for ColdFusion versions 6 and 7 (aka ColdFusion MX).

Companies using ColdFusion

- Bank of America
- BMW USA
- The Wharton School of the University of Pennsylvania
- Smithsonian
- Citigroup
- JPMorgan Chase
- Wells Fargo
- Department of Homeland Security
- NSA
- Federal Reserve Bank
- U.S. Senate
- Blue Cross Blue Shield
- NIH
- Mayo Clinic
- Eli Lilly
- eBay
- ESRI
- McAfee
- Cisco
- Symantec
- Boeing
- Xerox
- Under Armour
- Moen
- Hasbro
- Community Transit (Washington State)
- Washington Metro Transit Authority (DC)
- AT&T
- Cingular Wireless
- Sprint
- Verizon
- U.S. Olympic Committee
- PGA of America
- New York Giants
- Chicago Bears

[7]

Notes and references

[1] http://digitalcolony.com/2007/08/consuming-web-service-in-aspnet.aspx
[2] http://www.adobe.com/devnet/coldfusion/articles/cf_aspnet08.html
[3] Web Hosting Resource and Directory since 1997 - Tophosts.com (http://www.tophosts.com/articles/?3016.html)
[4] Adobe Launches Flash Builder 4 (http://www.pcmag.com/article2/0,2817,2361629,00.asp)
[5] nictunney.com - Coldfusion MoFo (http://www.nictunney.com/index.cfm?mode=entry&entry=AE4A4A21-65B8-F252-775A757FC01D0C30)
[6] Using the CFC Proxy (http://www.forta.com/misc/cfcproxy.htm)
[7] "Who's using ColdFusion" Adobe.com (http://www.adobe.com/products/coldfusion/customers/)

- "Adobe Ships ColdFusion 8" (http://www.adobe.com/aboutadobe/pressroom/pressreleases/200707/073007ColdFusion.html). *Adobe Systems Incorporated*. 2007-07-30.

See also

- 4GL
- BlueDragon - Proprietary .NET-based CFML Engine and Free Open Source Java-based CFML Engine (Open BlueDragon)
- ColdFusion Markup Language
- Comparison of programming languages
- Railo - Free, Open Source CFML Engine
- SmithProject - Free, Open Source CFML Engine
- CFUnited - annual ColdFusion conference

External links

- Official ColdFusion site (http://www.adobe.com/products/coldfusion/)
- ColdFusion documentation (http://help.adobe.com/en_US/ColdFusion/9.0/Developing/index.html)
- Official Railo site (http://www.getrailo.com/) (open source)
- Official Open BlueDragon site (http://www.openbluedragon.org/)
- ColdFusion (http://www.dmoz.org/Computers/Programming/Internet/ColdFusion//) at the Open Directory Project
- The ColdFusion section of Rosetta Code (http://rosettacode.org/wiki/Category:ColdFusion)
- ColdFusion technical mailing list (http://www.houseoffusion.com/groups/cf-talk)
- EasyCFM.COM - Learn ColdFusion (http://www.easycfm.com/)
- ColdFusion Resource Center (http://livedocs.adobe.com/coldfusion/8/htmldocs/help.html?content=Part_1_Installing_1.html)
- cf.Objective() - The Only Enterprise ColdFusion Conference (http://www.cfobjective.com)

ColdSpring Framework

ColdSpring is a web application framework for the ColdFusion application programming language, based on the Java Spring Framework, it provides Dependency injection, inversion of control and aspect-oriented programming design pattern capabilities in an effort to make the configuration and dependencies of ColdFusion components (CFCs) easier to manage.

Integration

A noted[1] strength of ColdSpring is its ability to provide complimentary services to other applications and frameworks. ColdSpring has been deeply embedded within the core of the Model-Glue framework since Model-Glue 2.0. Also, Fusebox since 5.0 ships with a ColdSpring-specific lexicon.

In reverse, ColdSpring ships with connection points for Model-Glue, Mach-II and the unit testing framework CFCUnit.

History

ColdSpring has historically had a long development and release cycle when compared to other ColdFusion frameworks. ColdSpring was first mentioned by Dave Ross when he released a pre-alpha version on February 9, 2005[2] . Interest was found quickly within the ColdFusion community and a support group was formed around the software later in 2005[3] , as was the ColdSpring Framework web site. Eventually, a release candidate was released June 2, 2006[4] .

ColdSpring 1.0

June 25, 2006 ColdSpring 1.0 was finally released just three days before CFUnited[5] where Dave Ross was scheduled to speak on the topic.

ColdSpring 1.2

September 12, 2008 The 1.2 release[6] included changes to make working with beans, especially when using the XML Bean Factory, much easier, including creating bean aliases, including other bean configuration files, creating collections within the configuration file and other fixes[7] .

Future ColdSpring 2.0

The future of ColdSpring includes a full rewrite of the core libraries by Mark Mandel, and is codenamed Narwhal[8] .

External links

- ColdSpring Framework [9]
- Manage dependency injection for ColdFusion with the ColdSpring framework by Brian Kotek [10]
- Using the ColdSpring Dependency Injection Framework for ColdFusion [11]

References

[1] Better Coding with the Model-Glue:Unity ColdFusion Application Framework (http://articles.techrepublic.com.com/5100-10878_11-6120921.html?tag=rbxccnbtr1)
[2] http://www.d-ross.org/index.cfm?objectid=F7D09312-A7F9-DF09-3E8E59AC861E3651 Dave Ross, ColdSpring Pre-Alpha Release
[3] http://www.d-ross.org/index.cfm?objectid=D79C3C72-06D5-43E7-5BD79ACF04EACA5C Dave Ross, ColdSpring Shout Outs
[4] http://www.d-ross.org/index.cfm?objectid=95E4B4DE-C407-A9D9-88996A41797143CB Dave Ross, ColdSpring 1.0 RC1 Release
[5] http://www.mattwoodward.com/machblog/index.cfm?event=showEntry&entryID=0193362D-F722-89EC-82A8092554E467E6 Matt Woodward, ColdSpring 1.0 Released
[6] http://corfield.org/blog/index.cfm/do/blog.entry/entry/ColdSpring_12_Released Sean Corfield, Coldspring 1.2 Released
[7] http://www.briankotek.com/blog/index.cfm/2008/9/22/Whats-New-In-ColdSpring-12 Brian Kotek, What's New in ColdSpring 1.2
[8] http://www.compoundtheory.com/?action=displayPost&ID=463 Mark Mandel, CFObjective 2010 topics include ColdSpring 2.0
[9] http://www.coldspringframework.org
[10] http://articles.techrepublic.com.com/5100-10878_11-6132004.html
[11] http://www.theserverside.com/news/1363647/Using-The-ColdSpring-Dependency-Injection-Framework-for-ColdFusion

Fusebox (programming)

Fusebox is a web application framework for ColdFusion and PHP. Originally released in 1997, the current version, 5.5, was released in December 2007.

Fusebox is intended to be easy to learn and provides benefits by helping developers structure their code through a set of simple conventions. Fusebox also allows advanced developers to build large applications, leveraging design patterns and object-oriented programming techniques if they wish.

Overview

Fusebox provides web application developers with a standardized, structured way of developing their applications using a relatively straightforward and easy to learn set of core files and encouraged conventions. In addition to the framework itself, Fusebox has become closely associated with a Web application development methodology developed by its proponents known as "FLiP" (for Fusebox Lifecycle Process). (Many people refer to Fusebox as a "methodology", but in fact, as stated, it's a development framework. FLiP, however, is a methodology). Many frameworks provide comparable advantages; however, Fusebox (probably on account of both its relatively long history and the sizable and active community that supports it) seems to be the most popular one for ColdFusion. The framework has been ported and used in ASP, JSP, Perl/CGI and PHP as well, though the ColdFusion and PHP versions of Fusebox are the only versions to gain momentum.

It is important to note that Fusebox deals primarily with the effort of wiring together view states (pages) with controller actions (form submits, etc.) and the front-end of the business-logic tier. The framework does not address creating and maintaining business logic such as database interaction or service layers.

Concepts

Fusebox, Circuits and Fuseactions

The original concepts behind Fusebox were based on the household idiom of an electrical fusebox that controls a number of circuits, each one with its own fuse. In a Fusebox web application, all requests are routed through a single point (usually index.cfm for ColdFusion) and processed by the Fusebox core files. The application is divided into a number of circuits (usually in sub-directories) which are intended to contain related functionality. Each circuit in the application is further divided into small files called fuses that should perform simple tasks. As such, Fusebox is considered an implementation of the front controller, a common design pattern.

URLs within a Fusebox web application are usually of the form index.cfm?fuseaction=cname.fname where "cname" is the name of a circuit and "fname" is an XML-defined "method" within that circuit known as a fuseaction. The query-string variable name "fuseaction" can vary depending on configuration parameters, so not all applications using Fusebox need to use the action variable "fuseaction".

Naming Conventions

Fusebox encourages, but does not enforce, separation of presentation logic from business logic. It uses a number of file naming conventions to encourage this separation: presentation files begin with dsp (display) or lay (layout), database access files begin with qry (query) and general business files begin with act (action). Typical file names are in the format [prefix]_[filename] like dsp_loginform.cfm. Additional naming conventions are used by some Fusebox developers but these are the most common ones.

Exit Fuseactions

Another concept that Fusebox encourages is to parameterize any exit points in a web page, coding them as variables that are set in the circuit control file. These exit points are known as XFAs - eXit FuseActions. The idea is that by parameterizing the exit points in a web page, the flow of control can be updated more easily, allowing more reuse of web pages or fragments thereof.

FuseDocs

Associated with the framework, but not strictly part of it, is the concept of FuseDocs which is a semi-formalized form of documentation written in XML that specifies the inputs and outputs of each fuse file. There are third-party tools available which can use FuseDocs to do things like generate test harness code.

History

Fusebox has had several major revisions over the years. The most popular versions in use today are Fusebox 3, 4 (including 4.1) and 5. In Fusebox 3, the control files were all written in the underlying programming language (e.g., fbx_Switch.cfm for ColdFusion). Fusebox 4 and later versions use XML for the control files (fusebox.xml and circuit.xml), but other framework components are written using the underlying programming language (e.g. fusebox5.cfm, again for ColdFusion). In theory, this helps improve tool support for the framework. It also allowed for the pre-parsing and generation of a single template for processing each fuseaction, greatly increasing performance. Fusebox 5.5 allows the XML files to be omitted if certain conventions are followed.

Fusebox (version 1)

Fusebox 1 grew out of a conversation on the CF-Talk mailing list in April 1998. The participants included Michael Dinowitz, Josh Cyr, Steve Nelson and Gabe Roffman. Nelson and Roffman are credited with creating the original Fusebox though the first Fusebox program was written by Josh Cyr. The methodology was constantly evolving and beyond a whitepaper and a handful of examples, no official documentation existed. Very few developers were exposed to Fusebox during these early days.

Fusebox 2

Craig Girard and Steve Nelson (along with Hal Helms and Nat Papovich) wrote a book, Fusebox: Methodology and Techniques, which was published in 2000 by Fusion Authority. Programmers who followed the practices described in the book were said to be doing "Fusebox 2."

XFB

Hal Helms built upon Fusebox 2 and called his ideas eXtended FuseBox, or XFB.

Fusebox 3

Fusebox 3 (written primarily by John Quarto-von Tivadar and Nat Papovich) was an effort by leading members of the Fusebox community to incorporate XFB and other ideas into a reusable library, known as the "core files." A simple API allowed application code to communicate with the core files. Upon release in the fall of 2001, Fusebox became a framework rather than a methodology. A subsequent 3.01 release addressed minor issues. Fusebox 3 was something of a sea-change from Fusebox 2. Only the original principles remained relatively unchanged; a Fusebox 2 and Fusebox 3 application are structured very differently.

Fusebox 4

Fusebox 4 was a complete rewrite of Fusebox 3. The license [1] for the core files (which is open source) is held by a private company owned by John Quarto-von Tivadar: The Fusebox Corporation [2] (which appears to be a defunct corporation).

Fusebox 4.1 introduced some new XML grammar elements beyond those available in 4.0 that let you declare, instantiate and manipulate objects (COM, Java and ColdFusion Components) as well as web services. These features have provided Fusebox developers with the means of tying object-oriented models (i.e. business-logic) directly into their controllers. However, many Fusebox developers used object-oriented or highly-structured models in earlier versions of Fusebox or in the current versions without use of these grammar elements.

Fusebox 5

In 2006, The Fusebox Corporation asked Sean Corfield to take the lead in developing the next iteration of Fusebox. Fusebox 5 was another complete rewrite with new features and improved performance. Fusebox 5 nearly completely maintained backwards-compatibility with Fusebox 4.1. In November 2006 The Fusebox Corporation transferred ownership of the core files and fusebox website to TeraTech under the guidance of TeraTech president and Fusebox speaker Michael Smith. TeraTech announced that Fusebox will remain open source and is seeking to increase community involvement in the project again. Fusebox 5.1 and all subsequent releases are licensed under the Apache Source License 2.0 [3]. In February 2007 the members of Team Fusebox [4] met at the Frameworks conference in Bethesda Maryland and created a plan of action for community involvement using volunteers in nine different areas of Fusebox.

Fusebox 5.5

This release focused primarily on adding a set of conventions that allow the creation of Fusebox applications without XML configuration files. The use of these new features instead of XML is called "implicit Fusebox".

- Alpha testing began in June 2007
- A Public Beta became available at Adobe MAX in October 2007
- The official release of Fusebox 5.5 became available at the beginning of December 2007

Current Status

The release of Fusebox 5.5.1 in March 2008 was the last release by Sean Corfield. In August 2008, Adam Haskell took over development[5] , but became frustrated with the Fusebox organization[6] , and attempted to branch a new framework called FuseNG (NG for Next Generation, a Star Trek reference). FuseNG quickly lost steam and ended without a release [7] .

As it stands now, Fusebox has no future development plan, no developers, and no support if bugs are found. The TeraTech company owns Fusebox and has no plans release it to any other group. The death of Fusebox has been greatly debated [8] [9] [10] .

See also

- Comparison of web application frameworks

External links

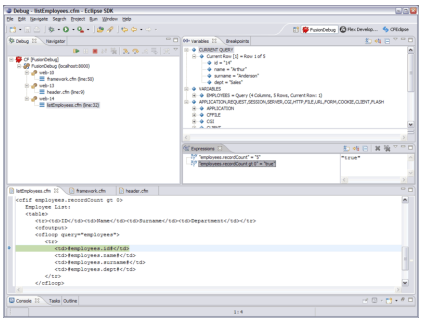
- Fusebox.org [11]
- Introduction to the Fusebox Framework [12] (adobe.com)
- Tap the power of the popular Fusebox 4 [13] (builder.com.com, November 2003)
- Fusebox 4 Review [14] (sys-con.com, September 2003)
- Fusebox 3 Feature [15] (sys-con.com, November 2001)
- Fusebox development project wiki [16]
- Fusebox 4 PHP wiki [17]
- Fusebox Framework Documentation Project [18]
- Fusebox mailing list (house of fusion) [19]
- Official Fusebox mailing list [20]
- Fusebox Light - A simplified variation for smaller projects [21]

References

[1] http://www.fusebox.org/index.cfm?fuseaction=fusebox.isFree
[2] https://esos.state.nv.us/SOSServices/AnonymousAccess/CorpSearch/CorpDetails.aspx?lx8nvq=6avzenGWEqvUpDJSQTEiOA%253d%253d
[3] http://www.apache.org/licenses/LICENSE-2.0.html
[4] http://trac.fuseboxframework.org/fusebox/wiki/TeamFusebox
[5] Adam Haskell, new lead developer (http://www.fusebox.org/go/news/new-fusebox-core-leader-adam-haskell-announced)
[6] Open Letter to Custodians of Fusebox (http://cfrant.blogspot.com/2009/08/open-letter-to-custodians-of-fusebox.html)
[7] http://cfrant.blogspot.com/2009/11/fuseng-update.html Final FuseNG Update]
[8] Sean Corfield on the Death of Fusebox (http://corfield.org/blog/index.cfm/do/blog.entry/entry/On_Hal_Helms_Ruby_on_Rails_and_the_Death_of_ColdFusion_and_Fusebox)
[9] Fusebox / FuseNG Status, discussion on HouseOfFusion (http://www.houseoffusion.com/groups/fusebox/thread.cfm/threadid:1334)
[10] FuseNG and therefore Fusebox by default are dead, by Peter Farrell (http://blog.maestropublishing.com/fuseng-and-therefore-fusebox-by-default-are-d)
[11] http://www.fusebox.org
[12] http://www.adobe.com/devnet/coldfusion/articles/fusebox_basics.html

[13] http://builder.com.com/5100-6371-5097705.html
[14] http://coldfusion.sys-con.com/read/42066.htm
[15] http://coldfusion.sys-con.com/read/41834.htm
[16] http://trac.fuseboxframework.org/fusebox/roadmap
[17] http://fbx4.salientdigital.com/
[18] http://fuseboxipedia.com/
[19] http://www.houseoffusion.com/groups/fusebox
[20] http://groups.yahoo.com/group/fusebox5/
[21] http://www.c2.com/cgi/wiki?FuseBoxLite

FusionDebug



Developer(s)	Intergral GmbH
Initial release	2005
Stable release	FusionDebug 3.0.1 / November 27, 2009
Operating system	Windows, Linux, MAC OSX, Solaris
Available in	English
License	Proprietary
Website	FusionDebug Homepage [1]

FusionDebug an interactive step debugger for Adobe ColdFusion and Railo CFML Engine. Step through code line-by-line, step into, step over or out of code to better understand how CFML code is running. FusionDebug can be used as an alternative to using CFOUTPUT/CFDUMP statements.

Features included in most recent version

Support for CF frameworks and Adobe Flex Builder
Comes with a full installer which includes a complete IDE containing CFEclipse
Step through code line-by-line (where needed)
Drill into variables and scopes
Run to line functionality
Associate custom file extensions with FusionDebug
Set breaks on runtime custom exceptions
View stack traces

Releases

2009 : FusionDebug version 3.0.1
[2]

2009 : FusionDebug version 3.0
2007 : FusionDebug version 2.0.1 [3]
2007 : FusionDebug version 2.0
2005 : FusionDebug version 1.0

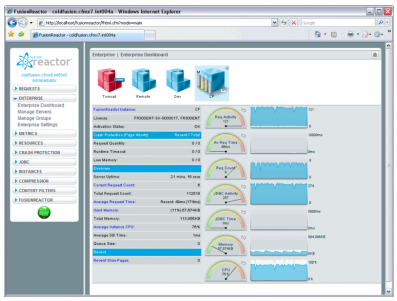
See also

- Adobe ColdFusion site [4]
- Railo CFML Engine [5]

References

[1] http://www.fusion-reactor.com/fd
[2] Intergral GmbH (2009-11-27). "FusionDebug 3.0.1 Release Notes" (http://www.fusion-reactor.com/support/kb/FDS-119.cfm). .
[3] Intergral GmbH (2007-05-25). "FusionDebug 2.0.1 Release Notes and Resolved Issues" (http://www.fusion-reactor.com/support/kb/FDS-96.cfm). .
[4] http://www.adobe.com/products/coldfusion/
[5] http://www.getrailo.com/

FusionReactor



FusionReactor running in Internet Explorer

Developer(s)	Intergral GmbH
Initial release	2005
Stable release	3.5.1 / October 22, 2009
Operating system	Windows, Linux, UNIX, MAC OSX
Available in	English
Type	Server Monitor
License	Proprietary
Website	FusionReactor Homepage [1]

FusionReactor is a commercial, server monitor developed by Intergral GmbH.

FusionReactor is a generic Java application server monitoring tool. Even before version 3 was released in January 2008, adding many new features, it was used in thousands of deployments by some of the world's largest organizations and by ColdFusion hosting companies, some of which have leveraged the tool since its initial release in November 2005.[2]

FusionReactor is designed for production server monitoring and uses less than 1% overhead.[3] FusionReactor supports Adobe ColdFusion, Railo, BlazeDS, LiveCycle and Flex Data Services, LiveCycle Enterprise Suite, and Acrobat Connect.[4]

Overview

How it works

FusionReactor works as a Servlet Filter - a light weight wrapper - around the CFML engine. The filter allows users to see information about requests from application servers as well as from databases.

Features

- Gather metrics on what is happening inside your servers
- Notifies you when server status changes
- Logs metric data for future analytics
- View stack traces to set you see what is happening to your server at that moment in time
- Has crash protection features with self-healing rules

Releases

- **2009** : FusionReactor version 3.5.1[5]
- **2008** : FusionReactor version 3.0.1
- **2008** : FusionReactor version 3.0
- **2007** : FusionReactor version 2.0.4
- **2006** : FusionReactor version 2.0
- **2005** : FusionReactor version 1.0

See also

- SeeFusion [6]
- Adobe ColdFusion site [4]
- Railo CFML Engine [5]

References

[1] http://www.fusion-reactor.com/fr
[2] Charlie Arehart (2008-05-17). "Coldfusion Server Healthcare" (http://www.fusionauthority.com/quarterly/do-more-code-less/fusionreactor-coldfusion-server-healthcare.pdf). .
[3] Intergral GmbH (2009-07-13). "Lightweight production server monitoring" (http://www.fusion-reactor.com/fr/). .
[4] Intergral GmbH (2009-07-13). "System Requirements" (http://www.fusion-reactor.com/fr/requirements.cfm). .
[5] Intergral GmbH (2009-10-22). "FusionReactor 3.5.1 Release Notes and Resolved Issues" (http://www.fusion-reactor.com/support/kb/FRS-230.cfm). .
[6] http://www.seefusion.com/

IgniteFusion

IgniteFusion is a freeware CFML script engine that runs cfm script files. Similar to Perl or PHP script engines the IgniteFusion script engine runs as an executable on the server. Other CFML engines include Adobe ColdFusion, New Atlanta BlueDragon, Railo, and Coral Web Builder.

<Note> This software is no longer supported / available from the authors.

See also

ColdFusion

External links

- Official Website [1]

< NOTE > The official website is no longer active.

References

[1] http://www.ignitefusion.com/

Mach-II

Developer(s)	Team Mach-II
Initial release	1 August 2003[1]
Stable release	1.6.1 / March 30, 2009
Preview release	1.8.0 RC2 / December 27, 2009
Written in	CFML
Operating system	Cross-platform
Development status	Mature
Type	Event-driven web application framework
License	Apache 2.0 for 1.6 and lower, GPLv3 with Classpath exception for 1.8+
Website	http://www.mach-ii.com

Mach-II is a web-application framework focused on easing software development and maintenance was the first Object-Oriented framework for CFML. It is maintained by a group of dedicated open source programmers

References

[1] "Mach-II Release History" (http://greatbiztoolsllc-trac.cvsdude.com/mach-ii/wiki/FAQReleaseHistory). . Retrieved 2008-10-14.

- Mach-II About Framework Mission (http://www.mach-ii.com/index.cfm/go/about/)
- Mach-II Features (http://www.mach-ii.com/index.cfm/go/features/)

External links

Mach-II Resources

- Mach-II web site (http://www.mach-ii.com/)
- The Harmonious Programmer (http://blog.maestropublishing.com/) - The blog of Peter J. Farrell
- Mach-II.info Resource Site (http://www.mach-ii.info) - Mach-II Resource Site
- Sean Corfield's Mach-II Page (http://corfield.org/index.cfm?fuseaction=machii.main) - Sean Corfield's Mach-II Page
- Mach-II.de the german side (http://www.mach-ii.de) - Mach-II (the German side)

Mach-II Open Source Applications

- MachBlog (http://www.machblog.org/) - A full-featured Mach-II blogging application.

See also

- Comparison of web application frameworks

Model-Glue

Model-Glue is a OO web application framework based on the MVC design pattern. Its goal is to simplify development of OO ColdFusion applications. It is released under the Apache Software License 2.0 (ASL2.0)[1]

Model-Glue Is

- An Implicit Invocation framework which facilitates use of the Model View Controller design pattern in ColdFusion applications.
- A framework encouraging clear separation of Model, View, and Controller
- Akin to Mach-II, another implicit invocation MVC framework.
- Written by Joe Rinehart, with feedback provided by Doug Hughes of Alagad, Inc.

External links

- The Model-Glue web site [2].

See also

- Comparison of web application frameworks

References

[1] http://www.model-glue.com/blog/index.cfm/2007/1/11/ModelGlue-License-Change
[2] http://www.model-glue.com

onTap

"Features Without Fixtures"

Developer(s)	S. Isaac Dealey
Stable release	3.2b / September 24, 2008
Operating system	Cross-platform
Type	web application framework
Licence	OpenBSD
Website	on.tapogee.com [1]

The **onTap** framework is a free service-oriented and "full stack" web application framework for ColdFusion.

In addition to providing an MVC controller like most other ColdFusion frameworks, it also includes an array of APIs for rapid application development, including e-mail, HTML templating (and associated DHTML widgets such as Section 508 compliant tabsets), AJAX, application branding and customization, form management and i18n internationalization features.

License

The onTap framework is distributed using the OpenBSD license [2].

Several of the early versions of the framework (prior to version 2.0) were released under the Lesser GPL. The LGPL had been chosen specifically for the purpose of allowing commercial software to be written using the framework as a starting-point. The OpenBSD license was later adopted for its even less restrictive terms, allowing commercial projects based on the framework to encrypt their own proprietary source code (which was not allowed by the LGPL).

Philosophy

The onTap framework has several key goals [3]:

- Speed and improve rapid application development (RAD) by simplifying common or tedious web development tasks as well as providing convenient methods of accomplishing very complex tasks such as and/or keyword search filtering (this example is from the object-relational mapping (ORM) tool which has split into a separate project called DataFaucet ORM [4]). The use of syntactic sugar is a primary method of achieving this goal.
- Enable better integration and collaboration between separate applications provided by different authors via a service-oriented architecture (SOA). The long-term goal is a software ecosystem similar to add-ons for the Mozilla Firefox browser in which plugin applications can be one-click installed via the existing browser-based interface.
- Enable easier customization of Software as a service (SaaS) applications by separating client customizations into their own directory structures thereby reducing conflicts between potentially incompatible customization requests. This is being described as a virtual private application (VPA) [5] as an analogy to the web hosting term virtual private server (VPS).

These goals are similar to and overlap the intent of agile software development methodologies or the Agile Manifesto seeking a "lightweight" method of software development that can produce versatile working software very

quickly.

To meet the objective of simplifying and improving the RAD process, the framework's core principles include Convention over Configuration (CoC) and Don't Repeat Yourself (DRY). One example of CoC and DRY principles can be found in the framework's form features. The form tools allow programmers to omit most of the code required to create common CRUD forms by relying on the database as the single point of truth for information about the type of data managed by the form. The following are examples of a form as created using the CFML native cfform tag as compared to using the onTap framework's CoC / DRY concepts for CRUD forms.

Sample Code

The following code sample shows how many ColdFusion forms are written:

```
<cfparam name="attributes.eventid" default="" />
<cfif len(trim(attributes.eventid))>
  <cfquery name="getEvent" datasource="primary">
    select * from tblEvent
    where eventid = <cfqueryparam value="#attributes.eventid#"
cfsqltype="cf_sql_idstamp" />
  </cfquery>
  <cfparam name="attributes.eventname"
default="#getevent.eventname#" />
  <cfparam name="attributes.eventdate"
default="#getevent.eventdate#" />
  <cfparam name="attributes.ticketprice"
default="#getevent.ticketprice#" />
</cfif>

<cfparam name="attributes.eventname" default="" />
<cfparam name="attributes.eventdate" default="" />
<cfparam name="attributes.ticketprice" default="0" />
<cfform format="xml">
  <cfinput type="hidden" name="eventid" value="#attributes.eventid#"
 />
  <cfinput type="text" name="eventname" label="Event"
    required="yes" value="#attributes.eventname#" />
  <cfinput type="text" name="eventdate" label="Date"
    required="yes" value="#attributes.eventdate#" validate="date" />
  <cfinput type="text" name="ticketprice" label="Ticket Price"
    required="yes" value="#attributes.ticketprice#" validate="numeric"
/>
</cfform>
```

The following code sample shows how the same form could be written using the onTap framework's XHTML template engine in combination with the DataFaucet ORM [4] plugin to speed development. This code anticipates the intent of the code in the previous sample by using a conventional relationship between database columns and form input elements. These two code samples produce an approximately similar result with mostly semantic differences in operation. This supports the philosophy of CoC because the programmer only needs to specify the value of an input element (or its default) or the type of validation (date, numeric, required, etc.) in atypical cases in which the input doesn't mirror the structure of the database.

```
<cf_html>
<tap:form tap:dbtable="tblEvent" xmlns:tap="xml.tapogee.com">
  <input type="hidden" name="eventid" />
  <input type="text" name="eventname" label="Event" />
  <input type="text" name="eventdate" label="Date" />
  <input type="text" name="ticketprice" label="Ticket Price" tap:default="0" />
</tap:form>
<cf_html>
```

History

Isaac Dealey began working on a content management system (CMS) in late 1998 following his first ColdFusion job at MCI WorldCom. The CMS transitioned through several names eventually becoming known as Tapestry (not to be confused with the Tapestry framework for Java). Isaac later abandoned the CMS but not before releasing an open source API for ColdFusion development called the Tapestry Application Programming Interface (TAPI) not to be confused with the Telephony Application Programming Interface (TAPI). The design of this early version focused on use within an existing application and within several months Isaac decided that the system requirements to support this strategy were too limiting. This led to the first release of the onTap framework (a complex clip of "on Tapestry") as an alternative to TAPI in August 2003. In spite of the fact that the name onTap shares pronunciation with a colloquial description of draught beer (which is often said to be "on tap"), the name engenders less confusion than either the TAPI acronym or the original CMS' name Tapestry.

Website

Some time between August 2003 and August 2004, an official website for the framework launched at fusiontap.com. In March 2007, Nick Tong and Kola Oyedeji interviewed Isaac [6] for a podcast about the framework on the cfFrameworks website [7]. Shortly after the interview, Isaac canceled the website's dedicated hosting service for personal reasons and the domain was subsequently purchased by domain scalpers. This created confusion in the following months with some people thinking the framework project might have been abandoned.

In December 2007 Isaac submitted the onTap framework and several related projects to the open source development community RIAForge.org, an alternative to SourceForge specifically for projects based on Adobe software platforms.

A new official site is now at http://on.tapogee.com starting in August 2008.

External links

- CFConversations podcast episode 19 [8]
- SitePoint interview blog with Kay Smoljak [9] - this article was a featured highlight on the SitePoint home page on Aug 25th, 2008
- ColdFusion Developer's Journal Special "Frameworks" Focus Issue article [10]
- cfFrameworks.com interview podcast [6]
- Kola Oyedeji : 8 things you didn't know about the onTap framework [11]
- Discussion of ORM techniques used in onTap with Peter Bell [12]
- Ray Camden's blog notice about ports of Galleon forums to several frameworks including the onTap framework [13]

See also

- Comparison of web application frameworks

References

[1] http://on.tapogee.com
[2] http://www.openbsd.org/policy.html
[3] http://ontap.wikispaces.com/Project+Goals
[4] http://www.datafaucet.com
[5] http://ontap.wikispaces.com/Virtual+Private+Applications+%28VPA%29
[6] http://www.cfframeworks.com/blog/index.cfm/2007/3/8/Isaac-Dealey-talks-about-the-onTap-framework
[7] http://www.cfframeworks.com
[8] http://www.cfconversations.com/index.cfm/2008/10/19/CFConversations-19-Roundtable-6-Controller-based-Frameworks-Part-1
[9] http://www.sitepoint.com/blogs/2008/08/25/isaac-dealey-on-the-ontap-framework/
[10] http://coldfusion.sys-con.com/read/176194.htm
[11] http://coolskool.blog-city.com/ontap.htm
[12] http://www.pbell.com/index.cfm/2007/3/22/Rethinking-the-Data-Access-Layer
[13] http://www.coldfusionjedi.com/index.cfm/2008/5/22/BlogCFC-and-Galleon-updates

Railo

Railo is a compiler for translating and executing of CFML-based websites. The compiler translates the CFML code into Java classes which can be executed on a Java server. Railo also comes with a runtime engine, which contains all necessary libraries for the translated code. Railo automatically detects when to translate a CFM file or when to use the runtime engine. Railo compares best with JSP. JSP uses different syntax but the main functionality is almost the same. Because Railo implements most of the JSP interfaces, it is highly compatible with JSP.

Railo Flavours

Railo comes in three main product editions:

- **Railo Express** (aka Railix) is the Live version of Railo, which utilises Jetty to run on a host machine without requiring installation. Railix is ideal for quickly trying out Railo, or for development away from one's main development machine, but it is not recommended for production use.
- **Railo Server** is the main version of Railo which can be integrated into a standard web server, and is suitable for production use.
- **Railo WAR** is the Web Archive version, suitable for use on any standardized Java EE server.
- There is also **Railo Custom** allowing you to customise Railo to your specific needs.

As of version 3.1, Railo is open source and is hosted by the jboss.org project.

Prior to version 3.1, Railo was available in four different versions, depending on what you need it for:

- **Railo Developer** is the default and for development use only. It has the same features like the enterprise version. The only restriction is that it only allows access from 10 different IP addresses.
- **Railo Community** is a free version for low budget business applications. It has some minor restrictions in functionality (CFVIDEO, Amazon S3 resource) but no restrictions in use. It is the same product as Railo Professional used to be, but without costs.
- **Railo Enterprise** is the same as professional, but without a limit on the number of webroots allowed. It is priced at €1800. In addition, it contains the full server administrator for configuring all web security and default settings for each single web.

CFML Compatibility

The current release, Railo 3.1, is mostly compatible with Coldfusion 8.0.1, but has a small number of discrepancies. There are also several additions/extensions to CFML provided by Railo, including the ability to quickly define Arrays, Structs and Querys in a single function. Performance is what makes Railo so interesting. Even with debugging turned on, Railo seems to be the fastest CFML-engine available.

Incompatibilities

Railo does not support the following tags: cfapplet, cfgrid, cfreport, cftree, cfformitem, cfformgroup, cftextarea, cfexchange, cfpresent, cffeed, cfpod, cflayout, cfmenu, cfprint, cfreport*, cfslider, cfsprydataset, cftooltip, cfcalendar, cfpdfform, cfpdfformparam, cfpdfsubform, cfNTauthenticate

Railo does not support the following functions: isDDX, isPDFFile, precisionEvaluate, getSOAP, getGatewayHelper, sendGatewayMessage, getPrinterInfo, queryConvertForGrid, verifyClient, ajax*, dotNetToCFType

Railo does not have the ability to decrypt encrypted CFX tags.

Framework Compatibility

Any CFML framework compatible with Coldfusion 8.0.1 is likely to work on Railo. The following is a list of popular frameworks known to run on Railo:

- Fusebox (Versions 4.x and 5.x)
- Mach-II
- Model-Glue
- Coldbox x
- ColdFusion on Wheels
- FarCry CMS
- Sava CMS
- Transfer ORM
- Reactor ORM

External links

- Railo website [1]
- "Railo Talk" Official Discussion List [2]
- "Get Railo" [5]
- "Railo Wiki" [3]

References

[1] http://www.railo.ch/en/
[2] http://groups.google.com/group/railo
[3] http://wiki.getrailo.org/

SmithProject

SmithProject is an Open Source CFML script engine that runs cfm script files.

The Smith Project was initiated by youngculture AG with the need to migrate a large ColdFusion based web application to Java.

Other CFML engines include Adobe ColdFusion, New Atlanta BlueDragon, Railo, and Coral Web Builder.

See also

ColdFusion

External links

- Official Website [1]

References

[1] http://www.smithproject.org/index.cfm

Article Sources and Contributors

ColdFusion Markup Language *Source*: http://en.wikipedia.org/w/index.php?oldid=351511735 *Contributors*: Allen3, AndrewHowse, AnmaFinotera, Anonymous Dissident, BD2412, BP, Bearcat, BenForta, Blurr, Bmeloche, C'est moi, CRGreathouse, Captainsteve@gmail.com, Carehart, Cmelbye, Coffeeflower, Czarofrandomness, DoohanOK, Elonka, FatalError, Fiftyquid, Frecklefoot, FusionA*, Gaius Cornelius, Graham87, Harej, HorsePunchKid, Imjustmatthew, JLaTondre, Jamelan, Jeff3000, Kunchaparthi, Lightmouse, Masondixon, Melaen, Michael614, Nklatt, Psiphiorg, Rgruchalski, Roberta F., Rror, Taeshadow, Tezeti, Toussaint, Twas Now, Wikitonic, Zoramite, 57 anonymous edits

BlogCFC *Source*: http://en.wikipedia.org/w/index.php?oldid=332654373 *Contributors*: BP, Rich Farmbrough, 1 anonymous edits

BlueDragon *Source*: http://en.wikipedia.org/w/index.php?oldid=334614688 *Contributors*: BP, Bbx, Blorg, Carehart, Coffeeflower, EagleOne, Encephalon, Firsfron, Gaius Cornelius, Grevian, HDCase, JLaTondre, Ketiltrout, Leandrod, Marudubshinki, Michael614, MuZemike, PC78, Plasticup, ReyBrujo, RxS, Sietse Snel, Thumperward, Werdna, 33 anonymous edits

CFEclipse *Source*: http://en.wikipedia.org/w/index.php?oldid=236411641 *Contributors*: Alexhubner, BP, Dreftymac, Khalid hassani, Leolaursen, Markdrew, Oswax, SlaveToTheWage, That Guy, From That Show!, 4 anonymous edits

CFUnit *Source*: http://en.wikipedia.org/w/index.php?oldid=349300544 *Contributors*: Andreas Kaufmann, BP, BigrTex, Djmckee1, Retired username, TheParanoidOne, 2 anonymous edits

cfcUnit *Source*: http://en.wikipedia.org/w/index.php?oldid=349011221 *Contributors*: Andreas Kaufmann, BP

ColdFusion on Wheels *Source*: http://en.wikipedia.org/w/index.php?oldid=357398915 *Contributors*: BP, Chrispetersweb, Josephgrossberg, Kingturtle, Kotepho, Matěj Grabovský, Pegship, Robofish, 2 anonymous edits

ColdFusion *Source*: http://en.wikipedia.org/w/index.php?oldid=359055908 *Contributors*: -Barry-, 0x6D667061, 21655, 7, Aaronwinborn, Ace23, Adidas, Adrian Lynch, Adrocknaphobia, After Midnight, Ahoerstemeier, Alainr345, Alansohn, Alanyst, AlexAnglin, AlistairMcMillan, Amigan, Anamanfan, AnmaFinotera, Arlogan0, Avono, BP, BenForta, Bkkbrad, Blobglob, Blorg, Bluemoose, Bmeloche, Bmicomp, Brian Gunderson, Bryant.cutler, Bsayusd, Cajunstrike, Caltas, Can't sleep, clown will eat me, Carehart, Charles Matthews, Chris the speller, Chris83, Chtito, Closedmouth, Cmcfarland, Coffeeflower, ColdFusion96, Colonies Chris, Computerjoe, Crania, Cutterbl, Daltenty, Damaster98, DamienMcKenna, DanInSanJose, Danakil, Danlance, Danmackey, Dave2anju, Davelowe, Davidtatt99, Dbabbitt, Dcoetzee, DeLarge, DeadEyeArrow, Defenestrate, Defenseman Emeritus, Deineka, Delux, Devnulled, Diberri, DoohanOK, Dragunova, Dreftymac, Duncancumming, Dysprosia, EagleOne, Eagleal, Elonka, Epolk, Evolutionbook, Falcon9x5, Flash200, Folajimi, Frankn12345, Frecklefoot, FrenchIsAwesome, FunnyYetTasty, Fusion sales 123, Fuzzie, Galoubet, Ggman, GregorB, Greyskinnedboy, H2g2bob, HDCase, Hasenstr, HebrewHammerTime, Hetzer, Hippy deluxe, HorsePunchKid, Hourback, Hyperlink, Iamserious, Illegal Operation, Ipggi, Irishguy, Isaacdealey, Ishako, Ithizar, JLaTondre, Jamelan, Jasy jatere, Jaybee, Jemptymethod, Jhartzell, Jhartzell42, Jimfbleak, John Mason II, Junkstar, Jvstein, Kbdank71, Kenguest, Kernel.package, Khalid hassani, King of Israel, Kippbakr, Kiteeye, Kku, Kreca, Lanasa, Leandrod, LeaveSleaves, Lennier1, Liamdaly620, Liempt, Linuxbeak, LittleSmall, Louison, Lupin, M1chu, M1ss1ontomars2k4, Maestrofjp, MarkusHagenlocher, Mason@fusionlink.com, Matrix mike2001, Mattbrundage, McGeddon, Mcnattyp, Mdinowitz, Mehdiirfani, MementoVivere, Mhenke, Michael614, MilesAgain, Mizst, Mjhagen, Mlliw, Mormegil, Mproud, MrNate, MrOllie, Mrileyaz, NapoliRoma, NauarchLysander, NeoDeGenero, Nikanorov, Nishantman, NurAzije, Obeattie, Ohnoitsjamie, Omarcheeseboro, Oolong, Orderud, Pablo X, Pegua, Perfecto, Pharos, Pmsyyz, Poor Yorick, Proxy User, RadicalBender, Rcooper123, RedWolf, RegularBreaker, Rhobite, Rightfully in First Place, Rikbrown2k, Rjwilmsi, Robert K S, RobertL, Ronark, Roxpace, Ruud Koot, Rwblackburn, Sander Säde, Satheeshpadmanabhan, Saxifrage, Sbauer318, Seano1, Sfitchet, Shadowjams, Shipmaster, SimCity4, Simoncpu, Simonwright, Soumyasch, Sprewell, Steeev, Stephen B Streater, Stevietheman, SubSeven, Taeshadow, Taka, Tawker, Teacurran, Template namespace initialisation script, Terrybader, ThD2007, Thalter, Thenarrowmargin, Thumperward, Timwi, Tkgd2007, Tmhunt2, Toby Woodwark, Toussaint, UnclejackDC, Uris, Utcursch, Uzume, Veinor, Vorratt, Watson Ladd, Webdev, Wernher, Wickethewok, Wilgeno, Will Beback, Wrathchild, Wwip, Xcentaur, Xpclient, Xyb, Ynhockey, Zanuto, 624 anonymous edits

ColdSpring Framework *Source*: http://en.wikipedia.org/w/index.php?oldid=361775078 *Contributors*: BP, Ekerazha, Elonka, Ian Bailey, JLaTondre, Maestrofjp, MrNate, Rfc1394, Toby Woodwark, Utcursch, 2 anonymous edits

Fusebox (programming) *Source*: http://en.wikipedia.org/w/index.php?oldid=361775768 *Contributors*: AnmaFinotera, BP, Beefyt, Bletch, DanMS, Dreftymac, Fred Bradstadt, Ian Bailey, Intgr, Isaacdealey, JLaTondre, Kevin@objectiveinternet.com, Leeborkman, Lifefeed, MStraw, Majorclanger, Marnen, Mattbrundage, Mav, Mcnattyp, Mernen, Merovingian, Michael614, MrNate, Nachoman-au, Oli Filth, Pmcelhaney, Pnevares, Poccil, Rfc1394, Rjwilmsi, Scopey42, Seancorfield, SymlynX, Uris, Woohookitty, Yendor1958, 60 anonymous edits

FusionDebug *Source*: http://en.wikipedia.org/w/index.php?oldid=357666147 *Contributors*: Hasenstr, Rjwilmsi, Tassedethe, 2 anonymous edits

FusionReactor *Source*: http://en.wikipedia.org/w/index.php?oldid=354890682 *Contributors*: Beagel, Bender235, Fortdj33, Hairhorn, Hasenstr, Hysteria18, 5 anonymous edits

IgniteFusion *Source*: http://en.wikipedia.org/w/index.php?oldid=335794622 *Contributors*: BP, IgniteFusion, Jevansen, 8 anonymous edits

Mach-II *Source*: http://en.wikipedia.org/w/index.php?oldid=352141002 *Contributors*: Ekerazha, Moonriddengirl, 6 anonymous edits

Model-Glue *Source*: http://en.wikipedia.org/w/index.php?oldid=352141035 *Contributors*: Arturus, BP, Dominic, Ekerazha, Frap, Gioto, Ian Bailey, Kelly Martin, Rfc1394, Topbanana, 14 anonymous edits

onTap *Source*: http://en.wikipedia.org/w/index.php?oldid=352141150 *Contributors*: AnmaFinotera, BP, Barticus88, Beefyt, Den fjättrade ankan, Duncancumming, Ekerazha, Gioto, Isaacdealey, JCLately, JimD, Joel7687, LinguistAtLarge, MacTed, Mcnattyp, Mkamensek, Pegship, Reedy, Saxifrage, Scientus, Wilhelm meis, 19 anonymous edits

Railo *Source*: http://en.wikipedia.org/w/index.php?oldid=353942211 *Contributors*: AVRS, BP, Blorg, Bmeloche, CanisRufus, Chrispetersweb, Davidcl, Hasenstr, JLaTondre, Kjkolb, MacGyverMagic, Mahanga, Mboverload, Omarcheeseboro, Pnevares, Rich Farmbrough, Teacurran, 26 anonymous edits

SmithProject *Source*: http://en.wikipedia.org/w/index.php?oldid=335794412 *Contributors*: Abmyers, Ekjon Lok, Jemptymethod, Jevansen, Moopet, Onorem, 4 anonymous edits

Image Sources, Licenses and Contributors

File:ColdFusion-on-Wheels.jpg *Source*: http://en.wikipedia.org/w/index.php?title=File:ColdFusion-on-Wheels.jpg *License*: Public Domain *Contributors*: User:Chrispetersweb

Image:ColdFusion icon.png *Source*: http://en.wikipedia.org/w/index.php?title=File:ColdFusion_icon.png *License*: unknown *Contributors*: Tkgd2007

Image:FusionDebugLogo.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:FusionDebugLogo.gif *License*: unknown *Contributors*: Hasenstr, Polly

Image:FusionDebugScreenshot.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:FusionDebugScreenshot.gif *License*: unknown *Contributors*: Hasenstr

Image:FusionReactorLogo.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:FusionReactorLogo.gif *License*: unknown *Contributors*: Hasenstr

Image:FRScreenshot.gif *Source*: http://en.wikipedia.org/w/index.php?title=File:FRScreenshot.gif *License*: unknown *Contributors*: Hasenstr

Image:MachII fullColor logo web.png *Source*: http://en.wikipedia.org/w/index.php?title=File:MachII_fullColor_logo_web.png *License*: unknown *Contributors*: GreatBizTools, LLC

Image:powerontap.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Powerontap.png *License*: unknown *Contributors*: Original uploader was Isaacdealey at en.wikipedia

License