



macromedia®
COLDFUSION®
MX

Migrating ColdFusion 5 Applications



Trademarks

Afterburner, AppletAce, Attain, Attain Enterprise Learning System, Attain Essentials, Attain Objects for Dreamweaver, Authorware, Authorware Attain, Authorware Interactive Studio, Authorware Star, Authorware Synergy, Backstage, Backstage Designer, Backstage Desktop Studio, Backstage Enterprise Studio, Backstage Internet Studio, ColdFusion, Design in Motion, Director, Director Multimedia Studio, Doc Around the Clock, Dreamweaver, Dreamweaver Attain, Drumbeat, Drumbeat 2000, Extreme 3D, Fireworks, Flash, Fontographer, FreeHand, FreeHand Graphics Studio, Generator, Generator Developer's Studio, Generator Dynamic Graphics Server, JRun, Knowledge Objects, Knowledge Stream, Knowledge Track, Lingo, Live Effects, Macromedia, Macromedia M Logo & Design, Macromedia Flash, Macromedia Xres, Macromind, Macromind Action, MAGIC, Mediamaker, Object Authoring, Power Applets, Priority Access, Roundtrip HTML, Scriptlets, SoundEdit, ShockRave, Shockmachine, Shockwave, Shockwave Remote, Shockwave Internet Studio, Showcase, Tools to Power Your Ideas, Universal Media, Virtuoso, Web Design 101, Whirlwind and Xtra are trademarks of Macromedia, Inc. and may be registered in the United States or in other jurisdictions including internationally. Other product names, logos, designs, titles, words or phrases mentioned within this publication may be trademarks, servicemarks, or tradenames of Macromedia, Inc. or other entities and may be registered in certain jurisdictions including internationally.

This product includes code licensed from RSA Data Security.

This guide contains links to third-party Web sites that are not under the control of Macromedia, and Macromedia is not responsible for the content on any linked site. If you access a third-party Web site mentioned in this guide, then you do so at your own risk. Macromedia provides these links only as a convenience, and the inclusion of the link does not imply that Macromedia endorses or accepts any responsibility for the content on those third-party sites.

Apple Disclaimer

APPLE COMPUTER, INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE ENCLOSED COMPUTER SOFTWARE PACKAGE, ITS MERCHANTABILITY OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. THE EXCLUSION OF IMPLIED WARRANTIES IS NOT PERMITTED BY SOME STATES. THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. THIS WARRANTY PROVIDES YOU WITH SPECIFIC LEGAL RIGHTS. THERE MAY BE OTHER RIGHTS THAT YOU MAY HAVE WHICH VARY FROM STATE TO STATE.

Copyright © 1999–2002 Macromedia, Inc. All rights reserved. This manual may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Macromedia, Inc. Part Number ZCF60M200

Acknowledgments

Project Management: Stephen M. Gilson

Writing: David Golden and Kristi Jabbour

Editing: Linda Adler and Noreen Maher

First Edition: May 2002

Macromedia, Inc.
600 Townsend St.
San Francisco, CA 94103

CONTENTS

ABOUT THIS BOOK V

Developer resources	vi
About Macromedia ColdFusion MX documentation	vii
Printed and online documentation set	vii
Viewing online documentation	viii
Getting answers	viii
Contacting Macromedia	viii

MIGRATING COLDFUSION 5 APPLICATIONS 1

Migrating to ColdFusion MX	2
Planning your migration	3
Using the ColdFusion Code Compatibility Analyzer	4
Using the settings migration wizard	5
Limitations of the wizard	5
Migrating settings later in Windows	6
Migrating settings later in UNIX	6
CFML tags and attributes	7
CFML functions and variables	13
Database operations	16
Connecting to a data source with ODBC Socket	16
Session management	16
QueryColumn object	17
Data type of query results	17
Appearance of query results	17
Localization	18
Extended character code support	18
Support for new currency formats	18
Changes to formatting functions	19
Other locale-specific changes	23
Other changes	24
Character encoding	24
Advanced Security	24
SNMP support	24
Variables	24
Operators	25
CFML data types	25
COM objects	25
Template handlers	25

Case in forms	25
Objects for date/time	26
Date pivot point for two-digit years	26
Request variable	26
Definition of pi	26
Null values	26
thisTag scope	27
Caching	27
Log files	27
Tag and function examples	28
Using cfregistry in ColdFusion MX	28
Using ListSort() in ColdFusion MX	30
Using the cfindex tag in ColdFusion MX	31

INDEX 33

ABOUT THIS BOOK

Migrating ColdFusion 5 Applications provides migration and incompatibility information for anyone who is migrating applications from ColdFusion Server 5 to ColdFusion MX. For information about migrating from earlier versions of ColdFusion, see the Support section of the Macromedia website (<http://www.macromedia.com/support>).

Contents

- [Developer resources](#) vi
- [About Macromedia ColdFusion MX documentation](#)..... vii
- [Getting answers](#) viii
- [Contacting Macromedia](#) viii

Developer resources

Macromedia, Inc. is committed to setting the standard for customer support in developer education, documentation, technical support, and professional services. The Macromedia website is designed to give you quick access to the entire range of online resources. The following table shows the locations of these resources:

Resource	Description	URL
Macromedia website	General information about Macromedia products and services	http://www.macromedia.com
Information on ColdFusion	Detailed product information on ColdFusion and related topics	http://www.macromedia.com/coldfusion
Macromedia ColdFusion Support Center	Professional support programs that Macromedia offers	http://www.macromedia.com/support/coldfusion
ColdFusion Online Forums	Access to experienced ColdFusion developers through participation in the Online Forums, where you can post messages and read replies on many subjects relating to ColdFusion	http://webforums.macromedia.com/coldfusion/
Installation Support	Support for installation-related issues for all Macromedia products	http://www.macromedia.com/support/email/isupport
Training	Information about classes, on-site training, and online courses offered by Macromedia	http://www.macromedia.com/support/training
Developer Resources	All the resources that you need to stay on the cutting edge of ColdFusion development, including online discussion groups, Knowledge Base, technical papers, and more	http://www.macromedia.com/desdev/developer/
Reference Desk	Development tips, articles, documentation, and white papers	http://www.macromedia.com/v1/developer/TechnologyReference/index.cfm
Macromedia Alliance	Connection with the growing network of solution providers, application developers, resellers, and hosting services creating solutions with ColdFusion	http://www.macromedia.com/partners/

About Macromedia ColdFusion MX documentation

The ColdFusion documentation is designed to provide support for the complete spectrum of participants. The print and online versions are organized to let you quickly locate the information that you need. The ColdFusion online documentation is provided in HTML and Adobe Acrobat formats.

Printed and online documentation set

The ColdFusion documentation set consists of the following titles:

Book	Description
<i>Installing ColdFusion MX</i>	Describes system installation and basic configuration for Windows NT, Windows 2000, Solaris, Linux, and HP-UX.
<i>Administering ColdFusion MX</i>	Describes how to use the ColdFusion Administrator to manage the ColdFusion environment, including connecting to your data sources and configuring security for your applications.
<i>Developing ColdFusion MX Applications with CFML</i>	Describes how to develop your dynamic web applications, including retrieving and updating your data, using structures, and forms.
<i>Getting Started Building ColdFusion MX Applications</i>	Contains an overview of ColdFusion features and application development procedures. Includes a tutorial that guides you through the process of developing an example ColdFusion application.
<i>Using Server-Side ActionScript in ColdFusion MX</i>	Describes how Macromedia Flash movies executing on a client browser can call ActionScript code running on the ColdFusion server. Includes examples of server-side ActionScript and a syntax guide for developing ActionScript pages on the server.
<i>Migrating ColdFusion 5 Applications</i>	Describes how to migrate a ColdFusion 5 application to ColdFusion MX. This book describes the code compatibility analyzer that evaluates your ColdFusion 5 code to determine any incompatibilities within it.
<i>CFML Reference</i>	Provides descriptions, syntax, usage, and code examples for all ColdFusion tags, functions, and variables.
<i>CFML Quick Reference</i>	A brief guide that shows the syntax of ColdFusion tags, functions, and variables.
<i>Working with Verity Tools</i>	Describes Verity search tools and utilities that you can use for configuring the Verity K2 Server search engine, as well as creating, managing, and troubleshooting Verity collections.
<i>Using ClusterCATS</i>	Describes how to use Macromedia ClusterCATS, the clustering technology that provides load-balancing and failover services to assure high availability for your web servers.

Viewing online documentation

All ColdFusion documentation is available online in HTML and Adobe Acrobat Portable Document Format (PDF) files. To view the HTML documentation, open the following URL on the web server running ColdFusion: http://web_root/cfdocs/dochome.htm.

ColdFusion documentation in Acrobat format is available on the ColdFusion product CD-ROM.

Getting answers

One of the best ways to solve particular programming problems is to tap into the vast expertise of the ColdFusion developer communities on the ColdFusion Forums. Other developers on the forum can help you figure out how to do just about anything with ColdFusion. The search facility can also help you search messages from the previous 12 months, allowing you to learn how others have solved a problem that you might be facing. The Forums is a great resource for learning ColdFusion, but it is also a great place to see the ColdFusion developer community in action.

Contacting Macromedia

Corporate
headquarters

Macromedia, Inc.
600 Townsend Street
San Francisco, CA 94103
Tel: 415.252.2000
Fax: 415.626.0554
Web: <http://www.macromedia.com>

Technical support

Macromedia offers a range of telephone and web-based support options. Go to <http://www.macromedia.com/support/coldfusion> for a complete description of technical support services.

You can make postings to the ColdFusion Support Forum (<http://webforums.macromedia.com/coldfusion>) at any time.

Sales

Toll Free: 888.939.2545
Tel: 617.219.2100
Fax: 617.219.2101
E-mail: sales@macromedia.com
Web: <http://www.macromedia.com/store>

MIGRATING COLDFUSION 5 APPLICATIONS

This chapter describes the known compatibility issues between ColdFusion Server 5 and ColdFusion MX. For late-breaking known compatibility issues, see the “Migrating ColdFusion 5 Applications” section of the Release Notes on the Macromedia website.

- [Migrating to ColdFusion MX](#)..... 2
- [Planning your migration](#)..... 3
- [Using the ColdFusion Code Compatibility Analyzer](#) 4
- [Using the settings migration wizard](#) 5
- [CFML tags and attributes](#)..... 7
- [CFML functions and variables](#)..... 13
- [Database operations](#)..... 16
- [Localization](#) 18
- [Other changes](#)..... 24
- [Tag and function examples](#) 28

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 3.

Whether you encounter incompatibilities depends on the features of ColdFusion and the CFML elements that your existing applications use. This book lists all of the known incompatibilities that you might encounter, 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 4.

For information about migrating from earlier versions of ColdFusion, see the Support section of the Macromedia website (<http://www.macromedia.com/support>).

Planning your migration

To facilitate the migration process, Macromedia recommends that you develop a structured migration strategy. The migration steps offered here are provided as examples. Because every application is different, you must tailor your own migration strategy to meet your environment's needs.

Adapt the following steps to fit your needs:

- 1 Study the migration section of the release notes and this book to anticipate the compatibility issues that might affect your system.
- 2 Archive and deploy your existing ColdFusion 5 application to a test server.
- 3 Migrate your production databases to the test server. If you have not already done so, migrate the data sources in your ColdFusion 5 application to ColdFusion MX.
- 4 Ensure that the application is functioning properly on the test server before upgrading it to ColdFusion MX. For detailed server specifications, contact your website administrator or hosting provider.
- 5 Install ColdFusion MX on the test server.
- 6 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 4](#).
- 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 Using a source control system, make the necessary changes to your ColdFusion application and run the application on your test server. Ensure that you test all application features under load, not just with a few users.

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 (<http://www.macromedia.com/support>).

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 for potential incompatibilities. Based on diagnostic information, the Code Compatibility Analyzer issues error and information messages to alert you to possible compatibility issues.

The Code Compatibility Analyzer detects the following 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 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.
- When testing only for possible ColdFusion MX incompatibilities, turn off CFML validation in the Code Compatibility Analyzer administration page.
- To validate strict CFML syntax, turn on CFML validation. You can also validate individual tags and functions. For more information, see the online Help in the Code Compatibility Analyzer administration page.

Using the settings migration wizard

If you have ColdFusion 5 installed on the same computer as ColdFusion MX, you are prompted to migrate your ColdFusion settings into ColdFusion MX. For Windows, you are prompted immediately after you finish the installation; for UNIX, you are prompted the first time that you open the ColdFusion MX Administrator.

If you choose to skip the migration, you can do it later (see the instructions in [“Migrating settings later in Windows” on page 6](#) or [“Migrating settings later in UNIX” on page 6](#)). However, running the wizard later might overwrite new ColdFusion MX settings, because the wizard copies settings from the ColdFusion 5 configuration.

Limitations of the wizard

The following table lists the known limitations of the settings migration wizard:

Limitation	Remedy
Does not migrate CORBA connector settings.	Manually reconfigure them in the ColdFusion Administrator.
Does not migrate DB2 data sources (ODBC and Native).	Manually recreate these data sources after installing ColdFusion MX.
Might incorrectly set the Port value in the JDBC Type IV Informix driver that it creates for an ODBC data source that uses the Native Informix driver.	Edit the data source in the ColdFusion Administrator to include the correct port value.
Might incorrectly set the Port value in the JDBC Type IV Informix driver that it creates for an ODBC data source that uses the Sybase ASE driver or Native driver.	
The BLOB and CLOB options are not selected for a data source by default, even if a data source supports BLOBs and CLOBs (for example, Oracle BLOBs and CLOBs, SQL Server text columns, or Microsoft Access memo columns).	Manually select the appropriate Long Text Retrieval option in the ColdFusion Administrator advanced settings for Data Source Names (DSNs).
If your ColdFusion 5 log directory setting is a value other than the default (cfusion\log), the wizard migrates it, but ColdFusion MX uses the default value until you restart ColdFusion MX. (For this reason, the migration log always resides in cfusion\log, no matter where your other log files are located.)	Restart ColdFusion MX after completing the migration.

Migrating settings later in Windows

To run the settings migration wizard later, you must configure the software to prompt you again to perform the migration.

To migrate settings later in Windows:

- 1 Close the ColdFusion Administrator if it is open.
- 2 Change the value of the following registry key from 0 to 1:
HKEY_LOCAL_MACHINE\SOFTWARE\Macromedia\Install Data\Neo\migrate
- 3 Restart the ColdFusion Administrator.
You are prompted again to perform the migration.
- 4 Click Yes, and follow the on-screen instructions.

Migrating settings later in UNIX

To run the settings migration wizard later, you must configure the software to prompt you again to perform the migration.

To migrate settings later in UNIX:

- 1 Stop the ColdFusion Administrator if it is running.
- 2 Create a file called **upgrade** (it can be blank) in your *cf_root/bin* directory.
Note: The user account under which ColdFusion MX runs must have read/write access to this file. The file is deleted after the migration completes.
- 3 Start the ColdFusion Administrator.
You are prompted again to perform the migration.
- 4 Click Yes, and follow the on-screen instructions.

CFML tags and attributes

The following table describes the tags and attributes in ColdFusion MX that differ from ColdFusion Server 5:

Tag	Status	Compatibility issue	For more information
cfapplet	Changed	Can now be used outside the context of <code>cfform</code> .	See <code>cfapplet</code> in <i>CFML Reference</i> .
cfapplication	Changed	Setting <code>clientstorage = "registry"</code> is obsolete in UNIX versions of ColdFusion MX.	See <code>cfapplication</code> in <i>CFML Reference</i> .
cfauthenticate, cfimpersonate cfinternaladmin security cfnewinternal adminsecurity	Obsolete	ColdFusion MX is based on a completely different security model than ColdFusion 5.	See the chapter on application Security in <i>Developing ColdFusion MX Applications with CFML</i> .
cfcache	Changed	<p>The <code>timeout</code> attribute is deprecated in ColdFusion MX; it will be obsolete in later versions. Use <code>timespan</code> instead. The new <code>timespan</code> attribute specifies a decimal number of days between when the page containing the <code>cfcache</code> tag is first cached, and when the cache contents are automatically flushed.</p> <hr/> <p><code>action="cache"</code> now caches on both the server and client, and the <code>cache</code> and <code>optimal</code> values are equivalent. A new <code>action</code> value, <code>serverCache</code>, caches pages on the server only. It is identical in function to the <code>cache</code> value in ColdFusion 5.</p> <hr/> <p>The <code>directory</code> and <code>cacheDirectory</code> attributes are now equivalent. You can use either attribute with any action. (In ColdFusion 5, you can only use <code>directory</code> with <code>action="flush"</code>, and you use <code>cacheDirectory</code> with all other action values.)</p> <hr/> <p>The <code>port</code> and <code>protocol</code> values now default to the values used to initially access the page. In earlier releases, the <code>port</code> default was "80", and the <code>protocol</code> default was <code>http</code>.</p> <hr/> <p><code>action="flush"</code> now correctly expires a cached page in the server, the client, or both locations.</p> <hr/> <p>The <code>cfcache.map</code> file now contains the template name and source timestamp.</p>	See <code>cfcache</code> in <i>CFML Reference</i> .

Tag	Status	Compatibility issue	For more information
cfcatch	Changed	<p>cfcatch.Message gets the current values every time an exception is thrown, and ColdFusion MX ignores any cfset assignment to a cfcatch member.</p> <p>To remedy this, construct a new message or variable to make extensions to changes in cfcatch values.</p> <hr/> <p>cfcatch.Message returns a different SQLState value for Type 4 database drivers; this affects the NativeErrorCode variable as well.</p> <hr/> <p>Can no longer use cfcatch to catch function validation errors; use the ColdFusion MX compiler instead.</p>	See the section about CFCATCH variables in <i>CFML Reference</i> .
cfcol, cftable	Changed	If you use the cfcol header attribute without any text, and the cftable colheader attribute, then ColdFusion MX displays the header as true. ColdFusion 5 displays the header as TRUE.	See cfcol and cftable in <i>CFML Reference</i> .
cfcollection	Changed	<p>The action attribute is now optional, and its default value is list, a new value in ColdFusion MX.</p> <hr/> <p>Added an attribute called name, which is required when action = "list". The name attribute provides the name for the query results.</p> <p>When you use cfcollection with action = "list" and name = "<query_results_name>", you retrieve a table with the properties for every Verity collection that is registered by ColdFusion or the K2 Server.</p>	See cfcollection in <i>CFML Reference</i> , and <i>Working with Verity Tools</i> .
cfdirectory	Changed	<p>(Windows) No longer returns rows for '.' and '..' in the query results, but only subdirectories, like on UNIX.</p> <p>If your application has code to skip these rows (for example, startrow="3"), you must change it.</p> <hr/> <p>(Windows) No longer supports sort="temporary", and it only supports sorting on the ReadOnly and Hidden attributes.</p> <hr/> <p>(UNIX) When action=list, cfdirectory only reports the ReadOnly and Hidden attributes.</p>	See cfdirectory in <i>CFML Reference</i> .
cferror	Changed	type="monitor" is obsolete in ColdFusion MX.	See cferror in <i>CFML Reference</i> .

Tag	Status	Compatibility issue	For more information
cfform	Changed	<p>ColdFusion MX only uses the <code>cfinput</code> tag's <code>checked</code> attribute if the corresponding form field value is undefined. ColdFusion 5 used the <code>checked</code> attribute if the field was undefined, or if it was defined but not equal to any of the specified options.</p> <p>For example, if you use a <code>cfparam</code> tag to default <code>form.foo</code> to "false" before the following <code>cfinput</code> tags, then neither of the options are checked. In ColdFusion 5, the "yes" option is checked.</p> <pre><cfinput name="foo" type="radio" value="yes" checked> <cfinput name="foo" type="checkbox" value="no"></pre> <hr/> <p>The <code>enablecab</code> attribute is deprecated in ColdFusion MX; it will be obsolete in later versions.</p>	See <code>cfform</code> in <i>CFML Reference</i> .
cfgraph, cfgraphdata	Deprecated	Will be obsolete in later versions of ColdFusion MX. These tags render differently than in ColdFusion 5. Use the new graphing tags instead: <code>cfchart</code> , <code>cfchartdata</code> , and <code>cfchartseries</code> .	See the chapter on charting and graphing data in <i>Developing ColdFusion MX Applications with CFML</i> .
cfhttp	Changed	<p>The <code>cfhttp</code> response headers in ColdFusion MX differ from those in ColdFusion 5, as follows:</p> <ul style="list-style-type: none"> • Mixed Case (returned by the server without changes) in ColdFusion MX; uppercase in ColdFusion 5 • Random order in ColdFusion MX; alphabetical order in ColdFusion 5 • "Status Code: 200" is followed by "OK" in ColdFusion MX (returned by the server without changes), and by "SUCCESS" in ColdFusion 5 <hr/> <p>The <code>timeout</code> attribute requires the Java Development Kit (JDK) version 1.4; otherwise, ColdFusion MX ignores this attribute.</p>	See <code>cfhttp</code> in <i>CFML Reference</i> .
cfindex	Changed	You can use the <code>cfindex</code> tag with the <code>query</code> attribute and <code>type="file"</code> or <code>type="path"</code> , for all actions that require information from the <code>key</code> attribute.	See <code>cfindex</code> in <i>CFML Reference</i> and "Using the cfindex tag in ColdFusion MX" on page 31 .

Tag	Status	Compatibility issue	For more information
cfldap	Changed	<p>You can no longer sort <code>cfldap</code> query results on the client side using the <code>sort</code> attribute. The <code>sort</code> attribute triggers a server-side sort. If the LDAP server does not support server-side sorting, <code>cfldap</code> throws an error.</p> <p>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.</p> <p>To do client-side sorting on the <code>cfldap</code> query results, use the ColdFusion Query of Queries feature.</p> <hr/> <p>When <code>action=query</code> and <code>dn</code> is in the list of attributes, <code>cfldap</code> returns each Distinguished Name (DN) with a comma followed by a space. (In ColdFusion 5, <code>cfldap</code> did not return DNs in a consistent format.)</p> <hr/> <p>If you select the <code>referral</code> attribute, <code>cfldap</code> returns the name of the referred server before the results.</p> <p>For example, if the referred server is <code>server1.rnd.anytown.anycorp.com:123</code>, the results could look as follows: <code>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</code></p>	See <code>cfldap</code> in <i>CFML Reference</i> .
cflog	Changed	<p>The <code>thread</code>, <code>date</code>, and <code>time</code> attributes are deprecated, and will be obsolete in later versions of ColdFusion MX. All of these attributes default to Yes. If you set any of these attribute values to No, ColdFusion MX throws a runtime exception.</p>	See <code>cflog</code> in <i>CFML Reference</i> .
cfloop	Changed	<p>ColdFusion MX returns struct keys in the order that you create them. ColdFusion 5 returns struct keys in alphabetical order.</p> <p>If you need sorted struct values or keys, use the <code>StructSort()</code> and <code>StructKeySort()</code> functions.</p>	See <code>cfloop</code> in <i>CFML Reference</i> .
cfmail	Changed	<p>ColdFusion MX formats undeliverable mail differently than in ColdFusion 5. If you have any custom code that relies on this format, you must change it.</p>	See <code>cfmail</code> in <i>CFML Reference</i> .
cfparam	Changed	<p>ColdFusion MX correctly disallows the assignment of complex data types (such as array, binary, query, and struct) to the client scope in this tag.</p>	See <code>cfparam</code> in <i>CFML Reference</i> .

Tag	Status	Compatibility issue	For more information
cfproccparam	Changed	<p>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. (Note: you cannot use this method with the Oracle ThinClient JDBC drivers.)</p> <hr/> <p>The <code>cfproccparam</code> tag 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.</p>	See <code>cfproccparam</code> and <code>cfprocresult</code> in <i>CFML Reference</i> .
cfquery	Changed	<p>The <code>cfquery</code> tag disallows SQL reserved words as variable or column names in a Query of Queries, unless they are escaped. To escape a reserved word, enclose it in brackets; for example: <code>select [COUNT] from myTable</code> or <code>select [FROM] from popQuery</code></p> <hr/> <p>Measures the <code>timeout</code> value in seconds, instead of milliseconds as in ColdFusion 5. You must convert each <code>timeout</code> value from milliseconds to seconds.</p>	See <code>cfquery</code> in <i>CFML Reference</i> .
cfquery, cfinsert, cfgridupdate, cfupdate, cfstoredproc	Changed	<p>In ColdFusion MX, all values of the <code>dbtype</code> attribute are deprecated except for <code>dbtype="query"</code>, and <code>dbtype="dynamic"</code> is obsolete.</p> <p>This is because ColdFusion MX uses the data source services provided by the underlying J2EE server, which provides better scaling and connection support in most cases. However, it only supports passing in the username and password, which is insufficient to define a data source. ColdFusion 5 can do this because it uses Open Database Connectivity (ODBC) to establish a database connection.</p> <p>To work around this, define a data source to connect to the database.</p> <hr/> <p>The <code>dbserver</code>, <code>dbname</code>, <code>connectstring</code>, <code>provider</code>, and <code>providerdsn</code> attributes are obsolete in ColdFusion MX.</p>	See <code>cfquery</code> , <code>cfinsert</code> , <code>cfgridupdate</code> , <code>cfupdate</code> , and <code>cfstoredproc</code> in <i>CFML Reference</i> , and the "Data Source Management" chapter in <i>Administering ColdFusion MX</i> .
cfregistry	Deprecated for UNIX Changed	<p>Registry operations will be obsolete in later UNIX versions of ColdFusion MX.</p> <hr/> <p>In Windows, you should not use <code>cfregistry</code> to read ColdFusion server settings—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 stored in a different location in the registry. The ColdFusion-related values that are stored in the registry are subject to change.</p>	See "Using <code>cfregistry</code> in ColdFusion MX" on page 28.

Tag	Status	Compatibility issue	For more information
cfreport	Changed	This tag no longer manages the data connection for a Crystal report. Passes <code>datasource</code> , <code>username</code> , and <code>password</code> attributes to the Crystal Reports server for it to manage the data connection directly.	See <code>cfreport</code> in <i>CFML Reference</i> .
cfervlet	Deprecated	Will be obsolete in later versions of ColdFusion MX. In <code>cfervlet</code> tags, you can only call servlets that were created in JRun version 3.1 or earlier.	See <code>cfervlet</code> in <i>CFML Reference</i> .
cfervletparam	Deprecated	Will be obsolete in later versions of ColdFusion MX.	See <code>cfervletparam</code> in <i>CFML Reference</i> .
cfsetting	Changed	The <code>catchExceptionsByPattern</code> attribute will be obsolete in later versions of ColdFusion MX. ColdFusion MX uses best-fit exception handling. You can no longer set <code>catchExceptionsByPattern</code> to <code>No</code> to use first-fit exception handling.	See <code>cfsetting</code> in <i>CFML Reference</i> .
cfswitch	Changed	You can no longer have code inside a <code>cfswitch</code> tag that is not in a <code>cfcase</code> or <code>cfdefaultcase</code> tag. ColdFusion 5 erroneously ignores the code, whereas ColdFusion MX correctly throws a compiler error.	See <code>cfswitch</code> in <i>CFML Reference</i> .

The ColdFusion section of the Macromedia website contains the most current compatibility information. For more information, go to <http://www.macromedia.com/coldfusion>.

CFML functions and variables

The following table describes other CFML functions and variables in ColdFusion MX that differ from ColdFusion Server 5:

Function	Status	Issue	For more information
ArrayAvg(), ArrayMin(), ArrayMax(), ArraySum()	Changed	Returns "0" when used with an empty array. ColdFusion 5 returns infinity (1.#INF).	See ArrayAvg(), ArrayMin(), ArrayMax(), and ArraySum() in <i>CFML Reference</i> .
AuthenticatedContext(), AuthenticatedUser(), IsAuthenticated(), IsAuthorized(), IsProtected()	Obsolete	ColdFusion MX is based on a completely different security model than ColdFusion 5.	See the "Application Security" chapter in <i>Developing ColdFusion MX Applications with CFML</i> .
DateDiff()	Changed	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 have to change it.	See DateDiff() in <i>CFML Reference</i> .
DeleteClientVariable()	Changed	Returns False when the variable does not exist. ColdFusion 5 ignores the function call.	See DeleteClientVariable() in <i>CFML Reference</i> .
GetBaseTagList()	Changed	Returns prefixed custom tags found in the tag stack in the format x:tag, where x is the prefix that you specified.	See GetBaseTagList() in <i>CFML Reference</i> .
GetLocale()	Changed	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.	See GetLocale() and SetLocale() in <i>CFML Reference</i> .
GetTempDirectory()	Changed	(Windows) Returns the temporary directory for the embedded Java application server, instead of the temporary directory for the Windows operating system, as in ColdFusion 5.	See GetTempDirectory() in <i>CFML Reference</i> .
IsArray()	Changed	Returns True with a QueryColumn argument. You can change the value of an array element, but you cannot add or remove an element.	See "QueryColumn object" on page 17, and IsArray() in <i>CFML Reference</i> .

Function	Status	Issue	For more information
IsWDDX()	Changed	Returns No for arrays and other complex values. ColdFusion 5 returns an error. (Both versions accept a basic value, such as a string, number, logical value, or date and time value.)	See IsWDDX() in <i>CFML Reference</i> .
Len()	Changed	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.	See Len() in <i>CFML Reference</i> .
ListSetAt()	Changed	No longer changes the first delimiter in the list to the first delimiter specified in the function.	See ListSetAt() in <i>CFML Reference</i> .
ListSort()	Changed	In a descending sort with <code>textnocase</code> , <code>ListSort()</code> returns elements in a different order than in ColdFusion 5.	See "Using ListSort() in ColdFusion MX" on page 30 .
LSCurrencyFormat()	Changed	Returns a negative number when passed a negative number. With the <code>local</code> argument, returns the currency in the locale's standard format, such as £ for pounds in the English(UK) locale. With the <code>international</code> argument, returns the currency with its international currency code, such as AUD for the Australian dollar.	See "Localization" on page 18 .
LSIsDate()	Changed	Supports <code>full date</code> and <code>full datetime</code> as parameters, which include the name of the day.	
LSParseDateTime()	Changed	For English (US) locales, the following date mask format is obsolete in ColdFusion MX: <code>23:15:16 4 Sept. 1998</code> . Use a Java standard date format instead.	

Function	Status	Issue	For more information
REFind REReplace() REFindNoCase() REReplaceNoCase()	Changed	<p>Excludes questionable punctuation, such as +, \$, and =, in the <code>[[:punct:]]</code> class of regular expression functions.</p> <hr/> <p>Recognizes accented letters in the appropriate character classes, such as <code>:upper:</code>, <code>:lower:</code>, <code>:alpha:</code>, and so on.</p> <hr/> <p>Does not allow the backslash character (<code>\</code>) in a regular expression, even if it is inside a character class <code>[]</code>. You must escape it by prefixing an additional backslash (<code>\\</code>). (ColdFusion 5 does not require you to escape a backslash in this case.)</p>	See <code>REFind()</code> , <code>REReplace()</code> , <code>REFindNoCase()</code> , and <code>REReplaceNoCase()</code> in <i>CFML Reference</i> .
REReplace(), REReplaceNoCase()	Changed	<p>In replacement strings, the following character pairs now represent case conversion codes: <code>\E</code>, <code>\I</code>, <code>\L</code>, <code>\u</code>, and <code>\U</code>. 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, <code>\\u</code>).</p> <p>To include special characters for backrefs (<code>{1}</code>), prefix them with an additional backslash (for example, <code>\\1</code>). To include a real backref, prefix an additional backslash (for example, <code>\\1</code>).</p> <p>For case conversion codes, additional backslash prefixes alternate between escaping the code and not. For backrefs, they always escape.</p>	See <code>REReplace()</code> and <code>REReplaceNoCase()</code> in <i>CFML Reference</i> .
StructKeyList	Changed	<p>Returns struct keys in the order they are created, not in alphabetical order like in ColdFusion 5.</p> <p>If you need sorted struct values or keys, you can use the <code>StructSort</code> and <code>StructKeySort</code> functions.</p>	See <code>StructKeyList()</code> in <i>CFML Reference</i> .

Note: The date pivot point for calculating the century in a two-digit year has changed from ColdFusion 5, affecting the results from date functions. For more information, see [“Date pivot point for two-digit years” on page 26](#).

The ColdFusion section of the Macromedia website contains the most current compatibility information. For more information, go to <http://www.macromedia.com/coldfusion>.

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, you can use the bundled Merant Type 4 database drivers or the SQL Link Type 3 database drivers. ColdFusion MX no longer uses the JDBC-ODBC bridge driver.

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 “[CFML tags and attributes](#)” on page 7 and “[CFML functions and variables](#)” on page 13 (for example, `cfquery`, `cfcatch`, and `cfreport`).

For more information about the Merant JDBC drivers included with ColdFusion MX, see <http://www.merant.com>.

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 show up in the list of data sources in the ColdFusion Administrator. It does not in ColdFusion MX; you must manually add the ODBC DSN entry.

To manually add the ODBC DSN entry:

- 1 In the ColdFusion 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 about data source management, see *Administering ColdFusion MX*.

Session management

ColdFusion MX now uses J2EE sessions as its default session management technique. For more information about using persistent data and locking, see *Developing ColdFusion MX Applications with CFML*.

QueryColumn object

The QueryColumn object is even more like an array than in ColdFusion 5, as shown by the following examples:

- 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']>`
- The following code returns True in ColdFusion MX, and False in ColdFusion 5:
`#isArray(query1['firstName'])#`
- The following code works in ColdFusion MX, and produces an error in ColdFusion 5:
`<cfset joeArray[1] = query1['firstName']`

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 may 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.

Appearance of query results

Certain database and operating system combinations can lead to query data being returned with space characters appended to the end. If you encounter this whitespace padding, you can remove it using the SQL command `rtrim()`, if your database supports the SQL `rtrim` function. For example, if `SELECT ColA, ColB FROM TableA` returns results for `ColB` with whitespace padding, you can rewrite the query as follows:

```
SELECT ColA, rtrim(ColB) AS ColB FROM TableA
```

Localization

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

- [Extended character code support](#)
- [Changes to formatting functions](#)
- [Other locale-specific changes](#)

Note: Results from localization functions might differ, depending on your Java Runtime Environment (JRE). The contents of this section reflect tests on the SUN JRE version 1.4 and the IBM JRE version 1.3.0. Macromedia has also tested using the SUN JRE version 1.3.1_01; however, many of these tests failed on locales other than English(US).

Extended character code support

ColdFusion MX supports character code values 0–65535, whereas ColdFusion 5 supports 1-255. This affects string-processing functions. For more information, see `Len()` in [“CFML functions and variables” on page 13](#).

Support for new currency formats

ColdFusion MX recognizes new currency formats for new locales. For example, 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. Also, `LSIsCurrency()` now returns “yes” for the locales and currency formats listed in the following table:

Locale	Newly supported currency formats
English(Australian)	AUD123,456.78
English(Canadian)	CAD123,456.78
English(New Zealand)	NZD123,456.78
English(UK)	£123,456.78 GBP123,456.78
French(Belgian)	123.456,78 123.456,78 FB BEF123.456,78
French(Canadian)	123 456,78 123 456,78 \$ CAD123 456,78
French(Standard)	123 456,78 123 456,78 F FRF123 456,78
French(Swiss)	123'456.78 SFr. 123'456.78 CHF123'456.78
German(Austrian)	öS 123.456,78 ATS123.456,78

Locale	Newly supported currency formats
Italian(Swiss)	123'456.78 SFr. 123'456.78 CHF123'456.78
Norwegian(Bokmal), Norwegian(Nynorsk)	123 456,78 kr 123 456,78 NOK123 456,78
Portuguese(Standard)	123.456,78 Esc. PTE123.456,78
Portuguese(Brazilian)*	12 de outubro de 1954 05h04min30s PST
Spanish(Mexican)	123,456.78 \$123,456.78 MXN123,456.78
Swedish	123 456,78 123 456,78 kr SEK123 456,78

* In ColdFusion MX for Portuguese (Brazilian), the Java medium time format (for example, 06:55:27) does not work. However, the short, long, and full formats all work as expected; for example, 05h04min30s PST, 5h4min30s PST, and 05:04.

For more information, see `LSIsCurrency()` 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`) at <http://java.sun.com/j2se/1.3/docs/api/index.html>.

Some functions in ColdFusion MX produce slightly different results than in ColdFusion 5. Many of these differences are improvements. 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 greatest improvement is that the same code produces the same results, regardless of the ColdFusion platform.

Some of the differences might require a slight adjustment in your ColdFusion application. 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.

The rest of this section describes changes that affect specific locales and every locale.

Changes to formatting functions for different locales

The following table shows the locales in which one or more functions produce slightly different results between ColdFusion 5 and ColdFusion MX:

Locale	Function	ColdFusion 5 results	ColdFusion MX results
English(Australian)	LSCurrencyFormat(123456.78,"international")	USD123,456.78	AUD123,456.78
	LSCurrencyFormat("-1234.56","none")	(1,234.56)	-1,234.56
English(Canadian)	LSCurrencyFormat(123456.78,"international")	USD123,456.78	CAD123,456.78
	LSCurrencyFormat("-1234.56","none")	(1,234.56)	-1,234.56
English (New Zealand)	LSCurrencyFormat(123456.78,"international")	USD123,456.78	NZD123,456.78
	LSCurrencyFormat("-1234.56","none")	(1,234.56)	-1,234.56
English(UK)	LSCurrencyFormat(123456.78,"local")	\$123 456.78	£123,456.78
	LSCurrencyFormat(123456.78,"international")	USD123,456.78	GBP123,456.78
	LSCurrencyFormat("-1234.56","none")	(1,234.56)	-1,234.56
English(US)	LSParseDateTime("23:15:16 4 Sept. 1998")*	{ts '1998-09-04 23:15:16'}	invalid date format error
French(Belgian)	LSCurrencyFormat(123456.78,"none")	123 456,78	123.456,78
	LSCurrencyFormat(123456.78,"local")	123 456,78 F	123.456,78 FB
	LSCurrencyFormat(123456.78,"international")	FRF123 456,78	BEF123.456,78
French(Canadian)	LSCurrencyFormat(123456.78,"local")	123 456,78 F	123 456,78 \$
	LSCurrencyFormat(123456.78,"international")	FRF123 456,78	CAD123 456,78
	LSCurrencyFormat("-1234.56","none")	-1 234,56	(1 234,56)
French(Swiss)	LSCurrencyFormat(123456.78,"local")	123 456,78 F	SFr. 123'456.78
	LSCurrencyFormat(123456.78,"international")	FRF123 456,78	CHF123'456.78
German(Austrian)	LSCurrencyFormat(123456.78,"local")	123.456,78 DM	öS 123.456,78
	LSCurrencyFormat(123456.78,"international")	DEM123.456,78	ATS123.456,78
Italian(Standard)	LSCurrencyFormat(123456.78,"none")	12.345.678	123.457
	LSCurrencyFormat(123456.78,"none")		
Italian(Swiss)	LSCurrencyFormat(123456.78,"none")	12.345.678	123'456.78
	LSCurrencyFormat(123456.78,"local")	L. 12.345.678	SFr. 123'456.78
	LSCurrencyFormat(123456.78,"international")	ITL12.345.678	CHF123'456.78
Portuguese (Brazilian)	LSCurrencyFormat("-1234.56","none")	(1.234,56)	-1.234,56
Portuguese (Standard)	LSCurrencyFormat("-1234.56","none")	(1.234,56)	-1.234,56
	LSCurrencyFormat(123456.78,"local")	R\$ 123.456,78	123.456,78 Esc.
	LSCurrencyFormat(123456.78,"international")	BRL123.456,78	PTE123.456,78

Locale	Function	ColdFusion 5 results	ColdFusion MX results
Spanish(Mexican)	LSCurrencyFormat(123456.78,"none")	12.345.678	123,456.78
	LSCurrencyFormat(123456.78,"local")	12.345.678 pta	123,456.78
	LSCurrencyFormat(123456.78,"international")	ESP12.345.678	MXN123,456.78
	LSCurrencyFormat("-1234.56","none")	-123.456	(1,234.56)
Spanish (Standard), Spanish(Modern)†	LSCurrencyFormat(123456.78,"local")	12.345.678 pta	123.457 Pts
	LSCurrencyFormat(123456.78,"none")	12.345.678	123.457
Swedish	LSCurrencyFormat(123456.78,"none")	123.456,78	123 456,78

* 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. 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>.

† Spanish (Standard) and Spanish(Modern) map to the same Java locale `ES_es`. Therefore, whether you set the locale to Spanish(Modern) or Spanish(Standard), `getLocale()` returns Spanish(Standard).

For more information, see *CFML Reference* for the appropriate function.

Changes to formatting functions for every locale

The following table outlines changes to formatting functions that apply to every locale:

Function	Description of change
IsDateFormat()	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: <ul style="list-style-type: none">• 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
IsTimeFormat()	ColdFusion MX supports the following masks for IsTimeFormat(): <ul style="list-style-type: none">• h:m:s IsTimeFormat(6:39, "h:m:s") returns 06:39:00• hh:mm:ss IsTimeFormat(6:39, "hh:mm:ss") returns 06:39:00• hh:mm:ss t IsTimeFormat(6:28:26, "hh:mm:ss t") returns 06:28:26 A• hh:mm:ss tt IsTimeFormat(6:28:26, "hh:mm:ss tt") returns 06:28:26 AM
LSCurrencyFormat() LSEuroCurrencyFormat()	ColdFusion 5 put "EUR" as the international currency 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. ColdFusion MX uses the Java standard locales. Therefore, ColdFusion MX processes LSCurrencyFormat() and LSEuroCurrencyFormat() as follows: <ul style="list-style-type: none">• LSCurrencyFormat() returns the non-Euro currency format for all locales, even if the locale is a member of the Euro countries. For example, for the French(Standard) locale, LSCurrencyFormat() returns the French franc as the currency symbol, instead of the Euro.• LSEuroCurrencyFormat() only 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 -1 234,56 € for the French(Standard) locale, but returns (\$1,234.56) for the English(US) locale.
LSDateFormat()	In ColdFusion 5, for the date parameter, you can only use a date/time object in the range of 100 AD-9999 AD. ColdFusion MX does not have this limitation. 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.
LSDateFormat() LSTimeFormat()	If you pass an invalid mask into LSTimeFormat() or LSDateFormat(); for example, "dd-mm-yy mm", the function 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.
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).

Function	Description of change
<code>LSParseDateTime()</code>	<p>In earlier versions, you could not pass in a time zone value such as “EST”. Now you can, but ColdFusion MX processes this information as follows:</p> <ul style="list-style-type: none"> • If the time zone specified in the date/time string parameter is different from the time zone setting of the computer hosting ColdFusion MX, ColdFusion MX changes the time value in the parameter to its equivalent value in the computer’s time zone. • If the time zone is not specified in the date/time string parameter, ColdFusion MX does not adjust the time value. <p>When developing an application, Macromedia recommends that you evaluate the entered time string for a time zone setting and, if necessary, change your application to adjust the value. You can use a <code>cfcatch</code> block to do this; for more information, see <code>LSParseDateTime()</code> and “CFCATCH variables” in <i>CFML Reference</i>.</p> <hr/> <p>If you use the IBM JRE version 1.3.0 with ColdFusion MX, <code>LSParseDateTime("Quarta-feira, 23 de Janeiro de 2002 12h34min56s PST")</code> fails when passing the month “Janeiro”. The following code works with the SUN JRE.</p>

Note: The date pivot point for calculating the century in a two-digit year has changed from ColdFusion 5, affecting the results from date functions. For more information, see [“Date pivot point for two-digit years” on page 26](#).

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

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 5 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.
- `GetLocale()` 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 Java Virtual Machine (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.)

Other changes

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

Character encoding

ColdFusion MX uses the UTF-8 character set encoding as its default 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):

```
<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.

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 `cfauthenticate` and `cfimpersonate` tags, and the `authenticatedContext()`, `authenticatedUser()`, `isAuthenticated()`, `isProtected()`, and `isAuthorized()` functions—no longer works in ColdFusion MX. These tags and functions are obsolete in ColdFusion MX.

For more information about application security, see *Developing ColdFusion MX Applications with CFML*.

SNMP support

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

Variables

Following are changes to variables in ColdFusion MX:

- 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, `last.name` creates a struct called `last` with a key called `name`, instead of creating a simple variable whose name has a period in it.
To work around this, use underscores in variable names instead.
- The following `cfcatch` variables have changed: `Message`, `NativeErrorCode`, and `SQLState`. For more information, see `cfcatch` in [“CFML tags and attributes” on page 7](#).

Operators

Following are changes in operands between ColdFusion 5 and ColdFusion MX:

- Exponent results differ; for example, ColdFusion 5 returns an error for $0 \wedge 3$, 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 `last.name`, ColdFusion MX reads this code and creates a struct called `last` with a key called `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 `a.b.c="foo"` to be `a["b.c"]="foo"`, whereas ColdFusion MX interprets it to be `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 different case to create more than one key.

For example, the following code produces one key, not two:

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

For best results, use consistent case for struct key names.

COM objects

ColdFusion MX uses the Java Native Interface (JNI) to call COM objects, which results in slower performance than in ColdFusion 5. How much slower depends on the application and COM, but in the Macromedia tests of the same code, ColdFusion 5 completed in 50 milliseconds and ColdFusion MX completed in 2-3 seconds.

Template handlers

The Runtime service in Windows has been re-implemented in ColdFusion MX so that you can specify a template handler in the ColdFusion Administrator (Settings page). (A template handler is template to execute when the ColdFusion application server cannot find a requested template.) However, you must specify a template handler as a path that is relative to the web root directory, such as `\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 of 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. Macromedia has also tested using the SUN JRE version 1.3.1_01; however, many of these tests have failed on locales other than English(US).

For more information, see the Java documentation for the `java.text.SimpleDateFormat` class, at <http://java.sun.com/j2se/1.3/docs/api/index.html>.

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 “Variables and Reserved Words” chapter of *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 (“ ”). ColdFusion MX preserves null values, but converts them to an empty string when you use them as simple values.

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 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`

Tag and function examples

This section provides remedies for some of the compatibility issues listed in “[CFML tags and attributes](#)” on page 7 and “[CFML functions and variables](#)” on page 13.

Note: For the most current information on ColdFusion MX compatibility, see the Support section of the Macromedia website (<http://www.macromedia.com/support>).

Using cfregistry in ColdFusion MX

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

As an alternative, you can use the 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 the `cfregistry` tag to read ColdFusion server 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 server 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.

Following are two common examples of using the `cfregistry` tag in ColdFusion 5 that are not compatible with ColdFusion MX.

Incompatible example

In ColdFusion 5, the following sample code checks if a Verity collection exists, and if not, creates one in the appropriate directory relative to the installation path. In ColdFusion MX, this sample code throws an error, because ColdFusion MX does not store information about Verity collections and the installation directory in the registry.

```
<CFREGISTRY ACTION = "GETALL"
  Branch = "HKEY_LOCAL_MACHINE\SOFTWARE\Allaire\ColdFusion\CurrentVersion\
    Collections"
  Type = "KEY"
  Name = "Collections">
<CFSET CollectionFound = 'No'>
<CFLOOP QUERY = "Collections">
  <CFIF Collections.Entry IS request.datasource>
    <CFSET CollectionFound = 'Yes'>
  </CFIF>
</CFLOOP>
<CFIF CollectionFound IS 'No'>
<!-- Create the collection -->
  <CFREGISTRY ACTION = "GET"
```

```

    Branch = "HKEY_LOCAL_MACHINE\SOFTWARE\Allaire\ColdFusion\CurrentVersion\
    Collections"
    Entry = "RootDirectory"
    Type = "STRING"
    Variable = "CFRootDir">
<CFIF Server.OS.Name IS NOT "UNIX">
    <CFSET CollectionPath = "#CFRootDir#\Verity\Collections">
<CFELSE>
    <CFSET CollectionPath = "#CFRootDir#/verity/collections/">
</CFIF>
    <CFCOLLECTION Action = "CREATE"
    Collection = "#request.datasource#"
    Path = "#CollectionPath#"
    Language = "english">
</CFIF>

```

Compatible example

In ColdFusion MX, you cannot retrieve a list of Verity collections that are registered by ColdFusion or the K2 Server using the registry. Instead, use `cfcollection` with `action="list"`, perform a Query of Queries on the `queryResult` that is specified in the name attribute of `cfcollection`, and use `cfdump` to display the collections. Following is sample code to accomplish these tasks:

```

<cfcollection action = "list" name = "verity" >
<cfoutput>List of all known verity collections.<br><br></cfoutput>
<CFDUMP var = "#verity#">
<br>
<cfoutput>Searching for a specific collection by name.<br><br></cfoutput>
<cfquery name = "qoq" dbtype = "query">
select * from verity where verity.name = 'test_db_neo'
</cfquery>
<br>
<CFDUMP var = "#qoq#">

```

This returns a table of every collection that is registered by ColdFusion or the K2 Server. The table has the following columns:

- **Name** The name of the CF registered collection or the alias collection name for the K2 Server registered collection. ColdFusion MX saves the K2 Server registered collection information, so it is available even when the K2 Server is not running.
- **Registered** “CF” if the collection is registered by ColdFusion, or “K2” if the collection is registered by the K2 Server.
- **Path** The path to the collection. If the collection is external or registered by the K2 Server, the path includes the collection name.
- **Mapped** (CF collections only) “YES” or “NO” if the collection is mapped or not.
- **External** (CF collections only) “YES” or “NO” if the collection is external or not.
- **Language** (CF collections only) The language used by the collection.
- **Online** (CF collections only) “YES” or “NO” if the collection is available for searching or not.

Incompatible example

The following example to retrieve ColdFusion mappings throws an error because ColdFusion MX does not store ColdFusion mappings in the registry.

```
<cfregistry action = "get"
  branch = "HKEY_LOCAL_MACHINE\SOFTWARE\Allaire\ColdFusion\CurrentVersion\
    Templates"
  entry = "/cfdao/" type = "String" variable = "RegValue">
<cfif isdefined("regvalue")>
  <cfset request.cfdao_mappeddir = "#regvalue#">
<cfelse>
  <cfoutput>
    Please create a CF Mapping named cfdao which points to your cfdao files before
    trying to run this template.
  </cfoutput>
  <cfabort>
</cfif>
<cfinclude template = "/cfdao/dao_application.cfm">
```

Compatible example

To work around this, you can include error handling in your templates that would notify the system administrator if a `cfinclude` failed due to a missing mapping. In the event of a missing mapping, the system administrator must create it in the ColdFusion Administrator.

For more information on performing registry operations in ColdFusion MX, see `cfregistry` in the *CFML Reference*.

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:

```
<cfset list = "orange,Orange,apple">
<cfset listAsc = ListSort(#list#, "textnocase", "asc")>
<cfset listDesc = ListSort(1#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:

Sort order	ColdFusion 5 results	ColdFusion MX results
Ascending (listAsc)	apple, orange, Orange	apple, orange, Orange
Descending (listDesc)	orange, Orange, apple	Orange, orange, apple

For more information, see `ListSort()` 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 `cfindex` tag with the query attribute and `type = "file"` or `type = "path"`, for all actions that require information from the key attribute. You can also still use `type = "custom"`.

When you use `type = "file"` or `type = "path"` with a query, the action attribute queries to get filenames or file paths from the 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:

BookID	URL	Title	Description
bookid1	url1	title1	description1
bookid2	url2	title2	description2
bookid3	url3	title3	description3
file	http://localhost/cfdocs/snippets	title4	c:\inetpub\wwwroot\cfdocs\snippets\file.cfm
path1	http://localhost/cfdocs/snippets/ /	title5	c:\inetpub\wwwroot\cfdocs\snippets/ /
path2	http://localhost/cfdocs/cfmlsynt axcheck	title6	c:\inetpub\wwwroot\cfdocs\cfmlsynt axcheck

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

- Populate the `snippets` collection with files specified in the description column of the database, as shown in the following example:

```
<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 `snippets` collection with paths specified in the description column of the database, as shown in the following example:

```
<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>
```


A

- advanced security feature 24
- array functions 13
- ASCII 0 (NUL) 14
- attributes 7
- AuthenticatedContext 13
- AuthenticatedUser 13

C

- caching 7
- cfapplet 7
- cfapplication 7
- cfauthenticate 7
- cfcache 7
- cfcatch 8
- cfchart 9
- cfchartdata 9
- cfchartseries 9
- cfcol 8
- cfcollection 8
- cfdirectory 8
- cferror 8
- cfform 9
- cfgraph 9
- cfgraphdata 9
- cfgridupdate 11
- cfhttp 9
- cfimpersonate 7
- cfindex 9, 31
- cfinput 9
- cfinsert 11
- cfldap 10
- cflog 10
- cfloop 10
- cfmail 10

CFML

- data types 25
- functions and variables 13
- tags and attributes 7
- cfparam 10
- cfproccparam 11
- cfquery 11
- cfregistry 11, 28
- cfreport 12
- cfervlet 12
- cfsetting 12
- cfstpredproc 11
- cfswitch 12
- cfstable 8
- cfupdate 11
- character encoding 24
- charting 9
- Code Compatibility Analyzer 4
- connecting to data sources 16
- Crystal reports 12
- currency formats 18
- currency formatting 14, 18, 22

D

- data types 25
- database operations 16
- date and time objects 26
- date functions 13, 14, 22
- date pivot point for two-digit years 26
- DateDiff 13
- decimal numbers 23
- default values, setting 9
- DeleteClientVariable 13
- dot notation 25
- DSNs, for ODBC socket 16
- dynamic database connections 11

E

- empty arrays 13
- empty strings 26
- exception handling 8, 12
- exponent results 25
- expressions 12
- extended character support 18

F

- formatting decimal numbers 23
- formatting functions
 - for currencies 14, 18
 - for dates 14
 - for different locales 20
 - for every locale 22
 - overview 19
- forms, case in 25
- functions 13

G

- GetBaseTagList 13
- GetLocale 13, 23
- GetTempDirectory 13
- graphing 9

I

- IsArray 13
- IsAuthenticated 13
- IsAuthorized 13
- IsDateFormat 22
- IsProtected 13
- IsTimeFormat 22
- IsWDDX 14

J

- Japanese version 23
- JDBC 16
- JNI 25

L

- LDAP directory servers,
 - accessing 10
- Len 14
- ListSetAt 14
- ListSort 14
- ListSort() 30
- locales
 - determining 23
 - formatting functions 19
 - overview 18
- logging 10, 27
- LSCurrencyFormat 14, 20, 22
- LSDateFormat 22
- LsDateFormat 22
- LSEuroCurrencyFormat 22
- LSIsCurrency 18, 22
- LSIsDate 14, 22
- LSParseDateTime 14, 23
- LSTimeFormat 22

M

- mail, format of 10
- migrating
 - functions and variables 13
 - overview 2
 - planning 3
 - tag and function examples 28
 - tags and attributes 7
 - using Code Compatibility Analyzer 4, 5

N

- null values 26

O

- ODBC 16
- ODBC socket, using 16
- operators 25

P

- pi, definition of 26

Q

- Query of Queries 11
- query results
 - appearance of 17
 - data type of 17
- QueryColumn object 17

R

- registry operations 7, 11
- regular expression functions 15
- request variable 26

S

- security
 - new model for 24
 - obsolete functions for 13
 - obsolete tags for 7
- servlets using cfservlet 12
- session management 16
- SetLocale 23
- settings migration wizard 5
- SNMP support 24
- sorting
 - lists 14
 - struct values or keys 10, 15
- SQL reserved words 11
- StructKeyList 15
- structs
 - case of keys in 25
 - generating 25

T

- tags 7
- template handlers 25
- thisTag scope 27
- time functions 14, 22
- time zones 23
- two-digit years 26
- type IV database drivers 16

U

- Unicode 24
- UTF-8 24

V

- validating functions 8
- variables 13, 24
- Verity changes 8, 9

W

- WDDX 14

Y

- years, two-digit 26