



An Article on ColdFusion
[By: CF Team METAOPTION]

ColdFusion

What Is ColdFusion?

ColdFusion is a programming language based on standard HTML (Hyper Text Markup Language) that is used to write dynamic web pages. It lets you create pages on the fly that differ depending on user input, database lookups, time of day or whatever other criteria you dream up!

Why ColdFusion is used?

- **Rapid Application Development:**
ColdFusion allows web developers to develop websites with less code. Thus reducing time to develop and support web applications.
- **Power:**
ColdFusion recent developments have made it a more powerful platform as it is based on a Java Enterprise Level Application Server. This improves performance and allows for almost infinite functionality.
- **Scalability:**
ColdFusion is now based on the J2EE platform therefore the web server and web applications can be scaled easily to accommodate server clustering (allows multiple servers to work in harmony to perform the same task). This means that even if you were to see a 100% or 200% increase in website traffic the server can be improved to add more horse power.
- **Features:**
In the latest release of ColdFusion there have been huge feature additions that would add a new level of sophistication to your website. Things like - on the fly PDF document creation would allow website content to be presented in multiple formats without having to constantly maintain multiple sources of content.
- **Defending the reputation of ColdFusion:**
ColdFusion is often viewed by PHP developers and ASP developers as inadequate and insufficient. Although this was the case with the 4.5 and 5 versions of ColdFusion, it has changed significantly since the release of versions 6, 6.1, 7 and now ColdFusion MX 8. When comparing ColdFusion to PHP or ASP it is important to compare not only performance but also development costs. After a site is all complete the same website may cost around \$3000 in ColdFusion but with ASP we

have seen it cost closer to \$15,000.

In Short we can say ColdFusion is having Two Quantifiable factors (performance and Cost). ColdFusion Develop the Applications whose performance level is high and Cost of client is Low.

Built-in functionality:

ColdFusion Is Tag Based Language. A few of the more powerful ColdFusion tags (some of them new to ColdFusion MX) are:

- **CFCOMPONENT:** Creates and defines a component object; encloses functionality that you build in CMFL within cffunction tags and makes it available to any ColdFusion page on your ColdFusion Server or other remote services, such as web services, Macromedia Flash MX, and so forth. This is a new feature in ColdFusion MX. Read more about this in the ColdFusion MX Application Developer Center
- **CFCHART:** Creates complex charts and graphs in Macromedia Flash, JPEG, or PNG formats on the fly from your database queries (or other types of queries as well). This is a new feature in ColdFusion MX. Read more about this in Tim Bunter's Basic Charting and Graphing with ColdFusion MX.
- **CFXML:** Creates an XML document. Additionally, the new ColdFusion XML functions enable you to parse and search XML easily. This is a new feature in ColdFusion MX. Read more about this in Nate Weiss's Utilizing XML and XSLT in ColdFusion MX PDF format (528k).
- **CFDUMP:** Dumps a complete variable, structure, or query to the page in order to debug your CF pages. This tag is especially handy for dumping a complete result set to the page in a nicely formatted table. This is very helpful when testing and debugging.
- **CFDIRECTORY:** Provides you with complete access to any directory on your hard drive for listing, creating, deleting, or renaming directories.
- **CFFILE:** Provides you with complete access to the files on your server, including the ability to upload, move, rename, copy, delete, read, read binary, write, or append, all from one simple tag.

- **CFREGISTRY:** Reads and writes to the Windows registry.
- **CFEXECUTE:** Executes a program, batch file, or other executable on the server.
- **CFPOP:** Manages and retrieves e-mail messages from a POP e-mail server.

New features in ColdFusion MX:

ColdFusion MX is a new release of a die-hard product that has been completely redesigned from the ground up. This version of ColdFusion is Java-based, but works almost entirely like the previous versions of ColdFusion. In fact, most applications created in previous versions of ColdFusion will run without modification.

One of the biggest new developments in ColdFusion MX is that you can use JSP tag libraries with your ColdFusion pages. This means that a ColdFusion programmer can take advantage of any existing tag libraries, as well as create new ones. This is in addition to all of the ColdFusion tags and COM objects that are already in a ColdFusion programmer's arsenal.

Another new, big feature in ColdFusion MX is ColdFusion Components (CFCs). CFCs are self-documenting and self-contained. You can use CFCs through any ColdFusion pages on the ColdFusion server. Other technologies can access your ColdFusion components if you publish them as web services (for instance, a Macromedia Flash application may use a ColdFusion component to run a database query based on a user's request).

Create CFCs with the `cfcomponent` tag, and save them with a CFC extension. Include your CFML logic (such as a `cfquery` tag, conditional code, and so forth) within a `cffunction` tag inside of the `cfcomponent` tag inside the CFC file. Then, any ColdFusion page on your server can call the functions (also called methods) within the CFC. For full details on how to use CFCs, refer to the other articles in the ColdFusion MX Application Developer Center, and to the ColdFusion MX documentation book, *Developing ColdFusion MX Applications*. Developing ColdFusion components that produce web services has one major advantage over producing web services in other technologies—you can create a ColdFusion component web service with standard CFML code in a fraction of the time. Simply add the attribute `access="remote"` to the `cffunction` tag to

activate the CFC as a web service. The following is a simple CFC that reads the contents of a directory and returns a list of JPEG files in that directory:

```
<cfcomponent>
  <cffunction name="getJPGList" returnType="string">
    <cfdirectory name="fileList"
      action="list"
      directory="H:\public_jpegs"
      filter="*.jpg">
    <cfreturn #ValueList(fileList.name)#>
  </cffunction>
</cfcomponent>
```

You can use this CFC on your ColdFusion page to get a comma-separated list of all JPEG files in the H:\public_jpegs directory.

To invoke the CFC on a ColdFusion page, use this simple code:

```
<cfinvoke component="getFiles" method="getJPGList"
  ReturnVariable ="theList"/>
```

After the `cfinvoke` tag, use the variable, `theList`, which contains the entire directory of JPEGs. Transform this simple component into a web service. Doing so makes the information accessible to anyone on the Internet using any capable technology. Simply add `access="remote"` to the `cffunction` tag in the CFC. ColdFusion MX takes care of all the details, including creating the WDSL file for you. You've successfully published a web service that lists the contents of one of your directories. This is a simple example. Yet, compared to that of another language, it illustrates the simplicity and power inherent in ColdFusion code. Read more about using ColdFusion components in the ColdFusion MX Application Developer Center.

XML parsing:

Also, ColdFusion MX has XML parsing built into the language as well. A new tag in ColdFusion MX is the `<cfxml>` tag. Using `<cfxml>` makes it easy to create an XML document. Also, new ColdFusion functions can take an XML document and turn it into a ColdFusion structure to easily manipulate the contents.

ColdFusion MX allows you to address the XML construct as a structure, using dot notation. Assuming that the preceding XML structure is contained in string variable named books, the following code pulls apart the structure created by the XML tags and sets variables to hold the data:

```
<cfset getBook = XMLParse(books)>
<cfset bookTitle = getBook.books.book.title.xmltext>
<cfset bookIsbn = getBook.books.book.isbn.xmltext>
<cfset bookUrl = getBook.books.book.url.xmltext>
```

Note that the `xmltext` method will retrieve the text within the XML tags. To display the variables on the page, you can use simple `CFOUTPUT` tags:

```
<cfoutput>
Book Title: #bookTitle#<br>
ISBN: #bookIsbn#<br>
URL: <a href="#bookUrl#">#bookUrl#</a>
</cfoutput>
```

ColdFusion stores XML internally as a structure, so a simple `<cfdump>` tag can display the contents of the XML structure to your web page while you are debugging:

```
<cfdump var=#getBook#>
```

Session Management:

ColdFusion has a sophisticated session-handling mechanism that works even if the user has cookies turned off. Session state is maintained in server memory using a unique identifier, and this can be passed from page to page if you determine that the user has turned off cookies. This makes for a more user-friendly web experience, allowing information to be maintained easily across pages.

Summary:

CFML is a versatile language that is self-contained, easy-to-learn, and promotes rapid application development. Also, with the new additions of JSP tag libraries, ColdFusion Components, XML functionality, and web services in ColdFusion MX, ColdFusion is more versatile than ever before.