

# What's New in this Release

- | Scorpio includes new features and enhancements in the following product areas:
  - Interaction with other technologies and Adobe products
  - Administration and Configuration
  - Debugging
  - Language Enhancements
  - Arithmetic and string operations
  - Database access
  - Generating reports and presentations
  - OEM support

## Interaction with other technologies and Adobe products

Feature/Tag	Description
AJAX UI support	<p>Several new features have been added that take advantage of AJAX capabilities. The following new tags let you control the layout and behavior of your application:</p> <ul style="list-style-type: none"> <li>• The <code>cflayout</code> tag can create four types of layout regions.</li> <li>• The <code>cflayoutarea</code> tag defines the regions contained inside a layout created by a <code>cflayout</code> tag.</li> <li>• The <code>cfwindow</code> tag creates a pop-up window in the browser that you can programmatically show and hide.</li> <li>• The <code>cfpod</code> tag creates a region with an optional title bar.</li> <li>• The <code>cfmenu</code> tag creates a horizontal or vertical menu with optional submenus.</li> <li>• The <code>cfmenuitem</code> tag creates an entry in a menu. A menu item can be the root of a submenu.</li> <li>• The <code>cftooltip</code> tag lets you specify an optionally HTML-formatted tool tip that appears when the user moves the mouse pointer over the items defined by any tags in the <code>cftooltip</code> tag body.</li> </ul>
	<p>The <code>cfgrid</code> tag now has an <code>html</code> attribute value of <code>AJAX</code> that creates a dynamic HTML grid. The grid can optionally use a bind expression (see below) to dynamically fill and update the grid contents. The grid can optionally be divided into pages with each page contents retrieved from the server when the user requests the specific page, thereby improving initial load time.</p>
	<p>The <code>cfgrid</code> tag has added the <code>bindOnLoad</code>, <code>onChange Refresh</code>, and <code>presevePageOnSort</code> attributes:</p>
	<p>The <code>cflayout</code> tag has a new <code>tabheight</code> attribute, used when <code>type="tab"</code>.</p>
	<p>The <code>cfmtree</code> tag now has a <code>format</code> attribute value of <code>html</code> that creates a dynamic HTML tree. The tree can optionally use a bind expression (see below) to dynamically fill and update the tree contents. Tree contents can optionally be retrieved from the server when the user opens tree branches, thereby improving initial load time.</p>
	<p>The HTML format <code>cfinput</code> tag now supports a <code>datefield</code> type that lets users select dates from a pop-up calendar.</p>
	<p>The HTML format <code>cftextarea</code> tag now supports a <code>richtext</code> attribute that creates a rich text editor that lets users easily insert rich HTML-formatted text. The editor has a developer-configurable toolbar with formatting commands that includes support for templates, custom styles, and several other features.</p>
	<p>You can now control the values of certain control attributes dynamically by using bind expressions and bind parameters. A bind parameter gets data from another form control attribute value and is updated every time a particular JavaScript event happens. A bind expression optionally uses the bind parameter to set a value dynamically.</p> <ul style="list-style-type: none"> <li>• The <code>cflayout</code>, <code>cfpod</code>, and <code>cfwindow</code> tags let you use URLs to set the value of their contents.</li> <li>• The <code>cfinput</code>, <code>cfselect</code>, and <code>cftextarea</code> tags let you use CFCs or JavaScript functions in bind expressions to set the value of their contents.</li> <li>• The <code>cfmtree</code> <code>cfmtreeitem</code>, and <code>cfgrid</code> tags let you use CFCs or JavaScript functions in bind expressions to fill the tree and grid contents.</li> <li>• The <code>cfinput</code> tag with a <code>text</code> type attribute lets you use a CFC bind expression to generate a suggestion drop-down list based on initial user input. After typing some characters, a user can select a suggested value from the list.</li> </ul>

Feature/Tag	Description
	The <code>cfselect</code> and <code>cftextarea</code> tags have added an <code>onBindError</code> attribute.
	The following features have also been added: <ul style="list-style-type: none"> <li>You can now use the <code>tooltip</code> attribute in HTML format versions of the <code>cfinput</code>, <code>cfselect</code>, and <code>cftextarea</code> tags. The tags can specify a URL as the source for the tool tip contents.</li> <li>A loading icon, and in some cases a "Loading..." text string, appears when a tag that uses a bind expression is loading the bound contents.</li> <li>A <code>QueryConvertForGrid</code> function has been added that converts a ColdFusion query into a paged data format that can be returned by a CFC to an AJAX format <code>cfgrid</code> tag to fill the grid contents.</li> </ul>
	Function signatures in JavaScript included dynamically through a <code>source</code> attribute must follow the format: <pre>&lt;functionName&gt; = function(&lt;arguments&gt;) {&lt;function body&gt;}</pre> The commonly adopted approach below may not work: <pre>function &lt;functionName&gt; (&lt;arguments&gt;) {&lt;function body&gt;}</pre>
	There is now URL bind support for data-bearing AJAX controls. This means that CFM files (rather than CFCs) can populate data for trees, grids, and so on.
	The <code>cfinput</code> , <code>cfpod</code> , <code>cfwindow</code> , <code>cflayout</code> , <code>cfsprydataset</code> , and <code>cfreeitem</code> tags now include an <code>onBindError</code> attribute to allow specification of a JavaScript error handler for normal and autosuggest binds.
	There is a new <code>ColdFusion.Ajax.submitForm</code> function.
	There is a new <code>ColdFusion.Grid.sort</code> JavaScript function.
	There is a new <code>AjaxOnLoad</code> function.
	There is a new <code>cfdiv</code> tag.
	The <code>source</code> attribute of the <code>cfdiv</code> tag has been renamed to <code>bind</code> . Values of the <code>bind</code> attribute now include all kinds of binds, including simple, JavaScript, URL, and CFC.
	For more information, see the tags listed above in <i>CFML Reference</i> .
AJAX application support	The new <code>cfajaxproxy</code> tag creates a JavaScript proxy for a ColdFusion component that you can then use in AJAX client code. The proxy and ColdFusion server-side support simplify the creation of AJAX applications that use CFCs by managing the interactions between the client and server. Features include providing JavaScript functions that correspond to the CFC functions and automatically serializing CFC return values into JSON format. AJAX support also includes three CFML functions: <code>SerializeJSON</code> , <code>DeserializeJSON</code> , and <code>IsJSON</code> , that manage JSON data. For more information, see <i>CFML Reference</i> .
	There is now <code>setForm</code> support for JavaScript proxies generated by the <code>cfajaxproxy</code> tag.
	The <code>cfajaximport</code> tag now has a <code>scriptsrc</code> attribute.
	The <code>cfajaximport</code> tag now has a <code>csssrc</code> attribute.
	The <code>serializeJSON</code> function now has a <code>serializeQueryByColumns</code> attribute.

Feature/Tag	Description
AJAX Application Generation wizard	<p>The AJAX Application Generation wizard is available as an Eclipse plugin. To install it, install the ColdFusion Plugins for Eclipse.</p> <p>To run the wizard, in Eclipse:</p> <ol style="list-style-type: none"> <li>1. Select File &gt; New &gt; Other.</li> <li>2. Under ColdFusion Wizards, select ColdFusion/AJAX Application Wizard, and then click Next.</li> <li>3. After reading the introductory text, click Next.</li> <li>4. Select the project whose settings you want to use or, if this is the first time you are running the wizard, click Next.</li> <li>5. Specify the RDS server on which you want the application to reside.</li> <li>6. Specify the data source to use, and then click Next.</li> </ol> <p>The wizard works similarly to the ColdFusion/Flex Application wizard. For information about that wizard, see <i>ColdFusion Developer's Guide</i>.</p>
JSON-related functions	<p>There are three new CFML functions: <code>SerializeJSON</code>, <code>DeserializeJSON</code>, and <code>IsJSON</code>, that manage JSON data.</p> <p>For more information, see <i>CFML Reference</i>.</p>
	<p>The <code>deserializeJSON</code> function has a new <code>strictMapping</code> argument.</p>
.Net assembly support	<p>You can access .NET assembly classes as ColdFusion objects by specifying <code>type=".Net"</code> in the <code>cfobject</code> tag or the <code>createObject</code> function. You can then use ColdFusion code to instantiate the class and call its methods and read and write its fields. The .NET support uses Java proxies to represent the .NET classes on the ColdFusion system, and requires that a .NET-side agent run on the system where the assemblies are located.</p> <p>For more information, see <i>ColdFusion Developer's Guide</i>.</p>
	<p>The <code>createObject</code> function and <code>cfobject</code> tag now support both "dotnet" and ".Net" as values for the <code>type</code> attribute.</p>
	<p>There is now a <code>DotNetToCFType(variable)</code> function. It converts a complex DotNet type variable to a ColdFusion array, structure, or query, depending on the type.</p>
Support for reading and creating Atom and RSS feeds	<p>The new <code>cffeed</code> tag can read RSS and Atom feeds in all commonly-used formats, and can create RSS 2.0 and Atom 1.0 feeds. The feed information can be represented in structure format or in a combination of structure and query formats. The tag also can save feed XML in a variable or file.</p> <p>For more information, see <i>CFML Reference</i>.</p>
Ability to manipulate existing PDF documents by using the new <code>cfpdf</code> and <code>cfpdfparam</code> tags	<p>The new <code>cfpdf</code> tag lets you manipulate existing PDF documents. ColdFusion also provides a subset of Adobe® LiveCycle™ Assembler functionality via the <code>processddx</code> action of the <code>cfpdf</code> tag</p> <p>Use the <code>cfpdfparam</code> tag with the <code>cfpdf</code> tag to merge several PDF documents into one document, or merge pages and page ranges from multiple PDF documents into one document.</p> <p>Use the new <code>IsPDFObject</code> function for validation.</p> <p>For more information, see <i>CFML Reference</i>.</p>
	<p>The <code>page</code> attribute changed to <code>pages</code>: This applies to both the <code>cfpdf</code> and <code>cfpdfparam</code> tags.</p>

Feature/Tag	Description
	The <code>TableOfContents</code> and <code>DocumentText</code> DDX elements are now supported by the <code>processddx</code> action.
	<p>Instead of sending a variable through “<code>#name#</code>”, you can use the variable name itself. The old syntax was:</p> <pre>&lt;cfpdf action="read" source="myPDF.pdf" name="myMPDF"&gt; &lt;cfpdf action="getInfo" source="#myPDF#" name="myInfo"&gt;</pre> <p>The new syntax is:</p> <pre>&lt;cfpdf action="read" source="myPDF.pdf" name="myMPDF"&gt; &lt;cfpdf action="getInfo" source="myPDF" name="myInfo"&gt;</pre> <p>For more information, see the <i>CFML Reference</i>.</p>
	Specifying the name or the destination attribute for <code>cfpdf</code> actions is no longer required. The <code>destination</code> attribute is required for <code>cfpdf action="write"</code> only. The <code>name</code> attribute is required for the <code>getInfo</code> , <code>processddx</code> , and <code>read</code> actions only. For more information, see the <i>CFML Reference</i> .
	You can specify multiple page ranges in the <code>pages</code> attribute of <code>thumbnail</code> action; for example, you can specify <code>&lt;cfpdf action="thumbnail" pages="1,3-4,7-11"...&gt;</code> For more information, see the <i>CFML Reference</i> .
	The default encryption for the <code>cfpdf action="protect"</code> is <code>RC4_128</code> . Before Beta 2, the default encryption was <code>RC4_40</code> . For more information, see the <i>CFML Reference</i> .
	The <code>cfpdf action="merge"</code> takes a new attribute called <code>stopOnError</code> . The default value is <code>yes</code> . If the value is set to <code>yes</code> , ColdFusion ignores all the files in a directory that are not PDF files. Set it to <code>No/False</code> if you want to merge all files in a directory that has some non-PDF files. No exception is generated if an invalid PDF file is encountered. For more information, see the <i>CFML Reference</i> .
	The <code>IsDDX</code> and <code>IsPDFFile</code> functions have been added. For more information, see the <i>CFML Reference</i> .
	The <code>processddx</code> action returns a reason if the process fails. For more information, see the <i>CFML Reference</i> .
	<p>The DDX element uses a new schema:</p> <pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt; &lt;DDX xmlns="http://ns.adobe.com/DDX/1.0/" xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance" xsi:schemaLocation="http://ns.adobe.com/DDX/1.0/ coldfusion_ddx.xsd"&gt;</pre> <p>For more information, see the <i>CFML Reference</i>.</p>
	The default value of the <code>stopOnError</code> attribute of the <code>cfpdf</code> tag has been changed to <code>false</code> .
	You can use the <code>Duplicate</code> function to create a clone of a PDF document in memory. You use the <code>read</code> action of the <code>cfpdf</code> tag to generate a PDF variable, and then you use the <code>Duplicate</code> function to create versions of it.
Ability to manipulate existing PDF forms by using the new <code>cfpdfform</code> , <code>cfpdfformparam</code> , and <code>cfpdfsubform</code> tags	<p>The new <code>cfpdfform</code> tag lets you manipulate existing Acrobat forms (PDF forms created with Adobe Acrobat 6.0 or earlier) and LiveCycle forms (PDF forms created with Adobe LiveCycle Designer).</p> <p>Use the <code>cfpdfformparam</code> tag to populate interactive fields in a PDF form.</p> <p>Use the <code>cfpdfsubform</code> tag to populate a PDF subform in the <code>cfpdfform</code> tag.</p> <p>For more information, see <i>CFML Reference</i>.</p>

Feature/Tag	Description
	The <code>cfpdfform</code> tag has a new <code>overwritedata</code> attribute.
	The <code>datafile</code> attribute of the <code>cfpdfform</code> tag has been renamed <code>xmldata</code> . When <code>action="read"</code> the <code>xmldata</code> attribute value can only be a variable name.
Ability to create bookmarks by using the <code>cfdocument</code> and <code>cfdocumentsection</code> tags	The <code>cfdocument</code> tag now supports bookmarks with the <code>bookmark</code> attribute. In addition, you can embed existing PDF forms using the new <code>cfpdfform</code> tags in the <code>cfdocument</code> tag. For more information, see <i>CFML Reference</i> .
<code>cfprint</code>	The new <code>cfprint</code> tag lets you collate, print, and finish PDF documents. Use this tag to generate automated batch printing jobs. For more information, see <i>CFML Reference</i> .
	The number of attributes have been greatly reduced and simplified for the <code>cfprint</code> tag. For more information, see <i>CFML Reference</i> .
	The <code>orientation</code> attribute has been removed from the <code>cfprint</code> tag.
	Use the new <code>GetPrinterInfo</code> function to determine which printer attributes are valid for a specified printer. For more information, see the <i>CFML Reference</i> .
	You can specify multiple page ranges for the <code>pages</code> attribute; for example, you can specify <code>pages="1,3-4,7-11"</code> .
	There is a new <code>GetPrinterInfo</code> function.
Flash Media Server event gateway	The FMS event gateway lets you modify data through the ColdFusion application or the Flash client, and reflect the change in the Flash Media Server shared object. The FMS event gateway listens to the shared object, and notifies ColdFusion when other clients modify shared objects. The FMS event gateway also lets ColdFusion modify shared objects.
Flex integration enhancements	The following enhancements have been made to ColdFusion/Flex integration: <ul style="list-style-type: none"> <li>• CFML applications can notify a Flex destination of changes in the data being managed by the destination.</li> <li>• CFML Assemblers can return a ColdFusion query object directly from the <code>fill</code> function without converting each row in to a value object (CFC).</li> <li>• CFML Assemblers can use structures instead of CFC's to represent data. The structure can contain type information which will be used as the ActionScript type when sent to the Flex application.</li> <li>• CFML applications can notify a Flex destination of changes in the data being managed by the destination via an event gateway. To use this feature, you must have the Beta 1 release of Borneo.</li> <li>• You can install LiveCycle Data Services as part of the Scorpio install.</li> <li>• You can run LiveCycle Data Services on the ColdFusion server.</li> </ul>
	Scorpio includes LiveCycle Data Services ES.

Feature/Tag	Description
Ability to interact with the Microsoft Exchange mail server through the <code>cfexchange</code> tags	<p>You can now use ColdFusion tags to interact with the Microsoft Exchange server. The Exchange interaction features include the following tags:</p> <ul style="list-style-type: none"> <li>• <code>cfexchangeconnection</code> -- Opens or closes a persistent connection with the Exchange server. You can use this connection in the other tags that interact with the server. You can also open temporary connections that last for the duration of a single tag by specifying connection attributes in other ColdFusion Exchange tags.</li> <li>• <code>cfexchangecalendar</code> -- Gets, creates, deletes, modifies, or responds to calendar events and gets calendar event attachments.</li> <li>• <code>cfexchangecontact</code> -- Gets, creates, deletes, or modifies contacts and gets contact attachments.</li> <li>• <code>cfexchangemail</code> -- Gets mail messages and attachments, deletes messages, and sets message properties. (Use the <code>cfmail</code> tag to create, reply to, and forward e-mail messages.)</li> <li>• <code>cfexchangegettask</code> -- Gets, creates, deletes, or modifies a user task, and gets task attachments.</li> <li>• <code>cfexchangefilter</code> -- Specifies the filter conditions, such as subject or time sent, to select the entries to get. Used only as a child tag of the <code>cfexchangemail</code>, <code>cfexchangecalendar</code>, <code>cfexchangegettask</code>, or <code>cfexchangecontact</code> tag get operations.</li> </ul> <p>For more information, see <i>ColdFusion Developer's Guide</i> and <i>CFML Reference</i>.</p>
	<p>The <code>cfexchangeconnection</code> tag has the following new attributes.</p> <ul style="list-style-type: none"> <li>• <code>formbasedauthentication</code></li> <li>• <code>ExchangeServerLanguage</code></li> </ul>
Ability to create and manipulate images with the <code>cfimage</code> tag and associated image functions	<p>You can create and manipulate ColdFusion images by using the <code>cfimage</code> tag or the <code>ImageNew</code> function. The <code>cfimage</code> tag provides shortcuts to the most common image actions, including reading, writing, resizing, rotating, and converting images. You can also create captcha images.</p> <p>You can use image manipulation functions to perform sophisticated image manipulation operations.</p> <p>For more information, see the <i>CFML Reference</i>.</p>
	<p>The following two functions check server image compatibility to determine the image formats that you can read and write:</p> <ul style="list-style-type: none"> <li>• <code>GetReadableImageFormats</code></li> <li>• <code>GetWriteableImageFormats</code></li> </ul> <p>For more information, see <i>CFML Reference</i>.</p>
	<p><b>CAPTCHA:</b> The <code>cfimage</code> tag no longer generates Base64 images by default. This means that you can write CAPTCHA images directly to the browser, even for Internet Explorer. You no longer have to write the CAPTCHA images to files before displaying them in an IE browser.</p>
	<p><b>IsImageFile function:</b> The new <code>IsImageFile</code> function lets you verify that an image file is a valid image. You no longer have to use the <code>&lt;cftry&gt;</code> and <code>&lt;cfcatch&gt;</code> statements to check for valid images uploaded to the server or extracted from a database.</p>
	<p><b>ImageAddBorder function:</b> The <code>thickness</code> parameter is now required.</p>
	<p><b>ImageRead function:</b> The syntax for the <code>ImageRead</code> function has changed. The new syntax is: <code>ImageRead("pathname")</code></p> <p>for example:</p> <pre>&lt;cfset myImage=ImageRead("images/aiden01.jpg")&gt; &lt;cfimage action="writeToBrowser" source="#myImage#"&gt;</pre>
	<p>The <code>ImageGetEXIFMetaData</code> and <code>ImageGetIPTCMetaData</code> functions no longer accept a structure as one of the parameters.</p>

Feature/Tag	Description
	<p><b>Image quality:</b> The default image quality for the <code>cfimage action="resize"</code> tag and for the <code>ImageResize</code> and <code>ImageScaleToFit</code> functions is greatly improved from the Beta 1 version of ColdFusion. Now, they use the <code>highestQuality (lanczos)</code> interpolation method to resample images. For a complete list of interpolation methods and their named equivalents, see the <i>CFML Reference</i>.</p> <p><b>Note:</b> For resize operation of 8-bit grayscale images, the supported interpolation algorithms are: <code>nearest</code>, <code>bilinear</code>, <code>bicubic</code>, <code>highestQuality</code>, <code>highQuality</code>, <code>mediumQuality</code>, <code>highestPerformance</code>, <code>highPerformance</code>, <code>mediumPerformance</code>.</p>
	You must specify both dimensions in the <code>ImageScaleToFit</code> function. In addition, the <code>ImageScaleToFit</code> function now has a <code>fitsize</code> argument.
	The <code>ImageResize</code> function has a new <code>blurFactor</code> attribute. The value range is 1-10. The higher the value, the more blurred the image is (and the more time it takes to process the image)
	Now, you can use the <code>cfimage</code> tag or the <code>ImageWrite</code> function to write GIF images on the Mac. For a list of standard image formats supported by ColdFusion, see the <code>cfimage</code> tag in the <i>CFML Reference</i> .
	The value of the <code>fonts</code> attribute of the <code>cfimage</code> tag can now be a comma-delimited list of fonts.
	<p>The <code>ImageNew</code> function has the following new attributes:</p> <ul style="list-style-type: none"> <li><code>canvasColor</code></li> <li><code>bgColor</code></li> </ul>
Ability to zip and unzip files and manipulate ZIP and JAR files by using the <code>cfzip</code> and <code>cfzipparam</code> tags	Two new tags, <code>cfzip</code> and <code>cfzipparam</code> , let you manipulate ZIP and JAR files. For more information, see <i>CFML Reference</i> .

## Administration and Configuration

Server Monitor	The Server Monitor is an Adobe® Flash® application that lets you keep track of activities on a ColdFusion Server. You can identify information about the server, including requests, queries, memory usage, and errors. You can start and stop collecting server information and take snapshots of the server.
Multiserver Monitor	The Multiserver Monitor is an Adobe® Flash® application that lets you keep track of the status of several ColdFusion Servers.
Ability to tune request performance in the ColdFusion Administrator	<p>In the ColdFusion Administrator, the Request Tuning page lets you specify:</p> <ul style="list-style-type: none"> <li>The maximum number of simultaneous requests by type, including Template requests, Flash Remoting requests, Web Service requests, and CFC function requests</li> <li>The maximum number of simultaneous report threads and threads available for the <code>cfthread</code> tag</li> <li>When to timeout requests waiting in the queue and the page to display when a request times out</li> <li>The maximum number of running and queued JRun requests</li> </ul>

Ability to disable creation of Java objects	In the ColdFusion Administrator, on the Settings page, use the Disable Creation Of ColdFusion Java Objects option to disable the ability for ColdFusion code to create Java object that are part of the ColdFusion package. This prevents non-authenticated CFML templates from reading or modifying administration and configuration information for this server.
Per application settings	You can set the following on a per-application basis: <ul style="list-style-type: none"> <li>• Mappings</li> <li>• Custom tag paths</li> </ul> For more information, in the ColdFusion Administrator, select Server > Settings, and then click Help. Click the Setting per application settings link.
RDS sandbox support	You can enable RDS globally or by user. For more information, in the ColdFusion Administrator, select Security > RDS, and then click Help.
User-based Administrator access	ColdFusion Administrator lets you specify permissions by user. The roles assigned to a user determine which pages in the ColdFusion Administrator and which functions in the Administrator API a user can access. To specify access for a user, in the ColdFusion Administrator, select Security > User Manager.
CFC type checking	The ColdFusion Administrator now lets you disable CFC type checking. This lets you simulate duck typing in CFCs. To disable type checking, in the ColdFusion Administrator, select Server Settings > Settings and then enable the Disable CFC Type Check option.

## Debugging

Scorpio includes a ColdFusion line debugger	The ColdFusion line debugger is an Eclipse plugin. You download the .ZIP file from the Adobe Prerelease site. Although the "How to Use the ColdFusion Debugger" demo illustrates stepping through code by using the Step In, it is preferable to use the Step Over button in the example. For more information, view the "Preparing to Use the ColdFusion Debugger" and "How to Use the ColdFusion Line Debugger" demos. Also, see <i>ColdFusion Developer's Guide</i> .
---	--

## Language Enhancements

Multithreaded processing	The new <code>cfthread</code> tag lets you create, end, join together, or temporarily suspend processing ColdFusion threads. Threads are independent streams of execution, and multiple threads on a page can execute simultaneously and asynchronously. Threads can continue processing after the page-level code completes. The new <code>Sleep</code> function temporarily suspends processing of a ColdFusion thread that was created by the new <code>cfthread</code> tag. It can also be used to suspend processing of a CFML page that does not use threads. A new lock scope attribute value, <code>Request</code> , has been added to support locking request-level code to enable you to prevent contention between threads on a page for Application or Request scope data that is shared among the threads. In Beta 1 a Name field was added to the thread metadata structure. For more information, see <i>ColdFusion Developer's Guide</i> and <code>cfthread</code> and <code>Sleep</code> in <i>CFML Reference</i> .
--------------------------	--

argumentsCollection attribute	<p>You can now specify an <code>argumentsCollection</code> attribute in most tags to specify the tag's attributes in a single structure. The attribute must be a structure containing all the tag's attributes as name value pairs. If you use the <code>argumentsCollection</code> attribute, you cannot use any other attributes directly in the tag.</p> <p>You cannot use the <code>argumentsCollection</code> attribute in the following tags. You can use it in all other tags.</p> <p><code>cfargument, cfbreak, cfcase</code>  <code>cfcatch, cfcomponent, cfdefaultcase</code>  <code>cfelse, cfelseif, cffunction</code>  <code>cfif, cfimport, cfinterface</code>  <code>cflogout, cflogin, cfloginuser</code>  <code>cfloop, cfparam, cfprocessingdirective</code>  <code>cfproperty, cfrethrow, cfreturn</code>  <code>cfset, cfsilent, cfswitch, cftry</code></p> <p><b>Note:</b> In Alpha 2, this attribute was named <code>arguments</code>.</p>
	<p>The <code>argumentsCollection</code> attribute has been renamed <code>attributeCollection</code>.</p>
CFC interfaces	<p>The new <code>cfinterface</code> tag lets you create object-oriented interface definitions. The <code>cfinterface</code> tag can contain only <code>cffunction</code> tags and their child <code>cfargument</code> tags; these tags define the signatures of the functions that make up the interface, and not the function implementations. You define the interface in a CFC file that has the <code>cfinterface</code> tag as its only top-level ColdFusion tag.</p> <p>You implement the interface in a ColdFusion component (CFC) that specifies the new <code>implements</code> attribute. Any CFC that implements the interface must implement each of the interface's functions as specified by the interface. A single CFC can implement multiple interfaces.</p> <p>The new <code>IsInstanceOf</code> function returns true if a ColdFusion object is an instance of a ColdFusion interface, a CFC, or a Java class, or is an object that extends the specified interface, component, or class.</p> <p>To get metadata for a ColdFusion interface, there is a new method <code>GetComponentMetaData</code> that accepts a component or interface name, and returns metadata.</p>
	<p>You can now duplicate a CFC with the <code>Duplicate</code> function.</p>
CFC support for an <code>onMissingMethod</code> function	<p>CFCs support an <code>onMissingMethod</code> function, which lets you handle calls to methods that are not implemented in the CFC, in the CFC itself instead of in the calling code. To use this feature, you define an <code>onMissingMethod</code> function in the <code>cfcomponent</code> tag body. If an application calls a method name in the CFC that is not defined, ColdFusion calls the <code>onMissingMethod</code> function and passes it the requested method's name and arguments. If you do not define an <code>onMissingMethod</code> function, a call to a method name that is not defined in the CFC causes ColdFusion to throw an error that must be handled in the calling code.</p>

<p>Application.cfc support for an <code>onMissingTemplate</code> function</p>	<p>The Application.cfc page can include an <code>onMissingTemplate</code> function, which will be invoked when ColdFusion encounters a file not found condition. When a ColdFusion page requests a page that cannot be found, ColdFusion searches for the Application.cfc file starting in the physical directory represented by the requested URI, and proceeds up the parent directories.</p> <p>The <code>onMissingTemplate</code> function must have the following format:</p> <pre>&lt;cffunction name="onMissingTemplate" returnType="boolean"&gt; &lt;cfargument type="String" name="targetPage" required="true"/&gt;   &lt;!--- Error handling code, such as cfoutput tags, goes here. ---&gt; &lt;cfreturn BooleanValue /&gt; &lt;/cffunction&gt;</pre> <p>The <code>onMissingTemplate</code> function must return true to indicate that the event has been processed or return false to indicate that the event has not been processed. If nothing is returned, its assumed to be true. If the function returns false, ColdFusion invokes the standard error handler. If an error occurs within the <code>onMissingTemplate</code> function, the error handler is not invoked</p> <p>If the <code>onMissingTemplate</code> function is invoked, the <code>onApplicationStart</code>, <code>onSessionStart</code>, <code>onRequestStart</code>, <code>onRequest</code>, and <code>onRequestEnd</code> handlers are not invoked, and processing of the request terminates when the <code>onMissingTemplate</code> handler returns.</p>
<p>Application.cfc <code>welcomefilelist</code> variable</p>	<p>You can now specify a <code>welcomefilelist</code> variable in the Application.cfc containing a comma delimited list of file names to tell ColdFusion not to call the <code>onMissingTemplate</code> method if any of the files on the list are not found. Use this variable to prevent ColdFusion from invoking the <code>onMissingTemplate</code> handler if the following are true:</p> <ul style="list-style-type: none"> <li>• The user is trying to browse a directory and you want to support directory browsing</li> <li>• Your web.xml uses a welcome file list and searches for files on the list when the user specifies only a directory path.</li> </ul> <p>You specify this variable only if the Application.cfc also specifies an <code>onMissingTemplate</code> handler. It should contain the same list of files as your web.xml welcome file list.</p>
<p>CFC serialization</p>	<p>Lets you use J2EE session replication in a cluster and have access to the CFCs in session data across all the machines in a cluster. This feature supports CF variables, including structures, in a replicated session scope. (Arrays are not supported in this release.)</p> <p>Also lets you preserve and access data in a CFC in the case of session failover. ColdFusion structures stored inside the session scope are available in the session scope, even after failover. For example, if you are running multiple ColdFusion instances to balance server load, you can store useful data, including CFCs, inside the session so that you can access the data across all the pages that are served in that session.</p> <p>The following example illustrates how to store a CFC in the session scope. Even if a client is switched to a different instance of the application server, the CFC data is still in the session scope.</p> <pre>&lt;cfset cfccomponent = CreateObject("component", "CFIDE.adminapi.datasource")&gt; &lt;cfset session.datasourcecomponent = cfccomponent&gt;</pre> <p>After failover, you can then access and call methods in the CFC as follows:</p> <pre>&lt;cfset check = session.datasourcecomponent.verifyDSN("testdsn")&gt;</pre>
<p>Implicit creation of arrays and structures</p>	<p>You can now create arrays and structures implicitly in assignment statements without using the <code>ArrayNew</code> or <code>StructNew</code> functions.</p> <p>To create a structure, delimit the structure contents in curly braces (<code>{}</code>). Inside the braces, create each key-value pair using <code>keyName="value"</code> (or <code>keyName=expression</code>) and separate the entries with commas, as in the following example:</p> <pre>&lt;cfset MyStruct={Mykey1="Myvalue1", Mykey2=MyString1 &amp; " is a string"}&gt;</pre> <p>To create an array, delimit the array contents in square brackets (<code>[]</code>). Inside the brackets, and separate array elements with commas, as in the following example:</p> <pre>&lt;cfset MyArray=[MyStruct, "Hello", 2, MyString2 &amp; "is another string", i++]&gt;</pre>

	<p>The following rules apply to implicit array and structure creation:</p> <ul style="list-style-type: none"> <li>You can also use implicit array and structure creation in CFScript.</li> <li>You cannot create nested arrays or structures in a single expression. For example, the following syntax is invalid:  <pre>&lt;cfset struct1 = {key1={innerstructkey1 = "innerstructvalue1"}}&gt;</pre> <b>Instead, use two statements, as follows:</b>  <pre>&lt;cfset innerstruct1={innerstructkey1 = "innerstructvalue1"}&gt; &lt;cfset struct1={key1=innerstruct1}&gt;</pre> </li> <li>You cannot include dots in key names when you create a structure. For example, <pre>&lt;cfset struct1={key1.key2="value"}&gt;</pre> is not valid.</li> <li>You cannot assign an implicitly created structure to an array element. For example, <pre>a[1]={key1="value1"}</pre> does not currently work.</li> </ul>
New file manipulation functions	<p>Several file manipulation functions have been added:</p> <ul style="list-style-type: none"> <li>The following functions let you read and write large files a chunk at a time, rather than in a single operation: <code>FileClose</code>, <code>FileIsEOF</code>, <code>FileOpen</code>, <code>FileRead</code>, <code>FileReadBinary</code>, <code>FileReadLine</code>, <code>FileWrite</code>, and <code>FileWriteLine</code>.</li> <li>The following file manipulation functions have also been added: <code>FileCopy</code>, <code>FileDelete</code>, <code>FileMove</code>, <code>FileSetAccessMode</code>, <code>FileSetAttribute</code>, <code>FileSetLastModified</code>, and <code>GetFileInfo</code>.</li> </ul>
New login-related functions	<p>The following functions have been added that provide information about a user's login status:</p> <p><code>IsUserInAnyRole</code> - Determines if the current user is in any of the specified roles in a list. This method checks the roles setup by the <code>cflogin</code> tag and by the servlet API.</p> <p><code>GetUserRoles</code> - Returns list of roles belonging to a user. This returns list of only ColdFusion roles and not roles set by servlet API</p> <p><code>IsUserLoggedIn</code> - A convenience function, the equivalent of <code>GetAuthUser() neq ""</code>.</p> <p>For more information, see <i>CFML Reference</i>.</p>

## Arithmetic and string operations

Ability to use JavaScript operators	<p>You can now use the following JavaScript operators in all CFML expressions: <code>++</code>, <code>--</code>, <code>%</code>, <code>+=</code>, <code>-=</code>, <code>*=</code>, <code>/=</code>, <code>%=</code>, <code>&amp;&amp;</code>, <code>  </code>, <code>!</code>.</p> <p>You can use the following operators in CFScript expressions only: <code>==</code>, <code>!=</code>, <code>&lt;</code>, <code>&lt;=</code>, <code>&gt;</code>, <code>&gt;=</code>.</p> <p>You can now use the <code>&amp;=</code> operator in all CFML expressions to concatenate the string specified to the right of operator on the end of the string variable specified in the left side of the operator.</p>
Arbitrary-precision decimal arithmetic	<p>The new <code>PrecisionEvaluate</code> function evaluates one or more string expressions using arbitrary-precision (BigDecimal) arithmetic to calculate the values of arithmetic expressions.</p> <p>For more information, see <i>CFML Reference</i>.</p>

## Database access

Ability to turn on logging for DataDirect data sources	<p>The DataDirect data sources (DB2 Universal Database, Informix, MS SQL Server, Oracle, and Sybase) include an option to turn logging on and off in the ColdFusion Administrator.</p>
<code>cfdbinfo</code>	<p>Lets you get information about a data source.</p> <p>For more information, see <i>CFML Reference</i>.</p>

cfprocparam	Lets you specify the name of parameters to stored procedures in any order using the dbVarName attribute.
-------------	--

## Generating reports and presentations

Report Builder enhancements	<p>Scorpio provides several enhancements to Report Builder to make it easier to use. Functionality includes:</p> <ul style="list-style-type: none"> <li>• Page breaks: You can add a page break anywhere in a report except in the page header or column header. To add a page break, click the page break button in the Toolbox, choose Insert &gt; Page Break, or use the right-click menu in the band where you want to insert the break.</li> <li>• Component locking: You can lock components so that they cannot be moved within the IDE. To lock a component, select the component and click the lock icon in the Toolbox or use the right-click menu. The status bar displays a locked status when one or more of the selected components are locked.</li> <li>• Component naming: You can name all components by using the new Name property. The Name property is located in the Data section of the Properties panel for the component. This functionality is required for future enhancements to Report Builder.</li> <li>• Component moving functionality is improved.</li> <li>• Line sizing functionality is improved.</li> <li>• Zoom functionality is improved.</li> <li>• Function Editor enhancements: You can edit the function name more easily by using the edit box at the top of the function section. You can use the Add Default Functions button to import and use the InitializeReport, BeforeExport, FinalizeReport functions as necessary. You can perform search and replace actions and use the right-click menu to perform commands in the Function Editor. You can use tag dialog boxes for any tags used in a function you are editing.</li> </ul>
Reporting Builder enhancements	<p>Scorpio provides several enhancements to Report Builder and the cfreport tag, including the following:</p> <p>Report Styles: In Report Builder you can create and apply Cascading Style Sheet (CSS) definitions by using the Report Styles Editor. Report Builder lets you do the following:</p> <ul style="list-style-type: none"> <li>• Create styles and apply them to elements in the report.</li> <li>• Export styles created in Report Builder to a CSS file. Report Builder generates the CSS code for you. Most of the generated styles are standard CSS styles; however, some are unique to Report Builder.</li> <li>• Import a CSS file. You can import a standard CSS file, or one generated by Report Builder, and apply it to the report. When you import a CSS file to a report, Report Builder automatically populates the Style Name drop-down lists with the styles in the report. This way, you can apply individual styles from the CSS file to report elements.</li> </ul> <p>See <i>CFML Reference</i> for a list of standard CSS styles supported by Report Builder and styles unique to Report Builder. (If your style sheet includes unsupported styles, ColdFusion ignores them).</p>

	<p>Follow these steps to define a new style in Report Builder:</p> <ol style="list-style-type: none"> <li>1. Choose Window &gt; Report Styles from the menu bar. The Report Styles window appears.</li> <li>2. Click the Add Report Style icon (the plus sign on the left). The Add Style dialog box appears.</li> <li>3. Enter a name for the style in the Name field. Click the tabs to define the font type, font size, color, style, transparency, and borders.</li> <li>4. (Optional) Click the Export Report Styles icon to export the style to a CSS file.</li> <li>5. (Optional) Click the Import Report Styles icon to import a CSS file from disk.</li> <li>6. Click OK to save the style.</li> </ol>
	<p>ColdFusion support for CSS styles: ColdFusion lets you override CSS styles in Report Builder reports and subreports at runtime. The <code>cfreport</code> and <code>cfreportparam</code> tags now have a new attribute called <code>style</code>. The <code>style</code> attribute takes a CSS file or an in-line style definition as its value. You can create CSS files in one of two ways: by exporting styles via the Export Report Styles icon in Report Builder or by creating a CSS file in any text editor. For the CSS styles to take effect; however, you must use Report Builder to assign the style names to the elements in the report. (The exception is the default style: you can use the <code>style</code> attribute to define the default style in ColdFusion and apply it to the report even if the default style is not defined in Report Builder.) After you have assigned the style names in Report Builder, you can update the style definitions in the CSS file at any time and apply them at runtime by using the <code>cfreport</code> and <code>cfreportparam</code> tags. If your report contains subreports, the default style applies to the master report and to all of the subreports. If the master report uses CSS styles other than the default style, the CSS styles do not apply to the subreports unless you specify them explicitly.</p> <p>See <i>CFML Reference</i> for a list of standard CSS styles supported by Report Builder and the styles unique to Report Builder. (If your style sheet includes unsupported styles, ColdFusion ignores them).</p>
	<p>The value of the <code>value</code> attribute of the <code>cfreportparam</code> tag can now be a variable.</p>
	<p>HTML and XML Report Formats: You can export your Report Builder reports in HTML or XML format. Follow these steps to export a report in HTML format from Report Builder:</p> <ol style="list-style-type: none"> <li>1. Choose Report &gt; Report Properties from the menu bar. The Report Properties dialog box appears.</li> <li>2. Choose HTML or XML from the Default Output Format list.</li> <li>3. Click OK. Report Builder writes any images in the CFR file to a temporary directory so that they display in the browser.</li> </ol> <p>Also, you can generate an HTML or XML report at runtime from ColdFusion by using the new HTML and XML values of the format attribute. For example:</p> <pre>&lt;cfreport template="myReport.cfr" format="HTML"&gt; &lt;cfreport template="myReport.cfr" format="XML"&gt;</pre> <p>When you export a report in HTML format, ColdFusion creates a temporary resource directory to store any image files or other resource files in the report when it is written to the browser. To specify when this resource directory is deleted, use the <code>resourceTimespan</code> attribute. The new attribute requires a time span (created using the <code>CreateTimeSpan</code> function) as a value. The following code deletes the resource directory after one hour:</p> <pre>&lt;cfreport template="myReport.cfr" format="HTML" resourceTimespan=#CreateTimeSpan(0,1,0,0)#&gt;</pre> <p>See <i>CFML Reference</i> for more information.</p>

	<p>Avoiding display problems in HTML reports: Unlike PDF and FlashPaper formats, which support free-form page content, the HTML format supports relative positioning of elements on a page for a grid-based layout. ColdFusion generates a <code>&lt;table&gt;</code> element for each report page and inserts each report element on that page inside a <code>&lt;td&gt;</code> tag. Because a table cell can contain only one element, report elements that overlap in the layout do not display correctly. When two elements overlap, the element behind does not appear in the exported HTML. To optimize a report for the HTML output format, follow these guidelines:</p> <ul style="list-style-type: none"> <li>• Minimize the number of rows and columns by aligning report elements, both on the horizontal and the vertical axis.</li> <li>• Eliminate any extra space between the elements.</li> <li>• Avoid overlapping report elements.</li> </ul>
	<p>Override queries in subreports and charts from ColdFusion: You can now override the queries used for subreports and embedded charts using <code>cfreportparam</code>. This way you can customize subreport and chart data from the CFM page without having to change your report. (The ColdFusion query must contain at least all of the columns included in the Report Builder query; however, the WHERE clause in the ColdFusion query can differ.) To support this feature, the <code>cfreportparam</code> has the following new attributes:</p> <ul style="list-style-type: none"> <li>• <code>subreport</code>: The name of a subreport. This value must match the Name property for the subreport defined in Report Builder.</li> <li>• <code>chart</code>: The name of a chart contained in a report or subreport. This value must match the Name property defined in Report Builder.</li> <li>• <code>query</code>: A query value to pass to the specified chart or subreport.</li> <li>• <code>series</code>: The number of the chart series for which to provide a query.</li> </ul> <p>See <i>CFML Reference</i> for the syntax.</p>
	<p>In Report Builder, charts and subreports now have a new property called Name. To pass queries from the CFM page to the report, the value of the subreport or chart attribute on the CFM page must match the Name property in the report. Follow these steps to specify a chart or subreport name in Report Builder:</p> <ol style="list-style-type: none"> <li>1. Select the subreport or chart element in the report.</li> <li>2. Choose Window &gt; Properties Inspector from the menu bar. The Properties Inspector for the selected element appears.</li> <li>3. Under the Data heading, enter the name for the element in the Name field. (Report Builder generates a default name.)</li> <li>4. Choose File &gt; Save from the menu bar to save your changes.</li> </ol>
	<p><b>7.0.2 Enhancements</b>  Report Builder introduced the following features in version 7.0.2, which were undocumented.</p> <p><b>XHTML Text Formatting:</b> You can specify XHTML text formatting for labels and text boxes. This means that if the field is XHTML-compliant, Report Builder applies the formatting to text when the report is printed. For example, if XHTML Text Formatting is set to True, Report Builder prints the following label in bold: <code>&lt;b&gt;This is label is bold&lt;/b&gt;</code>. Report Builder supports the following subset of XHTML-compliant tags:</p> <ul style="list-style-type: none"> <li>• <code>&lt;b&gt;</code>, <code>&lt;strong&gt;</code></li> <li>• <code>&lt;i&gt;</code>, <code>&lt;em&gt;</code></li> <li>• <code>&lt;u&gt;</code>, <code>&lt;underline&gt;</code></li> <li>• <code>&lt;s&gt;</code>, <code>&lt;strike&gt;</code></li> <li>• <code>&lt;font color="#HexValue" size="1 through 7 and +/-"&gt;</code></li> <li>• <code>&lt;p&gt;</code></li> <li>• <code>&lt;br&gt;</code></li> <li>• <code>&lt;a href="URL" anchor="anchor name"&gt;</code></li> <li>• <code>&lt;ul&gt;</code>, <code>&lt;ol&gt;</code>, <code>&lt;li&gt;</code> (no optional attributes for the <code>&lt;ul&gt;</code> and <code>&lt;ol&gt;</code> tags)</li> </ul>

	<p>To set XHTML Text Formatting, select the label or text box in the report band and then do one of the following:</p> <ul style="list-style-type: none"> <li>• Choose <b>Windows &gt; Properties Inspector</b> from the menu bar. The Properties Inspector for the selected element appears. Under the Formatting heading, choose <b>True</b> for XHTML Text Formatting.</li> <li>• Choose <b>Modify &gt; Properties</b> from the menu bar. The Properties dialog box appears. Click the Formatting Tab and select the XHTML Formatted Text checkbox. Click <b>OK</b>.</li> </ul>
	<p><b>Built-in Report Builder functions:</b> Report Builder provides three report-level built-in functions that you can access from the Report Function Editor. The functions exist only in Report Builder (not ColdFusion); therefore, you specify them in the report itself, not as part of the calling page. You do not call these functions explicitly; Report Builder calls them automatically when the report runs. The built-in functions are not required: do not define them if you do not need them. The built-in functions are as follows:</p>
	<ul style="list-style-type: none"> <li>• <b>InitializeReport:</b> This function executes before the report is run. If you are calling the report by using the <code>cfreport</code> tag in ColdFusion, you can use this function to access information on the CFM page. For example, you can verify that all the variables are initialized. You can include a <code>cfthrow</code> tag in the <code>InitializeReport</code> function to throw an error and force the report to not run. Note that, because the <code>InitializeReport</code> function executes before the report is run, you cannot use it to access any information in the report itself, only in the CFM page that calls the report. The following code shows an <code>InitializeReport</code> function that generates an error message if the user is not logged on:</li> </ul> <pre data-bbox="574 1010 1289 1136"> &lt;cffunction name="InitializeReport"&gt; &lt;cfif not(isDefined("url.loggedin"))&gt; &lt;cfthrow message="You need to be logged in to run this report."&gt; &lt;/cfif&gt; &lt;/cffunction&gt; </pre>
	<ul style="list-style-type: none"> <li>• <b>BeforeExport:</b> This functions lets you reorder pages in your report. For example, you can use this function to reposition a generated table of contents. To create an automatically generated table of contents, you need to place the table of contents element in the report footer so that it runs after the rest of the report. But, because the element is in the report footer, Report Builder prints it at the end of the report, by default. To print the table of contents at the beginning of the report, use the <code>BeforeExport</code> function. The following code shows how to use the <code>BeforeExport</code> function to reposition a generated table of contents after the report header but before the rest of the report:</li> </ul> <pre data-bbox="574 1430 1357 1791"> &lt;cffunction name="BeforeExport"&gt; &lt;cfargument name="pagecount"&gt; &lt;!-- Change this structure to return the pages you need. --&gt; &lt;cfset pages=StructNew()&gt; &lt;!-- Set the title page as the first page. --&gt; &lt;cfset pages["1"]="1"&gt; &lt;!-- Set the table of contents to the second page. --&gt; &lt;cfset pages["2"]="#arguments.pagecount#"&gt; &lt;!-- Set the rest of the report to follow the table of contents. --&gt; &lt;cfloop index="i" from="2" to="#arguments.pagecount - 1#"&gt; &lt;cfset pages["#i+1#" cant="i#"&gt; &lt;/cfloop&gt; &lt;cfreturn pages&gt; &lt;/cffunction&gt; </pre>

	<ul style="list-style-type: none"> <li>• <b>FinalizeReport:</b> This function executes after the report is run. Use it to clean up code or send e-mail notifications, for example, or run another report after the current report finishes. Note that if you write the report directly to the browser, without writing it to disk or a variable, Report Builder does not display the messages on the screen. In this case, you need to check the ColdFusion error log to check for errors. The following code uses the FinalizeReport function to write a line in the log file that confirms that the report finished running:</li> </ul> <pre>&lt;cffunction name="finalizeReport"&gt; &lt;cflog text="In FinalizeReport" type="Information" thread="yes" date="yes" time="yes" application="yes"&gt; &lt;/cffunction&gt;</pre>
	<p>Follow these steps to access the built-in functions in Report Builder:</p> <ol style="list-style-type: none"> <li>1. Choose Report &gt; Report Functions from the menu bar. The Report Function Editor appears.</li> <li>2. Click the Add Default Functions button (the first icon on the left). Report Builder adds the three functions to the list on the left.</li> <li>3. Click on a function in the list to display the edit area for that function.</li> <li>4. Enter the code for the function.</li> <li>5. Click OK to save the code and return to the report.</li> </ol>
	<p>The Justify option does not appear in the preview of the Properties dialog box or the Style dialog box. This is a limitation of the Preview Component in the Properties and Style dialog boxes.</p>
	<p>Styles are not passed from the parent report to any subreports (or subreport of subreports, etc) with the exception of the default style which will propagate down to any reports beneath it.</p>
Ability to create slide presentations dynamically from HTML code on a CFM page or Adobe Connect (SWF) files by using the new <code>cfpresentation</code> , <code>cfpresentationslide</code> , and <code>cfpresenter</code> tags	<p>The following tags let you generate Adobe Connect slide presentations. You can write the presentation files to disk or directly to a browser. The presentation can access live data from a database which you can use to populate charts and graphs. The <code>cfpresentation</code> tag is the container for the slide presentation. The <code>cfpresentationslide</code> tag creates a slide dynamically from an HTML page or SWF file. The <code>cfpresenter</code> tag describes a presenter in a slide presentation.</p> <p>For more information, see <i>CFML Reference</i>.</p>
	<p>The pathnames specified for the <code>audio</code> and <code>video</code> attributes of the <code>cfpresentation-slide</code> tag must be relative to the CFM page or the web root.</p>
	<p>The following attributes have been added to the <code>cfpresentation</code> tag: <code>proxyHost</code>, <code>proxyPort</code>, <code>proxyUser</code>, <code>proxyPassword</code>, <code>authUser</code>, <code>authPassword</code>, <code>userAgent</code>.</p>
	<p>The following attributes have been added to the <code>cfpresentationslide</code> tag: <code>authUser</code>, <code>authPassword</code>, <code>userAgent</code>.</p>
	<p>The <code>cfpresentation</code> tag now allows common colors and web color codes. Common colors include red, green, blue, black, yellow, magenta, pink, orange, cyan, gray, lightgray, and darkgray. Web colors are in the format <code>#FFFFFF</code>.</p>

## OEM support

JBOSS support	Scorpio supports JBOSS with the following specifications: <ul style="list-style-type: none"><li>• JBOSS 4.0.3SP1 (inherently uses Tomcat 5.5 Server Container)</li><li>• Sun JRE 1.4.2_09 and Sun JRE 1.5</li><li>• Host OS - Windows 2003 Server SP1 and RedHat Linux 4 AS</li></ul>
JDK 1.6 support	Scorpio supports JDK 1.6 as the runtime environment. By default, Scorpio installs and runs on JDK 1.6.
Support for the latest JDBC drivers and Database support	Scorpio supports the DataDirect Version 3.6 JDBC drivers.
Verity	The Verity Binaries are updated to V5.5.0 Service Pack 2 Patch 05 on all platforms.
Security	ColdFusion now includes the RSA JSafe encryption, which provides FIPs certified encryption, on Enterprise edition.

## What's Different in this Release

In addition to the new features listed in the previous section, the following changes have been made in Scorpio

- When deploying J2EE ColdFusion on WebSphere, the Application Name has been changed from MacroMedia ColdFusion MX.ear to Adobe Scorpio.ear.
- The Apache FTP library (provided by the Apache Commons commons-net library) bundled with ColdFusion has been upgraded from version 1.2.2 to version 1.4.0.
- The version of Log4J that is bundled with ColdFusion has been upgraded from version 1.1.3 to version 1.2.12.

The following table lists changes to existing ColdFusion tags and functionality:

Feature/Tag	Description
<p>When deploying J2EE ColdFusion on WebSphere, the Application Name has been changed from Macromedia ColdFusion MX.ear to Adobe Scorpio</p>	<p>For example,</p> <pre data-bbox="557 436 1370 520">/opt/WebSphere/AppServer/installedApps/cfqa40/MacromediaColdFusionMX.ear/ is now: /opt/WebSphere/AppServer/installedApps/cfqa40/AdobeScorpio.ear/</pre>
<p>Scorpio supports the DataDirect 3.6 drivers</p>	<p>Scorpio supports the following:</p> <ul style="list-style-type: none"> <li>• Auto-generated key retrieval - return a list of the ids of the inserted data</li> <li>• Transaction savepoints</li> <li>• The ColdFusion Administrator enables logging of database activity</li> <li>• DataDirect drivers map all new data types to JDBC 3.0 types, all of which ColdFusion supports</li> <li>• Multiple result sets can now be open simultaneously</li> <li>• Oracle 10g (R2)</li> <li>• SQL Server 2005 <ul style="list-style-type: none"> <li>- New varchar(max), nvarchar(max), varbinary(max), and xml data types</li> <li>- Support for named parameters</li> <li>- New XML data type supports native XML handling and XQuery syntax</li> <li>- Optional OUTPUT clause to insert, update and delete statements can return a result set</li> </ul> </li> <li>• Sybase ASE 15 <ul style="list-style-type: none"> <li>- Support for new bigint, unsigned bigint, unsigned int, unsigned smallint, and unitext data types</li> </ul> </li> <li>• Informix 10</li> <li>• DB2 UDB 8.2</li> <li>• DB2 UDB for OS/390 v6.1</li> <li>• OEM PostgreSQL JDBC 3.0 Type IV driver to ship with Scorpio supports Postgres 8.1 <ul style="list-style-type: none"> <li>- Supports transactions and savepoints</li> </ul> </li> </ul>
<p>Flash forms</p>	<p>Flash Form MXML keyword restrictions have been removed.</p>
<p>Event Gateways</p>	<p>The limits for Event Gateways have been changed in the Developer edition, as follows:</p> <ul style="list-style-type: none"> <li>• Event gateway are now supported</li> <li>• The limit on Buddies/SMS client has been increased from 1 to 3.</li> </ul>
<p>JNDI port</p>	<p>Scorpio uses 2930 as the JNDI port, so that it can coexist with ColdFusion MX 7.</p>
<p>Administrator API</p>	<p>Administrator API <code>cfide.adminapi.runtime.cleartrustedcache()</code> function now takes an optional <code>templateList</code> argument, which is comma separated list containing the template paths to be deleted from trusted cache. The template paths should be full physical paths to the templates.</p> <p>If any of the templates in the list do not exist, ColdFusion throws an error with a cumulative list of the non-existent templates and clears the template cache for rest of templates.</p> <p>The following example show how to use this function:</p> <pre data-bbox="557 1688 1349 1730">cfide.adminapi.runtime.clearTrustedCache(templateList="c:/cfmx/wwwroot/example.cfm,c:/cfmx/wwwroot/example2.cfm");</pre>

Feature/Tag	Description
cfdirectory	<p>The <code>cfdirectory</code> tag now includes following attributes:</p> <pre>listInfo = " name all" type = "file dir all"</pre> <p>The <code>listInfo</code> attribute lets you limit the returned query to include only the name of the files. It can also improve performance significantly.</p> <p>The <code>type</code> attribute lets you restrict the returned list to include only files or directories. The default action is to include both.</p>
cfdocument, cfdocumentsection, cfdocumentitem	<p>The <code>cfdocumentsection</code> tag now includes the following attributes:</p> <pre>name = "bookmark name"</pre> <p>For more information, see <i>CFML Reference</i>.</p>
	<p>Two new scope variables have been added: <code>totalSectionPageCount</code> and <code>currentSectionPageNumber</code>.</p>
	<p>If you do not specify the <code>mimeType</code> attribute for the <code>cfdocument</code>, <code>cfdocumentitem</code>, or <code>cfdocumentsection</code> tag, ColdFusion determines the MIME type based on the filename of the source.</p>
	<p>The following attributes have been added to the <code>cfdocument</code> tag: <code>proxyHost</code>, <code>proxyPort</code>, <code>proxyUser</code>, <code>proxyPassword</code>, <code>authUser</code>, <code>authPassword</code>, <code>userAgent</code>.</p>
	<p>The following attributes have been added to the <code>cfdocumentsection</code> tag: <code>authUser</code>, <code>authPassword</code>, <code>userAgent</code>.</p>
	<p>The following attribute was added to the <code>cfdocument</code> tag in Alpha 1, but was undocumented: <code>saveAsName</code>, which provides the option to supply a PDF filename in the SaveAs dialog when a user saves the output from the browser (before this attribute was added, ColdFusion supplied the CFM page name).</p>
cfdump	<p>The <code>cfdump</code> tag lets you specify which columns of a query to display, which keys and how many keys of a structure to display, and whether or not to include UDFs. You can send the output of the <code>cfdump</code> tag to the console, the browser, or a file.</p> <p>For more information, see <i>CFML Reference</i>.</p>
cfexchangeconnection	<p>Added <code>getsubfolders</code> action. This action gets a query with the names, paths and sizes of subfolders of a folder and can optionally get information about all lower level subfolders.</p> <p>For more information, see <i>CFML Reference</i>.</p>
cfexchange* tags	<p>The <code>cfexchangecalendar</code>, <code>cfexchangechangecontact</code>, and <code>cfexchangechangecontact</code> tags now return an additional, <code>HTMLmessage</code> column in the query attribute variable, and the <code>cfexchangechangecontact</code> tag returns an <code>HTMLDescription</code> column in the query attribute variable. These columns contain an HTML-formatted version of the Message or Description field text.</p>
cform	<p>If you specify the <code>scriptsrc</code> attribute in the <code>cform</code> tag and are using Flash forms, you must include the <code>CF_RunActiveContent.js</code> file in the directory you specified in the <code>scriptsrc</code> attribute. You can copy this file from <code>wwwroot/CFIDE/scripts/CF_RunActiveContent.js</code>.</p> <p>You only need it if:</p> <ul style="list-style-type: none"> <li>• You override the default</li> <li>• You are using Flash</li> </ul>
cfftp - secure FTP support	<p>The <code>cfftp</code> tag now includes the following attributes:</p> <pre>key = "private_key" passphrase = "passphrase" fingerprint = "ssh-dss.ssh-rsa" secure = "yes no "</pre> <p>For more information, see <i>CFML Reference</i>.</p>

Feature/Tag	Description
cffftp	The cffftp tag has new values for the <code>action</code> attribute: <code>allo</code> , <code>site</code> , <code>acct</code> , <code>quote</code> ; a new <code>bufferSize</code> attribute; and a new <code>actionparam</code> attribute.
cfargument and cffunction	The cffunction tag <code>returnType</code> attribute and cfargument tag <code>type</code> attribute now accept the value <code>component</code> to identify that the argument type or function return type is a ColdFusion component. In other words, you can now specify <code>component</code> instead of <code>WEB-INF.cftags.component</code> as these attributes' values. As in previous releases, you can specify that the argument or return type is a specific component by using the component name.
cffunction	The cffunction tag has a new <code>returnFormat</code> attribute that controls the format of data that is returned to a remote caller such as a web service or an AJAX page.
cfhttp	The cfhttp tag has the following new attributes: <code>ClientCert</code> - full path to a PKCS12 format file containing the client certificate for the request <code>ClientCertPassword</code> - the password used to decrypt the client certificate.
cfinvoke	The cfinvoke tag now has a <code>refreshWSDL</code> attribute, which lets you reload a WSDL and regenerate all of the artifacts used to invoke this web service. This can be used for development when the WSDL for a service is changing.
cfldap	The cfldap tag's <code>returnAsBinary</code> attribute now accepts a comma as the delimiter in a list, for example: <code>&lt;cfldap returnAsBinary="objectGUID ,objectSID"&gt;</code> .
cflocation	The cflocation tag has a new <code>statusCode</code> attribute that allows you to specify a redirection status code in the range between 300 and 307. The default value of 302, which is the redirection code used in previous versions of ColdFusion. For details on the status codes, see HTTP RFC 2616 - section 10.3 Redirection 3xx.
Locale-specific functions have new <code>locale</code> attribute	The following functions now support a <code>locale</code> parameter, which lets you the specified locale instead of the current locale of the page. <ul style="list-style-type: none"> <li>• DayOfWeekAsString</li> <li>• LSCurrencyFormat</li> <li>• LSDateFormat</li> <li>• LSEuroCurrencyFormat</li> <li>• LSIsCurrency</li> <li>• LSIsDate</li> <li>• LSIsNumeric</li> <li>• LSNumberFormat</li> <li>• LSParseCurrency</li> <li>• LSParseDateTime</li> <li>• LSParseEuroCurrency</li> <li>• LSParseNumber</li> <li>• MonthAsString</li> </ul>

Feature/Tag	Description
cfloop enhancements	<p>The following enhancements have been made to the cfloop tag:</p> <p>You can use the cfloop tag with the file attribute to read a file in a loop. You can read one line at a time, by specifying the index="line" attribute without a from or to attribute, or a specified number of characters at a time, as shown in the following example:</p> <pre>&lt;cfloop file="c:\temp\myfile.txt" index="line"&gt;   &lt;cfoutput&gt;#line#&lt;/cfoutput&gt; &lt;/cfloop&gt;</pre> <pre>&lt;cfloop file="c:\temp\myfile.txt" index="line" from="5" to="10"&gt;   &lt;cfoutput&gt;#line#&lt;/cfoutput&gt; &lt;/cfloop&gt;</pre> <p>A new array attribute lets you iterate over entries in an array, as shown in the following example:</p> <pre>&lt;cfset x = ["mars","earth", "venus", "jupiter"]&gt; &lt;cfloop array=#x# index="name"&gt;   &lt;cfoutput&gt;#name#&lt;/cfoutput&gt;&lt;br&gt; &lt;/cfloop&gt;</pre>
Ability to cache query objects when you use the cfqueryparam tag with the cfquery tag.	<p>You can now cache query objects when you use the cfqueryparam tag in a query, as the following example shows:</p> <pre>&lt;cfset CourseID=1&gt; &lt;cfquery name="getCourse1"   datasource="courses"   cachedafter="1-1-99"&gt;   select * from Courses   where ID = &lt;cfqueryparam value = "#CourseID#"   CFSQLType = "CF_SQL_INTEGER"&gt; &lt;/cfquery&gt;</pre>
cfreport	<p>The cfreport tag's format attribute now allows XML and HTML values. The Report Builder also supports these output formats. The cfreport tag has two new attributes:</p> <ul style="list-style-type: none"> <li>• style: used for overriding CSS styles in Report Builder reports</li> <li>• resourceTimespan: used when exporting reports in HTML format</li> </ul> <p>For more information, see <i>CFML Reference</i>.</p>
cfreportparam	<p>The cfreportparam tag has four new attributes to support overriding data in Report Builder subreports and charts from the CFM page:</p> <ul style="list-style-type: none"> <li>• chart</li> <li>• query</li> <li>• series</li> <li>• subreport</li> </ul> <p>The new style attribute overrides CSS styles defined in Report Builder subreports. For more information, see <i>CFML Reference</i>.</p>
cfschedule	<p>The cfschedule tag's action attribute now allows pause and resume values. The ColdFusion Administrator also supports this functionality on the Scheduled Tasks page.</p>
Ability to cache the results of a stored procedure using the cfstoredproc tag	<p>The cfstoredproc tag now supports the cachedwithin and cachedafter attributes, like the cfquery tag.</p>
sendStringParametersAsUnicode	<p>The connect string sendStringParametersAsUnicode=true is no longer supported in Scorpio for SQL Server and Sybase data sources. Instead, you can enable the String Format option for SQL Server data sources in the ColdFusion Administrator.</p>
Updated Verity binaries	<p>Scorpio includes Verity V5.5.0 Service Pack 2 Patch 05. You can now connect to a Verity K2 installation with authentication enabled. To specify the username and password, in the ColdFusion Administrator, select Data &amp; Services &gt; Verity K2 Server.</p>

