Migrating Applications to ColdFusion MX 7
# CONTENTS

**CHAPTER 1: Migrating Applications from ColdFusion 5.**

- Migrating to ColdFusion MX ........................................ 5
- Planning your migration ................................................. 6
  - Web server considerations .......................................... 6
  - Migration strategy using external web server ...................... 7
  - Migration strategy using built-in web server ..................... 8
- Using the Configuration and Settings Wizard ........................ 9
- Using the ColdFusion Code Compatibility Analyzer .................. 10
- Database operations .................................................... 10
  - Connecting to Microsoft Access databases ......................... 10
  - Connecting to a data source with ODBC Socket .................... 11
  - Session management .................................................. 11
  - QueryColumn object .................................................. 11
  - Data type of query results .......................................... 11
- Localization and character handling .................................. 12
  - Extended character code and encoding support .................... 12
  - Support for new currency formats .................................. 13
  - Changes to formatting functions .................................... 13
  - Other locale-specific changes ....................................... 16
  - Other changes .......................................................... 17
  - Advanced security .................................................... 17
  - SNMP support ......................................................... 17
  - Variables .............................................................. 17
  - Operators ............................................................. 17
  - CFML data types ...................................................... 18
  - Template handlers .................................................... 18
  - Case in forms ......................................................... 18
  - Objects for date/time ................................................ 18
  - Date pivot point for two-digit years ............................... 18
  - Request variable ..................................................... 19
  - Definition of pi ...................................................... 19
  - Null values ........................................................... 19
  - The thisTag scope .................................................... 19
  - Caching .............................................................. 19
  - Log files ............................................................... 20
  - Using cfform in a multihomed environment ......................... 20
  - Moving to IBM WebSphere ........................................... 20
  - The application.cfc file ............................................ 20
Tag and function examples ................................................. 21
Using ListSort in ColdFusion MX ................................... 21
Using the cfindex tag in ColdFusion MX ......................... 22

CHAPTER 2: Changes in ColdFusion MX 7 ......................... 23
Configurations ................................................................. 23
Using eform in a multithomed environment ...................... 24
The application.cfc file .................................................... 24
Running multiple ColdFusion versions .............................. 24
Other changes in ColdFusion MX 7 ................................. 24

CHAPTER 3: Tag and Function Changes ............................... 33
Tag changes since ColdFusion 5 ....................................... 33
New tags, attributes, values, and changes ......................... 34
Deprecated tags, attributes, and values ............................ 49
Obsoleted tags, attributes, and values ............................. 50
Function changes since ColdFusion 5 ............................... 51
New functions, parameters, values, and changes ................ 51
Deprecated functions, parameters, and values ................... 58
Obsoleted functions, parameters, and values .................... 59
CHAPTER 1
Migrating Applications from ColdFusion 5

This chapter describes migration and known compatibility issues between ColdFusion Server 5 and later versions, including ColdFusion MX 6.1 and ColdFusion MX 7 (referred to in this chapter as ColdFusion MX).

Contents

Migrating to ColdFusion MX ............................................................... 5
Planning your migration ................................................................. 6
Using the Configuration and Settings Wizard ................................. 9
Using the ColdFusion Code Compatibility Analyzer .......................... 10
Database operations ........................................................................ 10
Localization and character handling ............................................... 12
Other changes ................................................................................ 17
Tag and function examples .............................................................. 21

Migrating to ColdFusion MX

When you migrate your existing ColdFusion Server 5 applications to ColdFusion MX, you might encounter a few incompatibilities between ColdFusion 5 and ColdFusion MX. Macromedia strongly encourages you to develop a migration plan for porting existing ColdFusion 5 applications to ColdFusion MX. For more information, see “Planning your migration” on page 6.

Whether you encounter incompatibilities depends on the features of ColdFusion and the CFML elements that your existing applications use. This guide lists known incompatibilities and suggests possible fixes. Also, you can use the Code Compatibility Analyzer to facilitate locating and fixing incompatible code; for more information, see “Using the ColdFusion Code Compatibility Analyzer” on page 10.

Note: This guide provides migration and incompatibility information for customers moving from ColdFusion 5 to ColdFusion MX. Most of this information also applies to ColdFusion 4.5, so references to ColdFusion 5 also applies to ColdFusion 4.5. For information about earlier versions of ColdFusion, see the Support section of the Macromedia website (www.macromedia.com/support).
Before planning your migration, make sure that you understand the ColdFusion MX configuration options:

**Server configuration**  Lets you install one instance of ColdFusion MX 7 with an embedded J2EE server. This configuration most closely resembles the ColdFusion MX base release and other releases prior to ColdFusion MX, such as ColdFusion 5 and ColdFusion 4.5. This was formerly known as the stand-alone configuration.

**Multiserver configuration (Enterprise Edition only)**  Installs JRun and automatically deploys ColdFusion MX 7 in a separate JRun server instance. This configuration supports server instance creation and ColdFusion deployment in the ColdFusion MX Administrator and lets you manage ColdFusion MX 7 deployments on multiple JRun servers.

**J2EE configuration (Enterprise Edition only)**  Lets you deploy ColdFusion MX 7 as a Java application running on a Java 2 Enterprise Edition (J2EE) application server, either using the bundled license of JRun or using a third-party J2EE server, such as IBM WebSphere or BEA WebLogic. When you use the J2EE configuration, you can deploy ColdFusion MX 7 multiple times on a single computer.

**Planning your migration**

To facilitate the migration process, Macromedia recommends that you develop a migration strategy. Because every application is different, you must tailor your own migration strategy to meet your environment’s needs. This section contains two possible migration strategies, following a discussion of web server differences in ColdFusion MX.

**Web server considerations**

ColdFusion 5 has a simple relationship with the web server:

- To install ColdFusion, you must have a web server (such as IIS, iPlanet, or Apache) running on the same computer as ColdFusion.
- The ColdFusion Administrator is located beneath the web root in a directory named CFIDE; CFM pages are located either within the web root or in a virtual directory that is accessible to the web server.
- The ColdFusion stub, a web-server specific module that is always configured as part of a ColdFusion 5 installation, uses extension mappings to route CFM pages to the ColdFusion application server.

ColdFusion MX provides a web server connector (similar to the ColdFusion stub), which also uses extension mappings to route CFM pages to ColdFusion MX. However, because both the ColdFusion stub and the ColdFusion MX web server connector process CFM files, you cannot configure a web server to handle both ColdFusion 5 and ColdFusion MX.

If you must use a ColdFusion 5 application on the same computer that contains ColdFusion MX, you can access ColdFusion MX pages through the built-in web server. The built-in web server is an all-Java, HTTP 1.0 server that is not intended for use in a production environment. However, the built-in web server is particularly useful when ColdFusion 5 and ColdFusion MX applications must co-exist on the same computer until you complete the migration process.
When running in the server configuration, ColdFusion MX uses the built-in web server, as follows:

- The ColdFusion MX Administrator is located under the `cf_root/wwwroot` directory in a directory named CFIDE; CFM pages are located within the `cf_root/wwwroot` directory.
- To access pages through the built-in web server, specify port 8500 in the URL, as the following example shows:
  
  `http://localhost:8500/CFIDE/administrator/index.cfm`

 **Tip:** If you upgrade to ColdFusion MX 7 and are already using the built-in web server, the installer selects a different port number (typically 8501).

When running ColdFusion MX 7 in the multiserver configuration, ColdFusion MX uses the built-in web server, as follows:

- The ColdFusion MX Administrator is located under the `jrun_root/servers/coldfusion/cfusion-ear/cfusion-war` directory in a directory named CFIDE; CFM pages are located within the `jrun_root/servers/coldfusion/cfusion-ear/cfusion-war` directory.
- To access pages through the built-in web server, specify port 8300 in the URL, as the following example shows:
  
  `http://localhost:8300/CFIDE/administrator/index.cfm`

When running in the J2EE configuration, you use the web server capabilities of the underlying J2EE application server.

If you are using the built-in web server, you must switch to the web server connector before moving to production. You do this by running the Web Server Configuration Tool, as explained in *Installing and Using ColdFusion MX* and *Configuring and Administering ColdFusion MX*.

### Migration strategy using external web server

If your site has a highly controlled environment with separate development, testing, staging, and production servers, you can migrate to ColdFusion MX on a testing server before moving to production. Adapt the following suggested steps to fit your needs:

1. Study the ColdFusion MX Migration section of the Macromedia website ([www.macromedia.com/go/cfmxmigration](http://www.macromedia.com/go/cfmxmigration)) and this migration guide to anticipate the compatibility issues that might affect your system.
2. Copy your existing ColdFusion 5 application to a test computer that contains a web server.
3. Redefine data sources to access a test database. You should not run tests against a production database.
4. Ensure that your ColdFusion 5 application is functioning properly on the test computer before upgrading it to ColdFusion MX.
5. Install ColdFusion MX on the test computer. The Install Wizard detects the previous version of ColdFusion and prompts you for the following:
   ■ Location of previous ColdFusion directory.
   ■ Location of CFIDE directory.
   ■ Whether to switch your web server from the previous ColdFusion version to ColdFusion MX. When using this migration strategy, it’s best to switch your web server to ColdFusion MX.

6. Open the ColdFusion MX Administrator. When starting the ColdFusion MX Administrator for the first time, the Configuration and Settings Migration Wizard runs.

7. Answer the Configuration and Settings Migration Wizard questions, selecting all migration options.

8. Test the code in your application using the ColdFusion Code Compatibility Analyzer. Ensure that you specify the location of any custom tags or other code outside of the ColdFusion MX web root directory. For more information, see “Using the ColdFusion Code Compatibility Analyzer” on page 10.

9. Based on the results of the Code Compatibility Analyzer report and the incompatibilities listed in this guide, make a migration plan. For example, you might plan to immediately fix code that causes errors, such as `dbtype="dynamic"` in query tags, and to schedule and prioritize other fixes, such as replacing deprecated tags and using new functionality.

10. Make the necessary changes to your ColdFusion application and run the application on your test computer. Ensure that you test all application features under load, not just with a few users. If possible, use a source control system to manage these changes.

   **Tip:** Remove the `web_root/cfdocs` directory before moving to production.

Your application is now ready for production on ColdFusion MX.

**Note:** A number of testing applications are available for download from the web. For more information, see the Support section of the Macromedia website (www.macromedia.com/support).

**Migration strategy using built-in web server**

If your environment does not multiple computers for development, testing, and staging. You can install ColdFusion MX directly on a staging or production computer and use the built-in web server to test your application in co-exist mode. Adapt the following suggested steps to fit your needs:

1. Study the ColdFusion MX Migration section of the Macromedia website (www.macromedia.com/go/cfmxmigration) and this migration guide to anticipate the compatibility issues that might affect your system.

2. Install ColdFusion MX on your staging or production computer. Do not configure an external web server; select the built-in web server instead.

3. Run the Configuration and Settings Migration Wizard to complete the installation. If ColdFusion 5 is already installed on the computer, the wizard will migrate your settings to ColdFusion MX.
4. Define data sources in the ColdFusion MX Administrator. If possible, do not run tests against a production database.

5. Copy CFM files to the cf_root/wwwroot (server configuration) or jrun_root/servers/coldfusion/cfusion-ear/cfusion-war (multiserver configuration) directory.

6. Test the code in your application using the ColdFusion Code Compatibility Analyzer. For more information, see “Using the ColdFusion Code Compatibility Analyzer” on page 10.

7. Based on the results of the Code Compatibility Analyzer report and the incompatibilities listed in this guide, make a migration plan. For example, you might plan to immediately fix code that causes errors, such as dbtype="dynamic" in query tags, and to schedule and prioritize other fixes, such as replacing deprecated tags and using new functionality.

8. Make the necessary changes to your ColdFusion application and run the application using the built-in web server. Ensure that you test all application features under load, not just with a few users.

9. When the application is ready for production, perform the following steps:
   a Run the Web Server Configuration Tool for the web server used in production. For more information, see Installing and Using ColdFusion MX or Configuring and Administering ColdFusion MX.
   b Rename the CFIDE directory under your production web root.
   c Copy the CFIDE directory and the application's CFM files to your production web root. Do not copy the cfdocs directory to a production server.

Your application is now ready for production on ColdFusion MX.

**Using the Configuration and Settings Wizard**

If you have ColdFusion 5 installed on the same computer as ColdFusion MX, the Configuration and Settings Migration Wizard prompts you to migrate settings to the ColdFusion MX installation. For Windows, the wizard typically runs immediately after you finish the installation; for UNIX, the wizard runs the first time that you open the ColdFusion MX Administrator.

If you choose not to migrate settings, you can rerun the wizard at a later time by performing the following steps:

1. Close the ColdFusion MX Administrator, if it is not already closed.

2. Open cf_root/lib/adminconfig.xml in a text editor, change runmigrationwizard to true, and save the file.

3. Restart the ColdFusion MX Administrator.

You are prompted again to migrate your settings.

**Caution:** If you migrate your ColdFusion 5 settings later, you might overwrite new ColdFusion MX settings.
Using the ColdFusion Code Compatibility Analyzer

To isolate compatibility issues between ColdFusion MX and ColdFusion Server 5, use the Code Compatibility Analyzer to check your ColdFusion pages. Based on diagnostic information, the Code Compatibility Analyzer issues error and information messages to alert you to the following compatibility issues:

- Obsolete CFML features in ColdFusion MX
- Deprecated CFML features in ColdFusion MX
- CFML features that might produce different behaviors in ColdFusion MX
- CFML syntax errors

To access the Code Compatibility Analyzer, click the link in the ColdFusion MX Administrator.

When using the Code Compatibility Analyzer, follow these guidelines:

- To avoid time-consuming operations and possibly degraded server performance for an extended period of time, do not run the Code Compatibility Analyzer on multiple application directories.
- Use the advanced options to custom the tags, functions, and operators that are analyzed.

Database operations

ColdFusion MX uses JDBC drivers to interact with databases; for example, to query, write, and update a database. (JDBC is an Application Programming Interface (API) for Java programs to access data.) To connect to Open Database Connectivity (ODBC) data sources on Windows, you can use the Data Direct SQL Link Type III ODBC Socket database driver, which is included with ColdFusion MX.

This section describes the known incompatibilities between the ODBC features of ColdFusion 5 and the JDBC features of ColdFusion MX, in addition to those that are documented in Chapter 3, “Tag and Function Changes,” on page 33 (for example, cfquery, cfcatch, and cfreport).

For more information about the Data Direct JDBC drivers included with ColdFusion MX, see www.datadirect.com/index.ssp.

Connecting to Microsoft Access databases

The ColdFusion MX Administrator includes options that make it easy to define data sources for Microsoft Access databases. When defining a data source in the ColdFusion MX Administrator, select Microsoft Access as the driver type. The Administrator automatically configures the ODBC Socket driver with Access-specific settings.

Note: When using Microsoft Access on a Unicode system, select the Microsoft Access with Unicode driver. This is a JDBC Type 2 driver that works with Microsoft Access Unicode databases.
Connecting to a data source with ODBC Socket

In ColdFusion 5, an ODBC Data Source Name (DSN) entry that was created with the ODBC Data Source Administrator would automatically appear in the list of data sources in the ColdFusion Administrator. It does not appear automatically in ColdFusion MX; you must manually add the ODBC DSN entry.

To manually add the ODBC DSN entry:
1. In the ColdFusion MX Administrator, select Data & Services > Data Sources.
2. In the Add New Data Source Name box, do the following:
   a. In the Data Source Name text box, enter the data source name. You can use the same name as the ODBC DSN entry.
   b. In the Driver drop-down list box, select ODBC Socket.
   c. Click Add.
3. In the ODBC DSN drop-down list box, select the ODBC DSN name.
4. Click Submit.

For more information, see the “Data Source Management” chapter of Configuring and Administering ColdFusion MX.

Session management

ColdFusion MX lets you use J2EE sessions for session management. For backward compatibility, J2EE session management is disabled by default. However, you should enable this feature whenever possible. For more information, see “Using Persistent Data and Locking” in ColdFusion MX Developer’s Guide.

QueryColumn object

In ColdFusion MX, the QueryColumn object is even more like an array than in ColdFusion 5, as the following examples show:

- The QueryColumn object is derived by referencing a query using associative array notation; for example:
  ```
  query1['firstName']
  ```

- You can set a query column to an array element; for example:
  ```
  <cfset query1['col'] = array2['moo']>
  ```

Data type of query results

ColdFusion 5 stores data in the query object as a string, regardless of how the data is stored in the database. When it outputs a piece of data, it just writes out the string. ColdFusion MX stores data in the query object differently, depending on its database type (for example, it might store a number as a Java Double). ColdFusion MX outputs the data by converting the stored type to a string, which might differ from the string that ColdFusion 5 outputs. If you need your output to be in a particular format, use the number and/or date format functions.
Localization and character handling

This section describes the changes between ColdFusion 5 and ColdFusion MX that affect locale functions. It contains the following topics:

- Extended character code and encoding support
- Support for new currency formats
- Changes to formatting functions
- Other locale-specific changes

Note: Results from localization functions might differ, depending on your Java Runtime Environment (JRE).

Extended character code and encoding support

ColdFusion MX uses Java Unicode as its internal character representation and supports character code values 0–65535, whereas ColdFusion 5 supports 1-255. This affects string-processing functions. For more information, see Len in “New functions, parameters, values, and changes” on page 51.

ColdFusion MX uses the UTF-8 character set encoding as its default web page encoding. ColdFusion 5 used the ISO-8859-1 character set encoding. This can affect the way that characters in the range 128–255 are displayed in the browser. To restore the ColdFusion 5 default encoding, place the following cfcontent tag in your page (or in the application's Application.cfm page):

```html
<cfcontent type="text/html; charset=ISO-8859-1">
```

Note: UTF-8 (USC Transformation Format, where USC is the Universal Character Set) enables computers to handle both ASCII and Unicode. Unicode (Unicode Worldwide Character Standard) is a system of setting up binary codes for text or script characters, so that the characters from the principal written languages of the world can be displayed and processed.

ColdFusion MX and ColdFusion MX 6.1 have added several attributes, parameters, and functions for determining the character encoding used in tags and functions, as follows:

<table>
<thead>
<tr>
<th>Tag or function</th>
<th>Attribute or parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfhttp</td>
<td>charset</td>
</tr>
<tr>
<td>cfmail</td>
<td>charset</td>
</tr>
<tr>
<td>cfprocessingdirective</td>
<td>pageEncoding</td>
</tr>
<tr>
<td>GetEncoding</td>
<td>all</td>
</tr>
<tr>
<td>SetEncoding</td>
<td>all</td>
</tr>
<tr>
<td>URLDecode</td>
<td>charset</td>
</tr>
<tr>
<td>URLEncodedFormat</td>
<td>charset</td>
</tr>
</tbody>
</table>

Additionally, an option has been added to the Mail page of ColdFusion MX Administrator that specifies the default mail character encoding.
For more information on support for character encodings see the “Developing Globalized Applications” chapter in *ColdFusion MX Developer's Guide* and the individual tag and function descriptions in the *CFML Reference*.

**Support for new currency formats**

ColdFusion MX recognizes new currency formats for locales. For example, when a euro currency is passed to the `LSIsCurrency` function, ColdFusion MX returns Yes if the locale is a Euro member, and No if not. Support for currency formats is JVM dependent. For more information, see the `LSCurrencyFormat` and `LSIsCurrency` discussions in *CFML Reference*.

**Changes to formatting functions**

ColdFusion MX uses Java standard locale formatting rules on all platforms. ColdFusion 5 uses Windows or UNIX locale formatting rules, depending on the ColdFusion platform. For information on Java standard locale formatting rules, see the Java documentation for the Locale class (java.util.Locale) for your JVM at [http://java.sun.com/j2se/](http://java.sun.com/j2se/).

Some functions in ColdFusion MX produce slightly different results than in ColdFusion 5. For example, in ColdFusion MX, `LSCurrencyFormat` returns the correct international currency code for countries other than the US, and correctly rounds the currency values for countries whose currency does not include cents. The same code now produces the same results, regardless of the platform.

Some of the differences might require adjusting your ColdFusion applications. For example, `LSCurrencyFormat` in ColdFusion MX formats some negative currency values differently than in ColdFusion 5, such as inserting a leading minus sign (–) instead of enclosing the value in parentheses.
Changes to formatting functions

The following table outlines changes to formatting functions. For more information, see CFML Reference for the appropriate function.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description of change</th>
</tr>
</thead>
</table>
| LSEuroCurrencyFormat | In ColdFusion 5, the LSEuroCurrencyFormat function used EUR or the euro symbol (€) for every locale, even if the locale did not support the euro. For example, LSEuroCurrencyFormat(123.45, "international") returned "EUR123.45", even for an English (US) locale.  
In ColdFusion MX, LSEuroCurrencyFormat returns the euro currency format if the set locale is a Euro member country. Otherwise, it returns the set locale’s currency format or symbol. For example, LSEuroCurrencyFormat(-1234.56) returns -1234,56 € for the French (Standard) locale, but returns ($1,234.56) for the English (US) locale. |
| LSCurrencyFormat     | In ColdFusion 5 this function had numerous errors where it used the wrong international currency designator. For example, in an English (Australian) locale, LSCurrencyFormat(123456.78, "international") returned USD123,456.78. This problem has been corrected, and this example now returns AUD. 
The format returned for negative currency values has also been corrected to use JVM format for the locale. For example, in the Australian (English) locale, LSCurrencyFormat(-123456.78, "local") now returns -$123,456.78 instead of ($123,456.78). 
In ColdFusion 5, the LSCurrencyFormat tag used the pre-euro currency identifiers for all countries in the Euro zone. ColdFusion MX uses the Java standard locales, and its behavior depends upon the JVM version. 
If you have a 1.4 JVM, including the JVM that is included in the ColdFusion MX 6.1 server configuration, LSCurrencyFormat returns the euro currency format for all locales in the Euro Zone. If you have a 1.3 JVM, LSCurrencyFormat returns the non-euro currency format for all locales, even if the locale is a member of the Euro countries. |
In ColdFusion MX, **LSDateFormat** returns the appropriate date for the set locale. For example, **LSDateFormat(12/01/02)** returns Jan-12-2002 if the locale is set to English (UK), and Dec-01-2002 if the set locale is English(US).

In ColdFusion 5, **LSDateFormat** throws an error or returns an incorrect format.

When no mask is specified, ColdFusion MX returns the Java medium date format for the locale. ColdFusion 5 returns one of two default masks, depending on the locale:

- For a locale where the date ends with the year, the default mask is **dd-mmm-yy**.
- For a locale where the date begins with the year, the default mask is **yyyy-mmm-dd**.

The medium date format for English (US) has changed to the Java standard; for example, “Sept 4, 1998 11:15:16 PM” and “11:15:16 PM Sept. 4, 1998” are valid. However, “23:15:16 4 Sept. 1998” is not valid; for example, **LSParseDateTime(“23:15:16 4 Sept. 2003”)** causes an error.

For a list of valid date formats, see the Java documentation for the java.text.DateFormat class at [http://java.sun.com/j2se/1.3/docs/api/index.html](http://java.sun.com/j2se/1.3/docs/api/index.html).

If you pass an invalid mask (for example, *dd-mm-yyyy*), **LSDateFormat** outputs the invalid characters in both ColdFusion 5 and ColdFusion MX. However, ColdFusion 5 delimits each character of the output with an apostrophe (’) character.

**LSTimeFormat**

ColdFusion MX supports the following masks for **LSTimeFormat**:

- **h:m:s**  \( \text{LSTimeFormat(“6:39”,”h:m:s”) returns 6:39:0} \)
- **hh:mm:ss**  \( \text{LSTimeFormat(“6:39”,”hh:mm:ss”) returns 06:39:00} \)
- **hh:mm:ss t**  \( \text{LSTimeFormat(“6:28:26”,”hh:mm:ss t”) returns 06:28:26 A} \)
- **hh:mm:ss tt**  \( \text{LSTimeFormat(“6:28:26”,”hh:mm:ss tt”) returns 06:28:26 AM} \)

If you pass an invalid mask, **LSTimeFormat** outputs the invalid characters in both ColdFusion 5 and ColdFusion MX. However, ColdFusion 5 delimits each character of the output with an apostrophe (’) character.

**LSIsCurrency**

When a euro currency is passed into the **LSIsCurrency** function, ColdFusion MX returns Yes if the locale is a Euro member, and No if not, independent of the JVM version. With 1.3 level JVMs only, this function also returns Yes if the currency is in a Euro zone country’s pre-Euro currency format.

**LSIsDate**

In ColdFusion MX, you can pass in dates with the full date and full datetime parameters. These formats include the name of the day (for example, Wednesday).

<table>
<thead>
<tr>
<th>Function</th>
<th>Description of change</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSDateFormat</td>
<td>Returns the appropriate date for the set locale. For example, <strong>LSDateFormat(12/01/02)</strong> returns Jan-12-2002 if the locale is set to English (UK), and Dec-01-2002 if the set locale is English(US). In ColdFusion 5, <strong>LSDateFormat</strong> throws an error or returns an incorrect format. When no mask is specified, ColdFusion MX returns the Java medium date format for the locale. ColdFusion 5 returns one of two default masks, depending on the locale: • For a locale where the date ends with the year, the default mask is <strong>dd-mmm-yy</strong>. • For a locale where the date begins with the year, the default mask is <strong>yyyy-mmm-dd</strong>. The medium date format for English (US) has changed to the Java standard; for example, “Sept 4, 1998 11:15:16 PM” and “11:15:16 PM Sept. 4, 1998” are valid. However, “23:15:16 4 Sept. 1998” is not valid; for example, <strong>LSParseDateTime(“23:15:16 4 Sept. 2003”)</strong> causes an error. For a list of valid date formats, see the Java documentation for the java.text.DateFormat class at <a href="http://java.sun.com/j2se/1.3/docs/api/index.html">http://java.sun.com/j2se/1.3/docs/api/index.html</a>. If you pass an invalid mask (for example, <em>dd-mm-yyyy</em>), <strong>LSDateFormat</strong> outputs the invalid characters in both ColdFusion 5 and ColdFusion MX. However, ColdFusion 5 delimits each character of the output with an apostrophe (’) character.</td>
</tr>
</tbody>
</table>
| LSTimeFormat | ColdFusion MX supports the following masks for **LSTimeFormat**:
• **h:m:s**  \( \text{LSTimeFormat(“6:39”,”h:m:s”) returns 6:39:0} \)
• **hh:mm:ss**  \( \text{LSTimeFormat(“6:39”,”hh:mm:ss”) returns 06:39:00} \)
• **hh:mm:ss t**  \( \text{LSTimeFormat(“6:28:26”,”hh:mm:ss t”) returns 06:28:26 A} \)
• **hh:mm:ss tt**  \( \text{LSTimeFormat(“6:28:26”,”hh:mm:ss tt”) returns 06:28:26 AM} \)
If you pass an invalid mask, **LSTimeFormat** outputs the invalid characters in both ColdFusion 5 and ColdFusion MX. However, ColdFusion 5 delimits each character of the output with an apostrophe (’) character. |
| LSIsCurrency | When a euro currency is passed into the **LSIsCurrency** function, ColdFusion MX returns Yes if the locale is a Euro member, and No if not, independent of the JVM version. With 1.3 level JVMs only, this function also returns Yes if the currency is in a Euro zone country’s pre-Euro currency format. |
| LSIsDate | In ColdFusion MX, you can pass in dates with the full date and full datetime parameters. These formats include the name of the day (for example, Wednesday). |
Other locale-specific changes

Following are other locale-specific issues that could cause an incompatibility in your ColdFusion 5 applications:

- For the Japanese version, sort functions such as ListSort and ArraySort return items in a slightly different order in ColdFusion MX than they do in the Japanese version of ColdFusion 5. This is because ColdFusion MX sorts characters by their Unicode char number, and the Japanese version of ColdFusion 5 sorts characters by their shift_jis char number.

- The format for how decimal numbers appear in a browser depends on the system locale of the computer that is hosting ColdFusion MX, not on the locale of ColdFusion MX. For example, if you run the English version of ColdFusion MX on a computer whose system locale is set to German or French, the decimal point displays as a comma, not a period. ColdFusion 5 displays a period for the decimal point, regardless of the system locale.

- The GetLocale function checks if the locale was explicitly set; for example, from Setlocale. If not, ColdFusion MX gets the default locale for the operating system from the JVM. If the operating system has no set locale, or if ColdFusion MX does not support the operating system’s default locale, GetLocale returns en_US "English (US)". (ColdFusion MX sets the locale in the JVM to en_US "English (US)"; this value persists until the server is restarted or the value is reset with the SetLocale function. This setting does not affect anything but ColdFusion MX.)

For more information, see CFML Reference for the relevant function.
Other changes

This section describes other changes between ColdFusion 5 and ColdFusion MX.

Advanced security

ColdFusion MX provides a new and easy way for you to build user authentication and roles-based security into your applications. However, it is based on a completely different security model than ColdFusion 5. Also, ColdFusion MX no longer includes a licensed version of Netegrity Siteminder. Therefore, any existing Advanced Security code—including the \texttt{cfauthenticate} and \texttt{cfimpersonate} tags, and the \texttt{authenticatedContext}, \texttt{authenticatedUser}, \texttt{isAuthenticated}, \texttt{isProtected}, and \texttt{isAuthorized} functions—no longer works in ColdFusion MX. These tags and functions are obsolete in ColdFusion MX.

For more information, see \textit{Configuring and Administering ColdFusion MX} and the “Application Security” chapter in \textit{ColdFusion MX Developer’s Guide}.

SNMP support

ColdFusion MX no longer supports Simple Network Management Protocol (SNMP) for monitoring ColdFusion applications from enterprise management systems.

Variables

ColdFusion MX includes the following changes to variables:

- You can no longer use a dot (.) in a variable name, because ColdFusion MX supports the dot notation as a dot operator to create a struct. For example, \texttt{last.name} creates a struct called \texttt{last} with a key called \texttt{name}, instead of creating a simple variable with a period in the name.
  
  To work around this, use underscores in variable names instead.

- The following \texttt{cfcatch} variables have changed: \texttt{Message}, \texttt{NativeErrorCode}, and \texttt{SQLState}.
  
  For more information, see \texttt{cfcatch} in “New tags, attributes, values, and changes” on page 34.

Operators

ColdFusion MX includes the following changes to operands:

- Exponent results differ; for example, ColdFusion 5 returns an error for $0^3$, and ColdFusion MX only returns 1.

- ColdFusion MX supports the dot notation as a dot operator. For example, if your ColdFusion 5 application has a variable called \texttt{last.name}, ColdFusion MX reads this code and creates a struct called \texttt{last} with a key called \texttt{name}.

  In addition, ColdFusion MX does not create keys with dots in them like ColdFusion 5 does, but instead creates cascading structs with non-dotted names. For example, ColdFusion 5 interprets \texttt{a.b.c="foo"} to be \texttt{a[b.c]="foo"}, whereas ColdFusion MX interprets it to be \texttt{a[b][c]="foo"}. 
CFML data types

ColdFusion MX preserves the case of a struct key, whereas ColdFusion 5 sets every struct key to uppercase. However, you cannot use dot notation with different case to create more than one key. For example, the following code produces one key, not two:

```coldfusion
x.Foo = 1;
x.FOO = 2;
```

If you specify `x['Foo']`, ColdFusion MX preserves case. However, for best results, use consistent case for struct key names.

Template handlers

The Runtime service in Windows has been reimplemented in ColdFusion MX so that you can specify a template handler in the ColdFusion MX Administrator (Settings page). (A template handler is a template that executes when ColdFusion MX cannot find the CFM file specified in a `cfinclude` tag.) However, you must specify a template handler as a path that is relative to the web root directory, such as `web_root\missing.cfm` for the file `web_root\missing.cfm`. Do not use a full path.

Case in forms

Unlike ColdFusion 5, ColdFusion MX preserves the case of field names submitted by forms, instead of forcing them to uppercase. Like ColdFusion 5, ColdFusion MX ignores case when evaluating field names (for example, `@form.myfield@` is the same as `@FORM.MYFIELD@`). However, you should change templates that use `Find` or `ListFind` to search through the `Form.Fieldnames` variable, to use case-insensitive equivalents.

Objects for date/time

ColdFusion MX no longer strips milliseconds off date/time objects; to do this, use the date formatting functions.

Date pivot point for two-digit years

When ColdFusion processes a date that has does not include the century, ColdFusion determines the century. To do this, ColdFusion compares the date to when it processes the date. With a few exceptions, ColdFusion MX uses 20 as the date pivot point for calculating the century in a two-digit year. ColdFusion 5 uses 29.

*Note:* The following locales use 28 as the date pivot point instead of 20: English (Australian), English (New Zealand), German (Austrian), German (Standard), German (Swiss), Portuguese (Brazilian), Portuguese (Standard), and Swedish. Macromedia has confirmed this inconsistency using the SUN JRE version 1.4 and the IBM JRE version 1.3.0.
So in ColdFusion MX, if the date is within 80 years before and 20 years after the date when it processes the date, then ColdFusion MX returns the date with the current century. Otherwise, it returns the previous century. In ColdFusion 5, if the date is within 71 years before and 29 years after the date when it processes the date, then ColdFusion 5 returns the current century. Otherwise, it returns the previous century. This can cause incompatible results. For example, LSParseDateTime("25-12-13 22:10:15") returns a year of 1925 in ColdFusion MX, and 2025 in ColdFusion 5.

**Note:** Results might differ, depending on your Java Runtime Environment (JRE). This information reflects tests on the SUN JRE version 1.4 and the IBM JRE version 1.3.0.

For more information, see the version-specific Java documentation for the `java.text.SimpleDateFormat` class, at [http://java.sun.com/j2se/](http://java.sun.com/j2se/).

**Request variable**

ColdFusion MX no longer supports the `RequestTimeout` attribute in the URL. For pages expecting the `RequestTimeout` attribute in the URL, add the following tag:

```cfsetting RequestTimeout = "#URL.RequestTimeout#"```

For more information, see the “Reserved Words and Variables” chapter in *CFML Reference*.

**Definition of pi**

ColdFusion MX and ColdFusion 5 define pi with slightly different precision. Therefore, you might obtain slightly different results between the two for trigonometric functions such as sin, cos, and tan at the boundaries (0, 90, 180, 270, ... degrees).

**Null values**

ColdFusion 5 converted all null values to an empty string ("""). For queries, ColdFusion MX preserves null values, but converts them to an empty string when you use them as simple values. However, when you call a Java method that returns a null value, ColdFusion MX destroys the variable to which the null value was returned.

**The thisTag scope**

In ColdFusion 5, you can change system values in the thisTag scope that persist throughout the execution of the tag. This is fixed in ColdFusion MX; it throws an exception if you try to set `thisTag.hasendtag` or `thisTag.executionmode`.

**Caching**

In ColdFusion MX, in the Caching page of the ColdFusion MX Administrator, the Template cache size (number of templates) option displays the number of templates in the cache. In ColdFusion 5, it displays the size of the cached templates in kilobytes.
Log files

ColdFusion MX does not support the following ColdFusion 5 log files:

- executive.log
- remote.log
- proxy.log
- cfadmin.log
- install.log
- server.stdout
- rdsservice.stdout

Using cfform in a multihomed environment

In ColdFusion 5, the `cfform` tag used inline JavaScript code for validation. In ColdFusion MX, `cfform` includes the CFIDE/scripts/cfform.js file, which contains all JavaScript. However, if your application runs in a hosted environment with multiple virtual websites, your virtual website may not contain a CFIDE directory, which causes JavaScript errors when using `cfform`.

If you encounter this problem, use one of the following solutions:

- Use the Default CFFORM ScriptSrc Directory field on the ColdFusion MX Administrator to specify the default path (relative to the web root) of the directory containing the cfform.js file.
- Ask your hosting provider to provide a copy of cfform.js and change your cfform tags to use the `scriptsrc` attribute, specifying the location of this file.
- Ask your hosting provider to modify your virtual website to include a virtual mapping for the CFIDE directory.

Moving to IBM WebSphere

IBM WebSphere does not use web server virtual mappings to locate CFM pages. If you are migrating a ColdFusion 5 application to ColdFusion MX running on WebSphere using the J2EE configuration and your application stores CFM pages in directories that use virtual mappings, you must reconfigure your application and move these pages to a nonvirtual directory.

For more information, see the WebSphere pages on www.macromedia.com/go/cfmxj2ee-cert.

The application.cfc file

The updated application framework in ColdFusion MX 7 lets you use a file named Application.cfc to control your application and keep track of application events. You use Application.cfc instead of Application.cfm. Application.cfc methods are triggered when application, sessions, and requests start or stop, and when the application encounters an exception.

If your application already uses a file named Application.cfc, you must rename it.
Tag and function examples

This section provides remedies for some of the compatibility issues listed in Chapter 3, “Tag and Function Changes.”

Using cfregistry in ColdFusion MX

The `cfregistry` tag is deprecated for UNIX and Linux platforms. Therefore, you must remove it on UNIX to avoid compatibility problems when migrating to a later version of ColdFusion MX.

As an alternative, you can store variables in the following scopes:

- Client
- Text file
- Encoded text file
- Database
- LDAP server

In Windows, you can continue to store client variables in the registry as in ColdFusion 5. However, you should not use `cfregistry` to read ColdFusion settings from the registry, such as the mail root and information about Verity collections and scheduled tasks, because much of this information is no longer stored in the registry, or it is in a different location in the registry. The ColdFusion settings that are stored in the registry are subject to change. Therefore, you should not rely on registry keys that ColdFusion creates to store its system settings.

Using ListSort in ColdFusion MX

When sorting with `textnocase` in a descending order, ColdFusion 5 and ColdFusion MX return the results in a different order. However, both are correct because in a `textnocase` sort, “apple” is equal to “APPLE”.

In ColdFusion MX, a descending `textnocase` sort returns the elements in the exact reverse order as in an ascending `textnocase` sort. This is different than in ColdFusion 5.

The following code produces different results in ColdFusion MX than in ColdFusion 5:

```html
<cfset list = "orange,Orange,apple">
<cfset listAsc = ListSort(#list#, "textnocase", "asc")>
<cfset listDesc = ListSort(#list#, "textnocase", "desc")>
<cfdump var = #listAsc#>
<cfdump var = #listDesc#>
```

This code produces the following results for ascending and descending sort operations in ColdFusion 5 and ColdFusion MX:

<table>
<thead>
<tr>
<th>Sort order</th>
<th>ColdFusion 5 results</th>
<th>ColdFusion MX results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ascending</td>
<td>apple, orange,Orange</td>
<td>apple,orange,Orange</td>
</tr>
<tr>
<td>Descending</td>
<td>orange, Orange, apple</td>
<td>Orange,orange,apple</td>
</tr>
</tbody>
</table>

For more information, see the `ListSort` function in CFML Reference.
Using the cfindex tag in ColdFusion MX

To populate a collection with the contents of the query results, you can now use the \texttt{cfindex} tag with the \texttt{query} attribute and \texttt{type = "file"} or \texttt{type = "path"}, for all actions that require information from the \texttt{key} attribute. You can also still use \texttt{type = "custom"}.

When you use \texttt{type = "file"} or \texttt{type = "path"} with a query, the \texttt{action} attribute queries to get filenames or file paths from the \texttt{key} attribute, and passes the query results to its actions. The actions use the filenames or file paths to execute their code.

The following table shows a sample database for an application on a Windows server:

<table>
<thead>
<tr>
<th>BookID</th>
<th>URL</th>
<th>Title</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookid1</td>
<td>url1</td>
<td>title1</td>
<td>description1</td>
</tr>
<tr>
<td>bookid2</td>
<td>url2</td>
<td>title2</td>
<td>description2</td>
</tr>
<tr>
<td>bookid3</td>
<td>url3</td>
<td>title3</td>
<td>description3</td>
</tr>
<tr>
<td>file</td>
<td><a href="http://localhost/cfdocs/snippets">http://localhost/cfdocs/snippets</a></td>
<td>title4</td>
<td>c:\inetpub\wwwroot\cfdocs\snippets\file.cfm</td>
</tr>
<tr>
<td>path1</td>
<td><a href="http://localhost/cfdocs/snippets/">http://localhost/cfdocs/snippets/</a></td>
<td>title5</td>
<td>c:\inetpub\wwwroot\cfdocs\snippets\path1.cfm</td>
</tr>
<tr>
<td>path2</td>
<td><a href="http://localhost/cfdocs/">http://localhost/cfdocs/</a> cfmlsyntaxcheck</td>
<td>title6</td>
<td>c:\inetpub\wwwroot\cfdocs\cfmlsyntaxcheck</td>
</tr>
</tbody>
</table>

You can populate a collection using either of the following scripts:

- Populate the \texttt{snippets} collection with files specified in the description column of the database:

  \begin{verbatim}
  <CFQUERY NAME = "bookquery"
  DATASOURCE = "book">
  SELECT * FROM book where bookid='file'
  </CFQUERY>
  
  <CFOUTPUT QUERY = "bookquery">
  #url#, #description# <BR>
  <cfindex collection = "snippets" action = "update" type = "file"
  query = "bookquery" key = "description" URLPath = "url">
  </CFOUTPUT>
  
  - Populate the \texttt{snippets} collection with paths specified in the description column of the database:

  \begin{verbatim}
  <CFQUERY NAME="bookquery"
  DATASOURCE="book">
  SELECT * FROM book where bookid='path1' or bookid='path2'
  </CFQUERY>
  
  <CFOUTPUT QUERY="bookquery">
  #url#, #description# <BR>
  <cfindex collection="snippets" action="update"
  type="path" query="bookquery" key="description" URLpath="url">
  </CFOUTPUT>
  \end{verbatim}
CHAPTER 2
Changes in ColdFusion MX 7

This chapter describes functionality changes in ColdFusion MX 7. It does not include information on tag and function changes. For information on tag and function changes, see Chapter 3, “Tag and Function Changes.”

Contents

Configurations ................................................................. 23
Using cfform in a multihomed environment ......................... 24
The application.cfc file ....................................................... 24
Running multiple ColdFusion versions ................................. 24
Other changes in ColdFusion MX 7 ....................................... 24

Configurations

You can install ColdFusion MX 7 in the following configurations:

**Server configuration**  Lets you install one instance of ColdFusion MX 7 with an embedded J2EE server. This configuration most closely resembles the ColdFusion MX base release and other releases prior to ColdFusion MX, such as ColdFusion 5 and ColdFusion 4.5. This was formerly known as the stand-alone configuration.

**Multiserver configuration (Enterprise Edition only)**  Installs JRun and automatically deploys ColdFusion MX 7 in a separate JRun server instance. This configuration contains the Enterprise Manager, which lets you create server instances with ColdFusion automatically deployed as well as manage clusters and ColdFusion MX 7 deployments on multiple JRun servers.

**Note:** If you installed ColdFusion MX 6.1 using the ColdFusion MX with JRun 4 option and wish to use the Enterprise Manager, you must uninstall JRun before installing ColdFusion MX 7. Alternatively, you can continue defining server instances using the JMC.

**J2EE configuration (Enterprise Edition only)**  Lets you deploy ColdFusion MX 7 as a Java application running on a Java 2 Enterprise Edition (J2EE) application server, either using the bundled license of JRun or using a third-party J2EE server, such as IBM WebSphere or BEA WebLogic. When you use the J2EE configuration, you can deploy ColdFusion MX 7 multiple times on a single computer.
Using `cfform` in a multihomed environment

In ColdFusion 5, the `cfform` tag used inline JavaScript code for validation. In ColdFusion MX, `cfform` includes the CFIDE/scripts/cfform.js file, which contains all JavaScript. However, if your application runs in a hosted environment with multiple virtual websites, your virtual website may not contain a CFIDE directory, which causes JavaScript errors when using `cfform`.

If you encounter this problem, use one of the following solutions:

- Use the Default CFFORM ScriptSrc Directory field on the ColdFusion MX Administrator to specify the default path (relative to the web root) of the directory containing the cfform.js file.
- Ask your hosting provider to provide a copy of cfform.js and change your cfform tags to use the `scriptsrc` attribute, specifying the location of this file.
- Ask your hosting provider to modify your virtual website to include a virtual mapping for the CFIDE directory.

The `application.cfc` file

The updated application framework in ColdFusion MX 7 lets you use a file named `Application.cfc` to control your application and keep track of application events. You use `Application.cfc` instead of `Application.cfm`. `Application.cfc` methods are triggered when application, sessions, and requests start or stop, and when the application encounters an exception.

If your application already uses a file named `Application.cfc`, you must rename it.

Running multiple ColdFusion versions

If you run ColdFusion MX 7 alongside a previous ColdFusion version, only one version can use an external web server. In this case, use the built-in web server with ColdFusion MX 7 until you are ready to switch the external web server. At that point, run the Web Server Configuration Tool, as described in *Configuring and Administering ColdFusion MX*.

Scheduler considerations

When you install ColdFusion MX 7 on a computer that is running a previous ColdFusion version, the ColdFusion Administrator automatically migrates a variety of settings, including scheduled tasks. If you run ColdFusion MX 7 alongside ColdFusion MX or ColdFusion MX 6.1, scheduled tasks with the same name can collide. To avoid this issue, open the ColdFusion Administrator and rename scheduled tasks for one version or the other.

Other changes in ColdFusion MX 7

This section describes changes and enhancements in ColdFusion MX 7. Many of these changes may result in changes to your applications. For a complete list of features, enhancements, and changes in ColdFusion MX 7, see the Release Notes at [www.macromedia.com/go/cfmx7_releasenotes](http://www.macromedia.com/go/cfmx7_releasenotes).
Forms and validation enhancements

If your application uses the `cfform` tag or server validation, ensure that you review the enhancement discussions in the Release Notes. In particular, rich forms (using Macromedia Flash) and skinnable forms (using XML) provide major enhancements to form display and processing.

Improvements to the `cfchart` tag

ColdFusion MX 7 includes the following enhancements to the `cfchart` tag:

• An updated charting engine provides improved output.
• Use the new `title` attribute in the `cfchart` tag to specify the title of the chart.
• You can now use dates on the y-axis of a chart.
• The `cfchart` tag lets you specify the style of the chart in either an XML file or string.

Query of Queries

Changes to the Query of Queries include the following:

• More lenient type comparisons. Query of queries now makes the best effort to compare expressions of different types by doing implicit casts.
• Support for the `CAST (expression AS casttype)` function.
• Support for the varchar concatenation operator `||`.
• Support for SQL and ODBC style dates inside a query of queries.
• `Size` is no longer a reserved word when you create a query of queries, so you can create a query of queries using the `cfdirectory` tag with a column named `Size`.
• Query metadata support. Lets you specify data type metadata for query columns that you create using the `QueryNew` and `QueryAddColumn` functions. The following parameters have been added to these functions:
  ■ `QueryAddColumn` `datatype` parameter
  ■ `QueryNew` `columnTypeList` parameter

Verity upgrade and enhancements

The ColdFusion MX Verity implementation has been completely rewritten to use Verity 5.5 and its all-Java interface, which has many implications, including the following:

• Verity 5.5 uses a client-server paradigm. ColdFusion MX no longer manages collections directly and all collections are managed by the Verity search server.
• There is no longer the concept of external and internal collections.
• Verity pages in the ColdFusion MX Administrator have changed.
• Because ColdFusion uses a pure Java interface to Verity, native libraries are no longer required.
• The Verity search service consists of three separate processes:
  
  **k2server** for searching
  **k2index** for indexing
  **k2admin** for administration

• Verity searches now return keyword scores for the alternate keywords returned in the Status structure.

• Although the Verity search server is very similar to K2 functionality that was previously available in ColdFusion MX, the interface between ColdFusion and Verity has completely changed. For example, there no longer is a k2server.ini file.

• There are major changes to the following tags. For more information, see *CFML Reference*.
  ■ cfcollection
  ■ cfindex
  ■ cfsearch

• There are additional fields in the search results.

• In general, it is no longer necessary to wrap **cfcollection**, **cfindex**, and **cfsearch** calls within a **cflock** tag.

For more information, see *CFML Reference* and *ColdFusion MX Developer’s Guide*.

**Web services enhancements**

Web services enhancements and changes include the following:

• The **CreateObject** function and the **port** attribute of the **cfcomponent** tag let you select the port to use in a WSDL that contains more than one service, such as the MapPoint web service.

• The **cfinvokeargument** tag has a new **omit** attribute that lets you omit arguments to a web service that you invoke.

• The following new **cfcomponent** attributes and extensions to existing attributes support ColdFusion MX when publishing document-literal style web services:
  ■ **style=“document”** switches the emitted WSDL to document/literal style.
  ■ **hint** is now used as the document element of the service in WSDL.
  ■ **displayName** is now used as the name of the `<service>` element in WSDL.
  ■ **servicePortName** is used as the name of the `<port>` element in the service.
  ■ **portTypeName** is used as the name of the `<portType>` element in WSDL.
  ■ **bindingName** is used as the name of the `<binding>` element in WSDL.
  ■ **namespace** is used as the default namespace of the service.
  ■ **wsdlFile=“path to file”** causes ColdFusion MX to use the specified file as the WSDL instead of generating WSDL on the fly.
• The following new functions enable access to SOAP headers in CFML clients and ColdFusion MX web services:
  ■ AddSOAPRequestHeader
  ■ AddSOAPResponseHeader
  ■ GetSOAPRequestHeader
  ■ GetSOAPResponseHeader
  ■ GetSOAPRequest
  ■ GetSOAPResponse
  ■ IsSOAPRequest

  For more information, see CFML Reference and ColdFusion MX Developer’s Guide.

ColdFusion components (CFCs)
  You can now call super methods using named arguments.
  CFCs can now return arrays of CFC instances.

Generating Java nulls
  You can now generate Java nulls when calling a Java function. For more information, see the JavaCast function in CFML Reference.

Support for JNDI data sources under J2EE
  When running the multiserver or J2EE configuration, the Data Sources page of the ColdFusion MX Administrator lets you specify a J2EE Data Source (JNDI). After it is defined, you can specify the data source name in the datasource attribute of SQL tags to access the J2EE data source. Datasources found in this way cannot specify additional JNDI properties.

Changes to J2EE configuration requirements
  The following changes have been made to configuration requirements when deploying ColdFusion MX as an EAR or WAR:

  Verity changes  ColdFusion MX 7 uses Java interfaces to communicate with Verity. You no longer need to add Verity binary file directories to your system path.

  Classloader changes  You no longer need to add cfusion/lib to the system classpath. ColdFusion MX 7 loads all JAR files in the cfusion/lib and cfusion/lib/update directories. Additionally, ColdFusion MX 7 has its own classloader, for which you define the classpath using the cf.class.path element in the WEB-INF/web.xml file.

  Tip: In the multiserver configuration, the installer creates a JRun user name admin, with the same password as the ColdFusion Administrator password.
Support for strong encryption algorithms

Additional parameters were added to the `Encrypt` and `Decrypt` functions after the ColdFusion MX 7 manuals went sent to the printer. These changes make it possible to interact with password-based encryption algorithms and other advanced encryption technologies.

The function signatures are now as follows:

```plaintext
Encrypt(string, key [ ,algorithm [ ,encoding [ ,IVorSalt [ ,iterations ] ] ] ] ]
Decrypt(string, key [ ,algorithm [ ,encoding [ ,IVorSalt [ ,iterations ] ] ] ] ]
```

For more information, see the TechNote at [www.macromedia.com/go/cfmx7_encrypt](http://www.macromedia.com/go/cfmx7_encrypt).

OEM updates

ColdFusion MX 7 includes updated versions of the following:

- Sun 1.4.2_05 JVM
- Verity 5.5 Patch level 12
- DataDirect Type 4 - 3.3 drivers build 48
- SequelLink 5.4 ODBC with Unicode Support
- MySQL 3.23.47
- Jintegra upgraded from COM 2.1 to 2.2
- Axis 1.2 RC2 plus fixes

Enhanced string and data encoding and decoding

Use the following new functions for encoding and decoding text and data:

- `BinaryEncode` Converts binary data to a string in hexadecimal, UU, or base64 encoding.
- `BinaryDecode` Converts a string containing data in hexadecimal, UU, or base64 binary encoding to binary data.
- `CharsetEncode` Converts a string in a specified character set to a binary object.
- `CharsetDecode` Converts a binary object to a string using a specified character set.

Additionally, the `Encrypt` and `Decrypt` functions have been enhanced. For more information, see “Support for strong encryption algorithms” on page 28.

XML support enhancements

XML handling has been enhanced with the addition of several new functions, including decision and information functions, the ability to create an XML document object, and validation functions and attributes. Support has been added for validation against both XML schemas and DTDs.
ColdFusion MX 7 includes the following new or modified XML functions, parameters, and attributes:

- IsXML
- IsXMLAttribute
- IsXMLDoc
- IsXMLElem
- IsXMLNode
- IsXMLRoot
- XmlGetNodeType
- XmlNew
- XmlParse
- XmlValidate

Additionally, ColdFusion MX 7 includes support for specifying a document path or URL, including a relative URL, in the XMLParse xmlText attribute.

**Sandbox security**

Sandbox security lets you secure the CreateObject function by type. You can now turn off access to Java, COM, or Web Service objects without turning off access to ColdFusion components through the CreateObject function.

**Note:** Turning off access to Java, COM, or Web Service objects using the CreateObject function also turns off access using the cfobject tag.

**User-defined functions**

You can now call user-defined functions using the same name as the following obsolete advanced security functions: IsAuthenticated, IsAuthorized, IsProtected, AuthenticatedUser, and AuthenticateContext.

**Localization**

ColdFusion MX 7 supports all the locales that Java supports. You can use the GetLocaleDisplayName function to get a locale value and display the name in a manner that is appropriate to a specific locale.

**Enhanced font support**

ColdFusion MX 7 supports font files that do not have a default code page, or that have non-ASCII names.

**Enhanced PDF and SWF streaming support**

ColdFusion MX 7 can now recognize Microsoft Internet Explorer agent conttype requests, and can send back content type responses. This results in more reliable PDF and SWF streaming in Internet Explorer.
Data source changes

The cfsnippets, CompanyInfo, and CFExamples data sources have been combined into the cfdocexample data source. This database includes all tables from the previous three data sources. It is provided to support the ColdFusion documentation code examples and does not necessarily represent best practices in database design (for example, there is an employee table and an employees table).

For more information on these changes, see *CFML Reference*.

By default, the Configuration Wizard defines sample data sources using the Microsoft Access with Unicode driver.

Multiple instance management

In ColdFusion MX 6.1, you managed multiple server instances and clusters of multiple server instances through the JRun Management Console (JMC). This functionality is now available through the Enterprise Manager area of the ColdFusion MX Administrator.

**Note:** The Enterprise Manager is only available if you install ColdFusion MX 7 using the multiserver configuration.

If you installed ColdFusion MX 6.1 using the ColdFusion MX with JRun 4 option and want to use the Enterprise Manager, you must uninstall JRun before installing ColdFusion MX 7. Alternatively, you can continue defining server instances using the JMC.

For more information on this changes, see *Configuring and Administering ColdFusion MX* or the ColdFusion MX Administrator online Help.

ColdFusion MX Administrator

In addition to changes that support new features, the ColdFusion MX Administrator contains the following changes:

- New look and feel
- The Settings page includes a Default CFFORM ScriptSrc Directory field, which lets you specify the default path (relative to the web root) to the directory containing the cfform.js file.
- Removed StringFormat from the data source pages for DB2 and Informix.
- Certain options, such as Maximum Number of Simultaneous Request, which were previously unavailable in the J2EE configuration, are now available in the multiserver configuration even though, technically, it is a variant of the J2EE configuration.

The cfmail tag and malformed e-mail addresses

In previous releases, the `cfmail` tag wrote invalid e-mail addresses to an undelivered directory. In ColdFusion MX 7, the `cfmail` tag returns an error if one of the recipient attributes contains a malformed e-mail address. Your application should validate e-mail address format before invoking `cfmail`. 
Crystal Reports integration

ColdFusion MX 7 supports Crystal Reports versions 9 and 10. Additionally, ColdFusion MX 7 now uses the RDC technology from the Crystal Reports API.

You must install Crystal Reports with the following options:

- Enable export to disk
- Enable export to HTML

Crystal Reports does not enable this options by default, so you must use the Custom Install option when installing Crystal Reports.

**Tip:** If you run the Crystal Report Designer and export a report to an HTML file, the necessary export options are installed dynamically.

ColdFusion MX 7 uses COM to call the following Crystal Reports modules:

**Version 9**  Craxdrt9.dll

**Version 10**  Craxdrt.dll

These files are typically located in
C:\Program Files\Crystal Decisions\Report Designer Component.

Web Server Configuration Tool and web server connector

The Web Server Configuration Tool and the web server connector have the following changes:

- There are new parameters for the command-line Web Server Configuration Tool. For more information, see Configuring and Administering ColdFusion MX.

- The Web Server Configuration Tool creates the following additional properties:
  - ProxyRetryInterval  The number of seconds to wait before trying to reconnect to an unreachable clustered server. The default value is 600.
  - ConnectTimeout  The number of seconds to wait on a socket connect to a JRun server. The default value is 15.
  - RecvTimeout  The number of seconds to wait on a socket receive to a JRun server. The default value is 300.
  - SendTimeout  The number of seconds to wait on a socket send to a JRun server. The default value is 15.

- In addition to the suffix mappings used previously (.cfm, .cfml, .cfc, .jsp, and .jws) the Web Server Connector Tool now creates web server mappings for .cfr (used by ColdFusion reports) and .cfswf (used for Flash forms generated by the cfform tag).

- In previous ColdFusion versions, web server connector source code was made available in a connectors/src directory. In ColdFusion MX 7, web server connector source code is in the wsconfig.jar file, located in cf_root/runtime/lib (server configuration) or jrun_root/lib (multiserver configuration).

For more information on web server configuration, see Configuring and Administering ColdFusion MX.
Other installation and configuration changes

The following changes have been made to ColdFusion configuration:

- In the server configuration, the server name is coldfusion (cf_root/runtime/servers/coldfusion). In ColdFusion MX 6.1, the server name was default (cf_root/runtime/servers/default).
- The installer for the multiserver configuration (formerly known as the ColdFusion MX with JRun 4 option) automatically creates a JRun user named admin with the same password as the ColdFusion MX Administrator password. Use this user ID and password to log into the JRun Management Console (JMC).
- The ColdFusion MX 7 installer now includes processes formerly done by the Configuration Wizard.
- ColdFusion MX 7 includes Flash Remoting MX 2004, which supports ActionScript 2.0 and uses a component. ServerSide ActionScript is deprecated in ColdFusion MX 7.
- During a multiserver configuration installation, the default JRun server is not installed, which means that the jrun_root/servers/default directory structure is not created, and a service for the JRun default server is not defined.
- The ports for the ODBC services have been changed from 19997 and 19998 to 19995 and 19996 in order to let ColdFusion MX and ColdFusion MX 7 ODBC services run side-by-side.
- In the Developer Edition, in addition to localhost, ColdFusion MX 7 allows two additional IP addresses.
- The JavaMail.jar file has been updated to version 1.3.1.
- The cfx.jar file has been moved to one of the following locations:
  - Server configuration cf_root/wwwroot/WEB-INF/lib
  - Multiserver configuration jrun_root/servers/cfusion/cfusion-ear/cfusion-war/WEB-INF/lib
  - J2EE configuration cf_webapp_root/WEB-INF/lib
- On UNIX, the ColdFusion executable is named cfmx7. It was named cfusion in previous releases.
CHAPTER 3
Tag and Function Changes

This chapter describes tag and function changes, by release, starting with the base ColdFusion MX release.

Contents
Tag changes since ColdFusion 5 .............................................................. 33
Function changes since ColdFusion 5 ...................................................... 51

Tag changes since ColdFusion 5
The following tables list tags, attributes, and values that have changed since ColdFusion 5 and indicate the specific release in which the change was made. Each change might affect code that was written for a previous release. In particular these changes might affect ColdFusion 5 applications. For complete information on CFML tags, see CFML Reference.

• New tags, attributes, values, and changes
• Deprecated tags, attributes, and values
• Obsolete tags, attributes, and values
## New tags, attributes, values, and changes

This table lists tags, attributes, and attribute values that have been added since the ColdFusion MX release:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfapplication</td>
<td>scriptProtect</td>
<td>ColdFusion MX 7</td>
<td>Previously, enabling this attribute caused CFID and CFTOKEN to use domain-wide cookies. Now it also causes client scope with cookies to use domain-wide cookies.</td>
</tr>
<tr>
<td></td>
<td>setDomainCookies</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>loginStorage</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>cfargument</td>
<td>xml value of type</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>cfcache</td>
<td>cacheDirectory, timespan attributes</td>
<td>ColdFusion MX</td>
<td>The action=&quot;cache&quot; attribute caches on both the server and client, and the cache and optimal values are equivalent. A new action value, serverCache, caches pages on the server only. It is functionally identical to the action=&quot;cache&quot; attribute value in ColdFusion 5. The directory and cacheDirectory attributes are now equivalent. You can use either attribute with any action. In ColdFusion 5, you used directory with action=&quot;flush&quot;, and cacheDirectory with all other action values. The default values for port and protocol are now the values that are used to initially access the page. In earlier releases, the default value for port was 80, and the default value for protocol was http.</td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------</td>
<td>---------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>cfcache</td>
<td>action=&quot;flush&quot; attribute</td>
<td>ColdFusion MX</td>
<td>The <code>action=&quot;flush&quot;</code> attribute now correctly expires a cached page in the server, the client, or both locations. The cfcache.map file now contains the template name and source timestamp.</td>
</tr>
<tr>
<td>cfcalendar</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfcatch</td>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>The <code>cfcatch.Message</code> attribute gets the current values every time an exception is thrown, and ColdFusion MX ignores any <code>cfset</code> assignment to a <code>cfcatch</code> member. To remedy this, construct a new message or variable to make extensions to changes in <code>cfcatch</code> values. The <code>cfcatch.Message</code> attribute returns a different SQLState value for Type 4 database drivers; this affects the <code>NativeErrorCode</code> variable as well. You can no longer use <code>cfcatch</code> to catch function validation errors; use the ColdFusion MX compiler instead.</td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------</td>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cfchart</td>
<td>style, title</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xAxisType, yAxisType</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfchartdata</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfchartseries</td>
<td>datalabelStyle</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>column value of type</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfcol</td>
<td>header attribute changed</td>
<td>ColdFusion MX</td>
<td>If you use the cfcol header attribute without any text, ColdFusion MX displays the header as true. ColdFusion 5 displays the header as TRUE.</td>
</tr>
<tr>
<td>cfcollection</td>
<td>categories</td>
<td>ColdFusion MX 7</td>
<td>New values of the language attribute</td>
</tr>
<tr>
<td></td>
<td>categoryList value of action attribute</td>
<td>ColdFusion MX</td>
<td>list value of action attribute</td>
</tr>
<tr>
<td></td>
<td>name attribute</td>
<td>ColdFusion MX</td>
<td>Required when action=list. This attribute provides the name for the query results. When you use cfcollection with action = &quot;list&quot; and name = &quot;&lt;query_results_name&gt;&quot;*, you retrieve a table with the properties for every registered Verity collection.</td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>cfcomponent</td>
<td>style, namespace, serviceportname, porttypename, wsdifile, and bindingname attributes</td>
<td>ColdFusion MX 7</td>
<td>Extended functionality for the hint and displayname attributes when publishing document-literal style web services.</td>
</tr>
<tr>
<td>cfcontent</td>
<td>variable attribute</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfdirectory</td>
<td>recurse attribute for list and delete actions</td>
<td>ColdFusion MX 7</td>
<td>Change (Windows) This tag no longer returns rows for . and .. in the query results, but only subdirectories, like on UNIX. If your application has code to skip these rows (for example, startrow=<em>3</em>), you must change it. (Windows) This tag no longer supports sort=&quot;temporary&quot;, and it only supports sorting on the ReadOnly and Hidden attributes. (UNIX) When action=list, cfdirectory only reports the ReadOnly and Hidden attributes.</td>
</tr>
<tr>
<td>cfdocument</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfdocumentitem</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfdocumentsection</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfdump</td>
<td>top attribute</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cferror</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>The type=&quot;monitor&quot; attribute is obsolete in ColdFusion MX.</td>
</tr>
<tr>
<td>cfexecute</td>
<td>variable attribute</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td><strong>cfile</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>result</strong> attribute for upload action</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>fixnewline</strong> attribute for action=&quot;append&quot; and action=&quot;write&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td>ColdFusion MX</td>
<td>When assigning the attributes, the file’s Archive attribute is always set.</td>
<td></td>
</tr>
<tr>
<td><strong>cform</strong></td>
<td></td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>name</strong> and <strong>action</strong> attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>accessible</strong>, <strong>format</strong>, <strong>height</strong>, <strong>width</strong>, <strong>method</strong>, <strong>onError</strong>, <strong>preloader</strong>, **scripts<code>, **skin</code>, <strong>style</strong>, <strong>timeout</strong>, <strong>wMode</strong> attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>ColdFusion MX only uses the cfinput tag’s checked attribute if the corresponding form field value is undefined. ColdFusion 5 used the checked attribute if the field was undefined, or if it was defined but not equal to any of the specified options. For example, if you use a cfparam tag to set the default value for form.foo to false before the following cfinput tags, neither of the options are checked. In ColdFusion 5, the Yes option is checked. &lt;cfinput name=&quot;foo&quot; type=&quot;radio&quot; value=&quot;yes&quot; checked&gt; &lt;cfinput name=&quot;foo&quot; type=&quot;checkbox&quot; value=&quot;no&quot;&gt;</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------</td>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cfformgroup</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td>The <code>cfformgroup</code> discussion in CFML Reference does not include the <code>id</code> attribute, which you use for accordion, tabnavigator, and repeater <code>cfformgroup</code> types. For more information, see the Release Notes.</td>
</tr>
<tr>
<td>cfformitem</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfftp</td>
<td>result attribute</td>
<td>ColdFusion MX 7</td>
<td>ColdFusion MX still allows the <code>proxyBypass</code> attribute, but it has no affect.</td>
</tr>
<tr>
<td>cffunction</td>
<td>description attribute</td>
<td>xml value for <code>returnType</code> attribute</td>
<td></td>
</tr>
<tr>
<td>cfgrid</td>
<td>format attribute and support for Flash and XML output enabled, <code>onChange</code>, <code>style</code>, <code>tooltip</code>, <code>visible</code> attributes (Flash format only)</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfgridcolumn</td>
<td>mask attribute</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cfhttp</td>
<td><code>result</code> attribute</td>
<td>ColdFusion MX 7</td>
<td>The <code>timeout</code> attribute requires the Java Development Kit (JDK) version 1.4. Otherwise, ColdFusion MX ignores this attribute. The <code>cfhttp</code> tag had many enhancements in ColdFusion MX 6.1. For more information, see CFML Reference.</td>
</tr>
<tr>
<td></td>
<td>Changes</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HEAD, PUT, DELETE, OPTIONS,</td>
<td>ColdFusion MX</td>
<td>The cfhttp response headers in ColdFusion MX differ from those in ColdFusion 5, as follows:</td>
</tr>
<tr>
<td></td>
<td>and TRACE values of method</td>
<td></td>
<td>• Mixed case (returned by the server without changes) in ColdFusion MX; uppercase in ColdFusion 5</td>
</tr>
<tr>
<td></td>
<td>attribute</td>
<td></td>
<td>• Random order in ColdFusion MX; alphabetical order in ColdFusion 5</td>
</tr>
<tr>
<td></td>
<td>multipart, getasbinary,</td>
<td></td>
<td>Status Code: 200 is followed by OK in ColdFusion MX (returned by the server without changes), and by SUCCESS in ColdFusion 5</td>
</tr>
<tr>
<td></td>
<td>proxyUser, proxyPassword</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>charset, firstrowasheaders</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cfftpparam</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>header</code> and <code>body</code> values</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>of type attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>encoded</code>, <code>mimetype</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cfimport</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cfindex</td>
<td>custom3, custom4, category, and categorytree attributes for update and refresh actions</td>
<td>ColdFusion MX 7</td>
<td>New values. For more information, see CFML Reference.</td>
</tr>
<tr>
<td></td>
<td>status attribute for Update, Refresh, Delete, Purge actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>language attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change</td>
<td></td>
<td>ColdFusion MX</td>
<td>You can use the query attribute and type=&quot;file&quot; or type=&quot;path&quot;, for all actions that require information from the key attribute.</td>
</tr>
<tr>
<td>cfinput</td>
<td>height and width attributes (all except checkbox and radio)</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>bind attribute (text and password)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>label attribute (all but radio, button, image, reset, and submit)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>mask attribute (text only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>validateAt attribute (all but radio, button, image, reset, and submit)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>datefield, button, file, hidden, image, reset, and submit values of type attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>daynames and monthnames attributes (type=&quot;datefield&quot; only)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>boolean, email, guid, maxlength, noblanks, range, submitOnce, URL, USdate, and uuid values of the validate attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tooltip, visible, and enabled attributes (for all types but hidden)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------</td>
<td>----------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>cfinput (continued)</td>
<td>src attribute (image only)</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfinvoke</td>
<td>servicePort attribute for web services</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>cfinvokeargument</td>
<td>omit attribute</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>cfldap</td>
<td>returnAsBinary attribute</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changes</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You can no longer sort cfldap query results on the client side using the sort attribute. The sort attribute triggers a server-side sort. If the LDAP server does not support server-side sorting, cfldap throws an error. The sort order depends on the LDAP server; for example, iPlanet Directory Server 5.0, Novell 6.0 server, Oracle Internet Directory 9i, Microsoft Active Directory, and others might each sort differently. To do client-side sorting on the cfldap query results, use the ColdFusion Query of Queries feature.</td>
<td></td>
</tr>
</tbody>
</table>

<p>| action=query and dn attribute | When action=query and dn is in the list of attributes, cfldap returns each Distinguished Name (DN) with a comma followed by a space. (In ColdFusion 5, cfldap did not return DNs in a consistent format.) |         |</p>
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfldap (continued) referral attribute</td>
<td>ColdFusion MX</td>
<td>If the referral attribute is selected, cfldap returns the name of the referred server before the results. For example, if the referred server is server1.rnd.anytown.anycorp.com:123, the results could look like the following: DN: ldap:// server1.rnd.anytown.anycorp.com:123/uid=bparker, ou=People, dc=siroe, dc=com, dc=siroe, dc=co, cn=Barry Parker, ou=Product Development, People</td>
<td></td>
</tr>
<tr>
<td>cflocation</td>
<td>Change</td>
<td>ColdFusion MX 7</td>
<td>The cflocation tag throws an InvalidCharInURLException if the CR/LF characters are in the query string (in ASCII or hexadecimal). Changes</td>
</tr>
<tr>
<td>cflogin</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfloginuser</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cflogout</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfloop</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>ColdFusion MX returns struct keys in the order that you create them. ColdFusion 5 returns struct keys in alphabetical order. To sort struct values or keys, use the StructSort and StructKeySort functions.</td>
</tr>
</tbody>
</table>
**Tag** | **Attribute or value** | **Added in this** | **Comment**
--- | --- | --- | ---
**cfmail** | spoolEnable attribute | ColdFusion MX | The cfmail tag had many enhancements in ColdFusion MX 6.1. For more information, see the ColdFusion documentation.
| charset, failto, replyTo, userName, password, wrapText attributes | ColdFusion MX 6.1 | Change |
| contentID, disposition attributes | ColdFusion MX 7 | |
| type attribute | ColdFusion MX 6.1 | |
**cfmailparam** | | ColdFusion MX 7 | |
**cfmailpart** | All | ColdFusion MX 6.1 | |
**cfNTauthenticate** | All | ColdFusion MX 6.1 | |
**cfobject** | All | ColdFusion MX | |
**cfobjectcache** | All | ColdFusion MX | |
**cfparam** | min, max, pattern attributes | ColdFusion MX 7 | ColdFusion MX correctly disallows the assignment of complex data types (such as array, binary, query, and struct) to the client scope in this tag.
| creditcard, email, eurodate, float, integer, range, regex, regular_expression, ssn, social_security_number, time, URL, USdate, XML, zipcode attributes of the type attribute | |
**cfprocessingdirective** | pageEncoding attribute | ColdFusion MX | |
<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfproccparam</td>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>The <code>maxrows</code> attribute is obsolete in ColdFusion MX. If you have ref cursors in packages or stored procedures, use <code>cfprocresult</code> instead. This causes Datadirect JDBC to place the Oracle ref cursors into a result set. <em>Note:</em> you cannot use this method with the Oracle ThinClient JDBC drivers. ColdFusion MX ignores the <code>dbvarname</code> attribute for all drivers. (ColdFusion 5 used it for Sybase, Oracle, and Informix native drivers.) ColdFusion MX is a JDBC client, which does not support named parameter passing like the Sybase <code>ctlib</code> and Oracle <code>oci</code> libraries.</td>
</tr>
<tr>
<td>cfproperty</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
</tbody>
</table>
### cfquery Changes

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfquery</td>
<td>result attribute</td>
<td>ColdFusion MX 7</td>
<td>When running in a locale other than US or Japanese, <code>cfquery</code> always returns date fields in numeric format; for example, 12:13:52 or 3:30PM.</td>
</tr>
</tbody>
</table>

### cfregistry Changes

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfregistry</td>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>In Windows, you should not use <code>cfregistry</code> to read ColdFusion server settings—such as the mail root and information about scheduled tasks—because much of this information is no longer stored in the registry, or it is stored in a different location in the registry. The ColdFusion-related values that are stored in the registry are subject to change.</td>
</tr>
</tbody>
</table>

### cfreturn

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfreturn</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------------</td>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cfreport</td>
<td>template, format, name, filename, query, overwrite, permissions, encryption, ownerpassword, and userpassword attributes</td>
<td>ColdFusion MX 7</td>
<td>This tag no longer manages the data connection for a Crystal report. The tag passes datasource, username, and password attributes to the Crystal Reports server for it to manage the data connection directly.</td>
</tr>
<tr>
<td>cfreportparam</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfsearch</td>
<td>category, categoryTree, status, suggestions, contextPassages, contextBytes, contextHighlightBegin, contextHighlightEnd, previousCriteria attributes natural, internet, and internet_basic values of type attribute</td>
<td>ColdFusion MX 7</td>
<td>New values for the language attribute</td>
</tr>
<tr>
<td>cfselect</td>
<td>selected attribute can take a list enabled, group, height, label, onKeyUp, onKeyDown, onMouseUp, onMouseDown, onChange, onClick, queryPosition, tooltip, visible, and width attributes</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfsetting</td>
<td>requestTimeOut attribute</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfstoredproc</td>
<td>result attribute</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------</td>
<td>----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>cfswitch</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>You can no longer have code inside a cfswitch tag that is not in a cfcase or cfdefaultcase tag. ColdFusion 5 erroneously ignores the code, whereas ColdFusion MX correctly throws a compiler error.</td>
</tr>
<tr>
<td>cftable</td>
<td>header attribute changed</td>
<td>ColdFusion MX</td>
<td>If you use the cftable header attribute without any text, ColdFusion MX displays the header as true. ColdFusion 5 displays the header as TRUE.</td>
</tr>
<tr>
<td>cftextarea</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cfthrow</td>
<td>object attribute</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cftimer</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>cftrace</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>cfxml</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
</tbody>
</table>
## Deprecated tags, attributes, and values

The following tags, attributes, and attribute values are deprecated. Do not use them in ColdFusion applications. They might not work, and might cause an error, in releases later than ColdFusion MX.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Deprecated as of this ColdFusion release</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfccache</td>
<td>cachedirectory, timeout attributes. Use the directory and timespan attributes instead.</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfcollection</td>
<td>map and repair options of the action attribute</td>
<td>ColdFusion MX 7</td>
</tr>
<tr>
<td>cferror</td>
<td>monitor option of the exception attribute</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cffile</td>
<td>system value for attributes attribute</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td></td>
<td>temporary value for attributes attribute</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfform</td>
<td>passthrough attribute</td>
<td>ColdFusion MX 7</td>
</tr>
<tr>
<td></td>
<td>enableCAB attribute</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfldap</td>
<td>filterFile attribute</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cflog</td>
<td>date, thread, time attributes. The default value of all of these attributes is Yes. If you set any of these attribute values to No, ColdFusion MX throws a runtime exception.</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfquery</td>
<td>connectString, dbName, dbServer, dbType, provider, providerDSN, sql attributes</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfregistry</td>
<td>All</td>
<td>(UNIX) ColdFusion MX</td>
</tr>
<tr>
<td>Tag</td>
<td>Attribute or value</td>
<td>Deprecated as of this ColdFusion release</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>cfsearch</td>
<td>external, language attributes</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfselect</td>
<td>passthrough attribute</td>
<td>ColdFusion MX 7</td>
</tr>
<tr>
<td>cfsvrllet</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfservlletparam</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfslider</td>
<td>img, imgStyle, grooveColor, refreshLabel, tickmarkImages, tickmarklabels, tickmarkmajor, tickmarkminor attributes</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfstoredproc</td>
<td>connectString, dbName, dbServer, dbtype, provider, providerDSN attributes</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cftextinput</td>
<td>All</td>
<td>ColdFusion MX 7</td>
</tr>
<tr>
<td>cfupdate</td>
<td>connectString, dbName, dbServer, dbtype, provider, providerDSN attributes</td>
<td>ColdFusion MX</td>
</tr>
</tbody>
</table>

### Obsolete tags, attributes, and values

The following tags, attributes, and attribute values are obsolete. Do not use them in ColdFusion applications. They do not work, and might cause an error, in releases later than ColdFusion 5.

<table>
<thead>
<tr>
<th>Tag</th>
<th>Attribute or value</th>
<th>Obsolete as of this ColdFusion release</th>
</tr>
</thead>
<tbody>
<tr>
<td>cfapplication</td>
<td>The clientstorage = &quot;registry&quot; setting</td>
<td>(UNIX) ColdFusion MX</td>
</tr>
<tr>
<td>cfauthenticate</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfchart</td>
<td>rotated attribute</td>
<td>ColdFusion MX 7</td>
</tr>
<tr>
<td>cfctfile</td>
<td>attributes attribute value archive</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfctform</td>
<td>enablecab attribute</td>
<td>ColdFusion MX 7</td>
</tr>
<tr>
<td>cfctimpersonate</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfctindex</td>
<td>action attribute value optimize</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfctinternaladminsecurity</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfldap</td>
<td>fileterConfig attribute</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>cfnewinternaladminsecurity</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
</tbody>
</table>
Function changes since ColdFusion 5

The following tables list functions, parameters, and values that have changed since ColdFusion 5 and indicate the specific release in which the change was made. Each change might affect code that was written for a previous release. In particular, these changes might affect ColdFusion 5 applications. For complete information on CFML tags, see CFML Reference.

- New functions, parameters, values, and changes
- Deprecated functions, parameters, and values
- Obsolete functions, parameters, and values

New functions, parameters, values, and changes

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameter, value, or change</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayAvg</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>These functions return 0 when used with an empty array. ColdFusion 5 returns infinity (1.#INF).</td>
</tr>
<tr>
<td>ArrayMin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ArrayMax</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ArraySum</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BinaryDecode</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>BinaryEncode</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>CharSetDecode</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>CharSetEncode</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>CreateObject</td>
<td>portName parameter</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>Date functions</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>The date pivot point for calculating the century in a two-digit year has changed from what it was in ColdFusion 5, which affects the results from date functions. For more information, see “Date pivot point for two-digit years” on page 18.</td>
</tr>
</tbody>
</table>

Tag | Attribute or value | Obsolete as of this ColdFusion release |
---|-------------------|----------------------------------------|
<p>| cfprocparam        | maxrows attribute      | ColdFusion MX                         |
| cfsetting          | catchExceptionsByPattern attribute | ColdFusion MX                         |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Parameter, value, or change</th>
<th>Added in this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>DateAdd</td>
<td>L key of datepart parameter</td>
<td>ColdFusion MX 6.1</td>
<td>In ColdFusion 5, there is one less second in a minute when calculating a negative difference. For example, it returns 60 seconds between 03:44:23 and 03:45:23, but -59 seconds between 03:45:23 and 03:44:23. ColdFusion MX calculates negative differences correctly, so if your application contains code that adjusts the ColdFusion 5 results, you might need to change it.</td>
</tr>
<tr>
<td>DateDiff</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>The w (weekdays) datepart option returns the same value as the ww (weeks) datepart option.</td>
</tr>
<tr>
<td>DatePart</td>
<td>L key of datepart parameter</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>Decrypt</td>
<td>algorithm and encoding parameters</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>DeleteClientVariable</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function returns False when the variable does not exist. ColdFusion 5 ignores the function call.</td>
</tr>
<tr>
<td>Encrypt</td>
<td>algorithm and encoding parameters</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GenerateSecretKey</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetBaseTagList</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function returns prefixed custom tags found in the tag stack in the format x:tag, where x is the prefix that you specified.</td>
</tr>
<tr>
<td>GetGatewayHelper</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetAuthUser</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Parameter, value, or change</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetContextRoot</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetEncoding</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>GetLocale</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>When the locale has not been explicitly set using SetLocale, GetLocale returns the default locale of the client computer's operating system. In ColdFusion 5, GetLocale returns English (US) when the locale has not been explicitly set.</td>
</tr>
<tr>
<td>GetLocaleDisplayName</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetMetaData</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>GetPageContext</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>GetProfileSections</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>GetSOAPRequest</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetSOAPRequestHeader</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetSOAPResponse</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetSOAPResponseHeader</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>GetTempDirectory</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>(Windows) This function returns the temporary directory for the embedded Java application server, instead of the temporary directory for the Windows operating system, as in ColdFusion 5.</td>
</tr>
<tr>
<td>Hash</td>
<td>algorithm and encoding parameters</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>IsArray</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function returns True with a QueryColumn argument. You can change the value of an array element, but you cannot add or remove an element.</td>
</tr>
<tr>
<td>IsObject</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>IsSOAPRequest</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>IsUserInRole</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>IsValid</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Parameter, value, or change</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------</td>
<td>---------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>IsWDDX</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function returns No for arrays and other complex values. ColdFusion 5 returns an error. ColdFusion MX and ColdFusion 5 accept a basic value, such as a string, number, logical value, or date and time value.</td>
</tr>
<tr>
<td>IsXML</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>IsXmlAttribute</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>IsXmlDoc</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>IsXmlElem</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>IsXmlNode</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>IsXmlRoot</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>Len</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>When calculating a length, a string-processing function processes an ASCII 0 (NUL) character and continues to process subsequent characters of the string, if any. In ColdFusion 5, these functions did not process any subsequent characters of the string after processing the ASCII 0 (NUL) character.</td>
</tr>
<tr>
<td>ListSetAt</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function no longer changes the first delimiter in the list to the first delimiter specified in the function.</td>
</tr>
<tr>
<td>ListSort</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>In a descending sort with textnocase, ListSort returns elements in a different order than in ColdFusion 5.</td>
</tr>
<tr>
<td>Function</td>
<td>Parameter, value, or change</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------</td>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LSCurrencyFormat</td>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>This function returns a negative number when passed a negative number. With the local argument, this function returns the currency in the locale’s standard format, such as £ for pounds in the English (UK) locale. With the international argument, this function returns the currency with its international currency code, such as AUD for the Australian dollar.</td>
</tr>
<tr>
<td>LSIsDate</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function supports full date and full datetime as parameters, which include the name of the day.</td>
</tr>
<tr>
<td>LSParseDateTime</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>For English (US) locales, the following date mask format is obsolete in ColdFusion MX: 23:15:16 4 Sept. 1998. Use a Java standard date format instead.</td>
</tr>
<tr>
<td>LSTimeFormat</td>
<td>I key of mask parameter</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>QueryAddColumn</td>
<td>datatype parameter</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>QueryNew</td>
<td>columntypelist parameter</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>Rand</td>
<td>algorithm parameter</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>Randomize</td>
<td>algorithm parameter</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>RandRange</td>
<td>algorithm parameter</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Parameter, value, or change</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REFind</td>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>These functions exclude questionable punctuation, such as +, $, and =, in the <code>[:punct:]</code> class of regular expression functions.</td>
</tr>
<tr>
<td>REReplace</td>
<td></td>
<td></td>
<td>These functions recognize accented letters in the appropriate character classes, such as :upper:, :lower:, :alpha:, and so on.</td>
</tr>
<tr>
<td>REFindNoCase</td>
<td></td>
<td></td>
<td>These functions do not allow the backslash character () in a regular expression, even if it is inside a character class []. You must escape it by prefixing an additional backslash (\). ColdFusion 5 does not require you to escape a backslash in this case.</td>
</tr>
<tr>
<td>REReplaceNoCase</td>
<td></td>
<td></td>
<td>REFind does not return the proper value when running in a Japanese locale and using a double byte-character as the search string.</td>
</tr>
<tr>
<td>Function</td>
<td>Parameter, value, or change</td>
<td>Added in this ColdFusion release</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REReplace</td>
<td>Changes</td>
<td>ColdFusion MX</td>
<td>In replacement strings, the following character pairs now represent case conversion codes: \E, \l, \L, \u, and \U. If your ColdFusion 5 application has a replacement string with one of these character pairs, escape it by prefixing it with a backslash; for example, \u. To include special characters for backrefs (\1), prefix them with an additional backslash; for example, \1. To include a real backref, prefix an additional backslash; for example, \\1. For case conversion codes, additional backslash prefixes alternate between escaping the code and not. For backrefs, they always escape.</td>
</tr>
<tr>
<td>ReleaseComObject</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>SendGatewayMessage</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>SetEncoding</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>StructKeyList</td>
<td>Change</td>
<td>ColdFusion MX</td>
<td>This function returns struct keys in the order they are created, not in alphabetical order like in ColdFusion 5. To sort struct values, you can use the StructSort function.</td>
</tr>
<tr>
<td>TimeFormat</td>
<td>I key of mask parameter</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>ToScript</td>
<td>All</td>
<td>ColdFusion MX 7</td>
<td></td>
</tr>
<tr>
<td>URLEncoder</td>
<td>charset parameter</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>URLEncodedFormat</td>
<td>charset parameter</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>URLSessionFormat</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>Wrap</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>XmlChildPos</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>XmlElemNew</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
</tbody>
</table>
## Deprecated functions, parameters, and values

The following functions, parameters, and values are deprecated. Do not use them in ColdFusion applications. They might not work, and might cause an error, in releases later than ColdFusion MX.

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameter or value</th>
<th>Deprecated as of this ColdFusion release</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetMetricData</td>
<td>cachepops parameter</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>GetK2ServerDocCount</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>GetK2ServerDocCountLimit</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>GetTemplatePath</td>
<td>All</td>
<td>ColdFusion MX</td>
<td></td>
</tr>
<tr>
<td>IsK2ServerABroker</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>IsK2ServerDocCountExceeded</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>IsK2ServerOnLine</td>
<td>All</td>
<td>ColdFusion MX 6.1</td>
<td></td>
</tr>
<tr>
<td>ParameterExists</td>
<td>All</td>
<td>ColdFusion MX</td>
<td>Use the <code>IsDefined</code> function.</td>
</tr>
<tr>
<td>SetLocale</td>
<td>locale = &quot;Spanish (Mexican)&quot; value</td>
<td>ColdFusion MX</td>
<td>Use <code>Spanish (Standard)</code>.</td>
</tr>
</tbody>
</table>
## Obsolete functions, parameters, and values

The following functions, parameters, and values are obsolete. Do not use them in ColdFusion applications. They do not work in releases later than ColdFusion 5.

As of ColdFusion MX 7, you can create user-defined functions that have the following names:

<table>
<thead>
<tr>
<th>Function</th>
<th>Parameter or value</th>
<th>Obsolete as of this ColdFusion release</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthenticatedContext</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>AuthenticatedUser</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>isAuthenticated</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>isAuthorized</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
<tr>
<td>isProtected</td>
<td>All</td>
<td>ColdFusion MX</td>
</tr>
</tbody>
</table>