



Make the Leap to CF8 Application.cfc

MDCFUG  
February 2009

*Ajay Sathuluri*

*<http://www.teratech.com>*

# Introduction

- Ajay Sathuluri
- Email: [ajay@teratech.com](mailto:ajay@teratech.com)
- URL: <http://www.teratech.com>
- 10+ years with ColdFusion
- 8 years with Fusebox
- Also work with SQL, Oracle, JavaScript, VB, HTML, Design, Print Media
- Server Tuning, Administration, Load Testing
- Teach one-on-one and custom classes
- On site and custom development
- Fusebox and Process Methodology



# The CF Application Framework

- Using Application.cfc
- Handling Application Events
- Handling Request Events
- Handling Session Events



# Application.cfc

- Introduced in CFMX7
- Application.cfc is intended to replace:
  - ✓ Application.cfm
  - ✓ OnRequestEnd.cfm
  - ✓ New features not available in those
- Uses an event-driven model to handle requests
  - ✓ Code must be tagged to execute in one or more of several pre-defined events



TeraTech

# Events in Application.cfc

- Application Startup
  - ✓ onApplicationStart()
- Application Shutdown
  - ✓ onApplicationEnd()
- Session startup
  - ✓ onSessionStart()
- Session timeout
  - ✓ onSessionEnd()



# Events in Application.cfc (cont.)

- Before each page request
  - ✓ onRequestStart()
- After each page request
  - ✓ onRequestEnd()
- Instead of the page being requested
  - ✓ onRequest()
- When an error occurs
  - ✓ onError()
- When the requested page doesn't exist (404 error) (New in CF8)
  - ✓ onMissingTemplate()



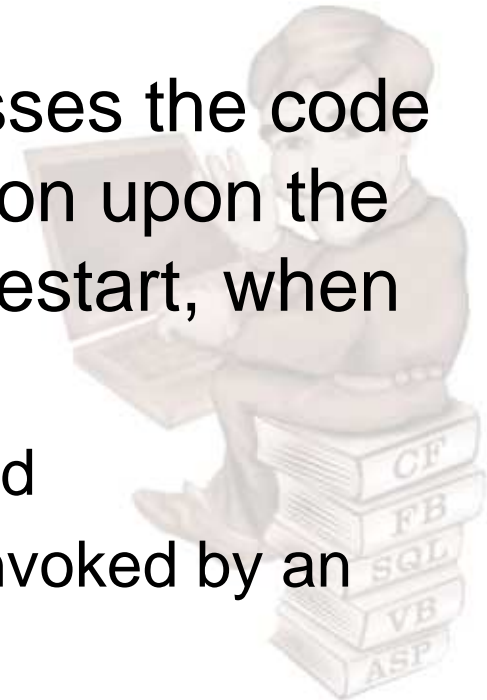
# Handling Application Events

- Using Application.cfc, you can respond to two application events:
  - ✓ onApplicationStart
    - Happens automatically on the first request of the application
  - ✓ onApplicationEnd
    - Happens behind the scenes when the application times out



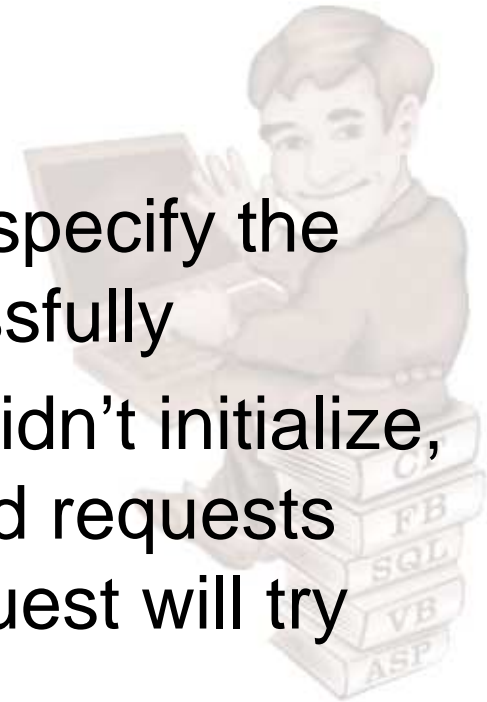
# onApplicationStart()

- ColdFusion automatically processes the code within onApplicationStart() function upon the following events after a system restart, when this Application doesn't exist yet:
  - ✓ first page of application is requested
  - ✓ first CFC within the application is invoked by an async gateway
  - ✓ first request to a web service or flash remoting CFC is encountered



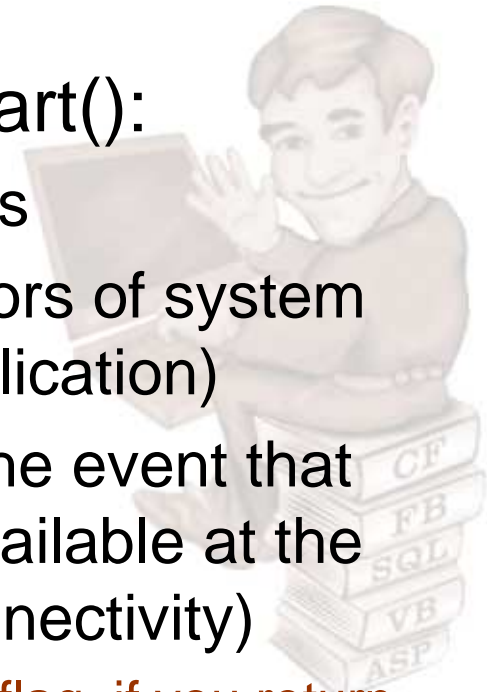
# onApplicationStart(), continued

- Returns a Boolean
  - ✓ True (or nothing, if you didn't specify the returnType): Initialized successfully
  - ✓ False: explicitly say that this didn't initialize, so application will not start and requests won't be processed. Next request will try again.



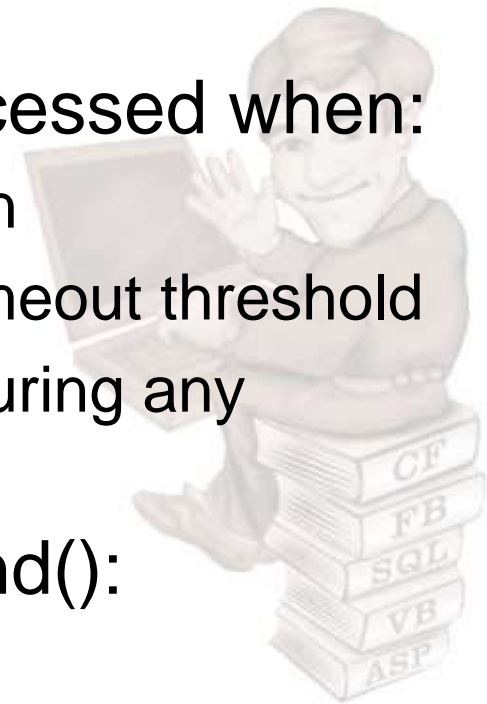
# onApplicationStart(), continued

- Typical uses for onApplicationStart():
  - ✓ Initialize global application variables
  - ✓ Log or otherwise notify administrators of system restarts (on first request of this application)
  - ✓ Deny access to the application in the event that access to critical resources is unavailable at the time of a restart (e.g. database connectivity)
    - (automatically, without having to set a flag, if you return False)



# onApplicationEnd()

- onApplicationEnd() function processed when:
  - ✓ ColdFusion Server is shutting down
  - ✓ Application reaches its inactivity timeout threshold
  - ✓ Happens behind the scenes, not during any request
- Typical uses for onApplicationEnd():
  - ✓ Event logging and notification
  - ✓ Writing cached information to a database (delayed write)



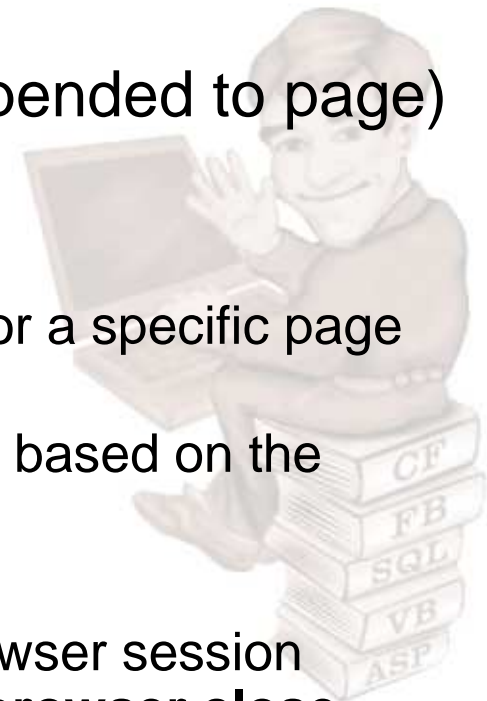
# Handling Request Events

- ColdFusion 7+ automatically processes the code contained within your Application.cfc's request functions when:
  - ✓ Before processing the requested page (onRequestStart)
  - ✓ After processing the requested page (onRequestEnd)



# onRequestStart()

- Runs when a page is requested (pre-pended to page)
- Possible uses
  - ✓ Redirects the user to a login page
  - ✓ Evaluates access rights to the whole site or a specific page or subdirectory
  - ✓ Performs redirection or method invocation based on the existence of url / form variables
  - ✓ Set request-level variables
  - ✓ Redo the CFID/CFTOKEN cookies as browser session cookies – enforce CF session closing on **browser close**
  - ✓ Optionally can reload application/session scope objects (but will not restart application/session; just manually rerun the onApplicationStart or onSessionStart method)



TeraTech

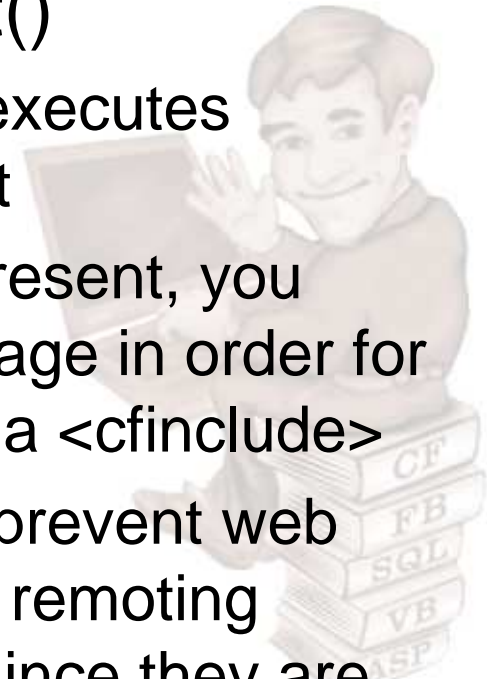
# onRequest()

- If defined, runs when a page is requested, instead of running that page directly
  - ✓ after onRequestStart()
  - ✓ before onRequestEnd()
- Possible uses:
  - ✓ Captures generated contents of a page and then performs some type of post-processing
    - e.g., the removal of white-space
  - ✓ Contains business logic to dynamically run a different page
    - There are generally much better ways to do that



# onRequest() : BE CAREFUL!

- Cautions when using onRequest()
  - ✓ The onRequest() method actually executes instead of the original page request
  - ✓ When an onRequest() method is present, you must explicitly execute the target page in order for it to be executed – usually through a <cfinclude>
  - ✓ Using an onRequest() method will prevent web services, event gateways, or Flash remoting requests from operating properly, since they are requests to a CFC method, and there is no template to include.



# onRequestEnd()

- Processed at the end of a page request
- In any Application.cfc, onRequestEnd() replaces OnRequestEnd.cfm which was previously used with an Application.cfm file
  - ✓ OnRequestEnd.cfm will no longer function if an Application.cfc file is present
- Typically used to output a common footer on your pages, such as a copyright notice



# Handling Session Events

- Application.cfc allows you to respond to two session-related events:
  - ✓ When a session starts
  - ✓ When a session ends



# onSessionStart()

- Processed at the start of each new user session
- Typically used to:
  - ✓ Initialize complex session data structures
  - ✓ Create session-based CFC instances



# onSessionEnd()

- Runs automatically when a user session ends
  - ✓ Not on a request
  - ✓ On the Session timeout, after the server hasn't seen the client cookies in that time period. (CFID/CFToken or J2EE sessionID cookie)
- Typical uses:
  - ✓ log end of session
  - ✓ write persistent data to database
- Does not run if application times out or if the server shuts down



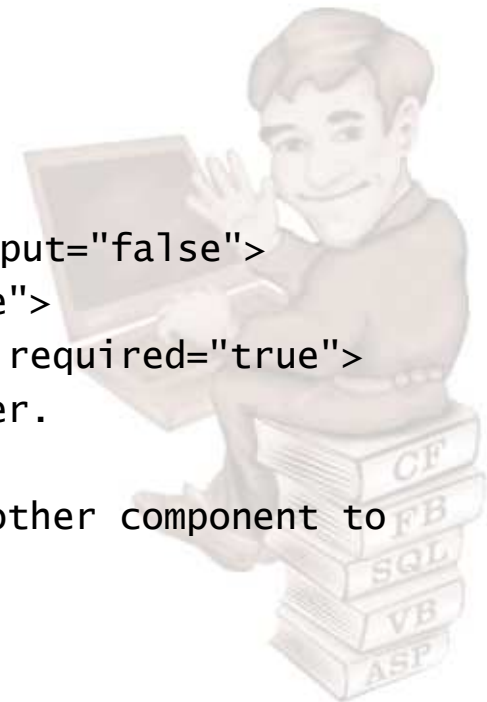
# Handling Error Events

- Application.cfc adds an OnError event
- Arguments:
  - ✓ Exception
    - CF Exception object, similar to the CFCatch
  - ✓ EventName
    - String, if error happened within an Application.cfc event instead of the requested page, gets the event name



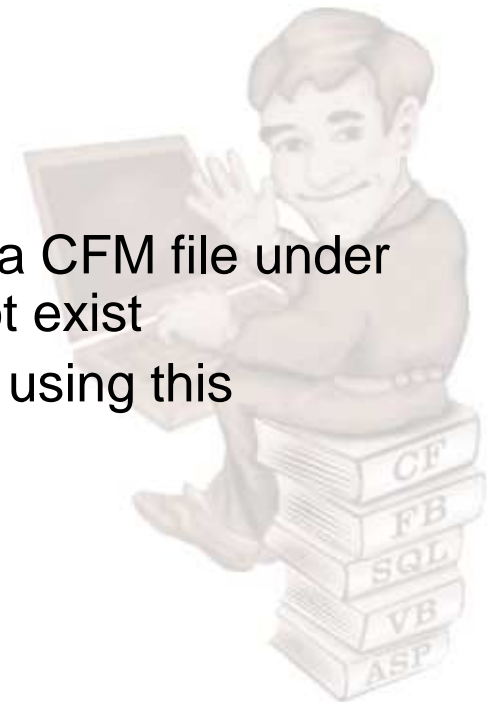
# onError() event

```
<!--- Runs on error --->  
<cffunction name="onError" returnType="void" output="false">  
  <cfargument name="exception" required="true">  
  <cfargument name="eventname" type="string" required="true">  
  <!--- Just dump the arguments to the browser.  
      NOT FOR PRODUCTION!  
      Handle better and more securely - use another component to  
      send an email? -->  
  <cfdump var="#arguments#"><cfabort>  
</cffunction>
```



# onMissingTemplate() event

- New in CF8
- 404 handler
  - ✓ Event launched by CF for any request for a CFM file under this Application.cfc, where the file does not exist
  - ✓ Check out the caveats in the LiveDocs for using this
- Arguments:
  - ✓ targetPage
- Returns:
  - ✓ True if this event handled it
  - ✓ False to rethrow the missing template error as an exception



# onMissingTemplate() cont'd

```
<!-- Used to help CF work with missing files and dir indexes -->
<cfset this.welcomeFileList = "">

<!-- Fired when user requests a CFM that doesn't exist. -->
<cffunction name="onMissingTemplate" returnType="boolean" output="true">
  <cfargument name="targetpage" required="true" type="string">

  <!-- Use a try block to catch errors. -->
  <cftry>
    <!-- Log all errors. -->
    <cflog type="error" text="Missing template: #Arguments.targetPage#">
    <!-- Display an error message. -->
    <cfoutput>
      <h3>#Arguments.targetPage# could not be found.</h3>
      <p>You requested a non-existent ColdFusion page.<br />
        Please check the URL.</p>
    </cfoutput>
    <cfreturn true />

  <cfcatch>
    <cfreturn false />
  </cfcatch>
</cftry>
</cffunction>
```



**TeraTech**

# onMissingMethod event

- New CFC event, new in CF8. Not just in Application.cfc
- OnMissingMethod method
  - ✓ Can handle calls to methods that are not implemented in the CFC
  - ✓ Definable in any CFCOMPONENT
  - ✓ Handle error in CFC itself rather than in calling code



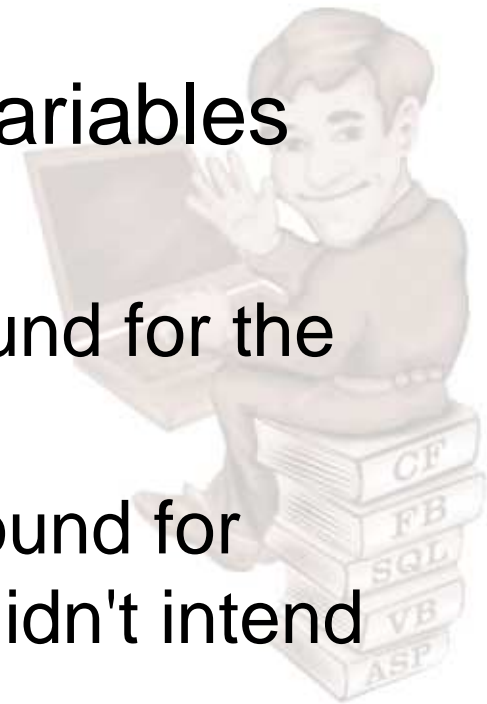
# OnMissingMethod Demo

- Demo: `cf8/usingOnMissingMethod.cfm`



# General comments

- Make sure to var-scope all variables within these methods!
  - ✓ The Application.cfc sticks around for the life of the application
  - ✓ Global variables may stick around for multiple requests where you didn't intend them



# Refactoring an Application.cfm

- Demo



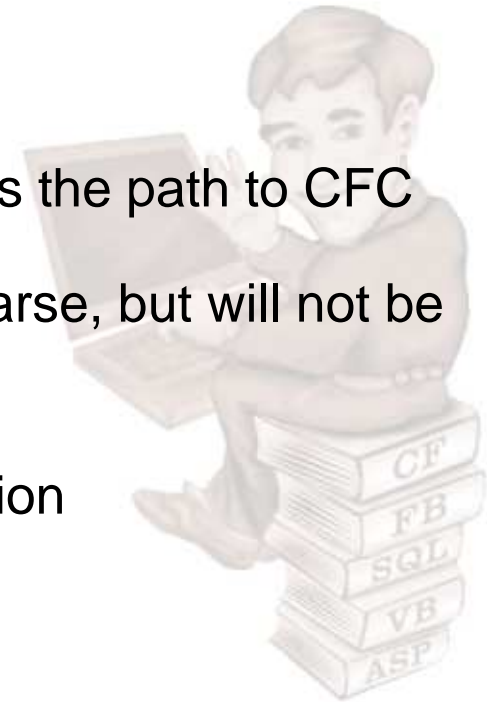
# Other new CF8 component functions



# Metadata - GetComponentMetaData

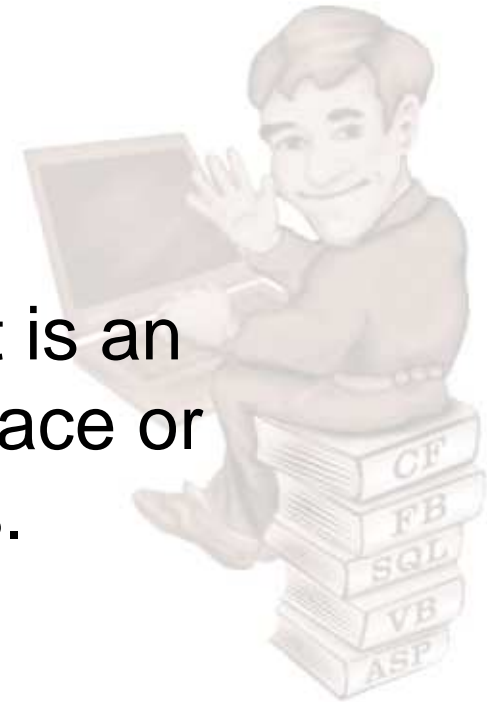
- GetComponentMetaData
  - ✓ New function in CF 8
  - ✓ Use in any code, on any component
  - ✓ Similar to getMetadata(object), except gets the path to CFC and doesn't instantiate it.
  - ✓ The component still needs to be able to parse, but will not be run
- Returns a whole heck of a lot of stuff!
  - ✓ Internal documentation, including description
  - ✓ All methods, with arguments and types
  - ✓ Properties
- Example:

```
<cfset simpleMeta = GetComponentMetaData("simple")>
<cfdump var="#simpleMeta#">
```
- Demo: cf8/getComponentMetaData.cfm



# IsInstanceOf()

- IsInstanceOf()
  - ✓ New in CF 8
  - ✓ Determines whether an object is an instance of a Coldfusion interface or component, or of a Java class.



# IsInstanceOf() - Demo

- Example:

```
<!-- component --->  
<cfset cfcsimple =  
createobject("component","simple")>  
#IsInstanceOf(cfcsimple, "simple")#  
(Expected = YES)  
#IsInstanceOf(cfcsimple, "simple1")#  
(Expected = No)
```

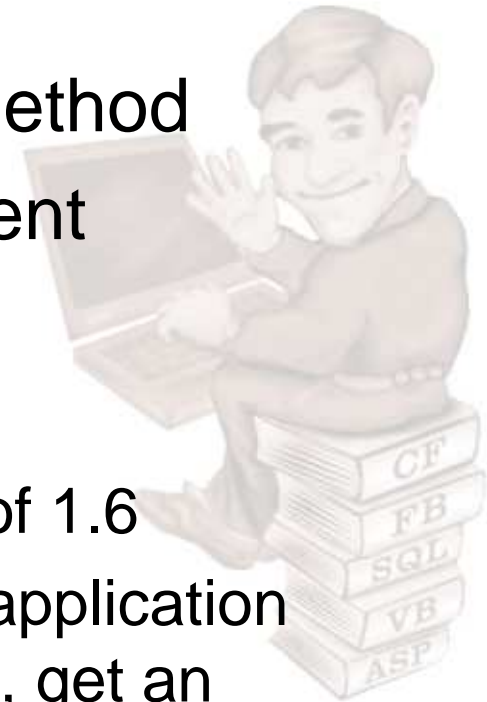
```
<!-- Create a Java object --->  
<cfset javaObj = createobject("java",  
"java.lang.System")>  
#IsInstanceOf(javaObj,"java.lang.System")# (Expected  
= YES)
```

- Demo: cf8/IsInstanceOf.cfm



# Other CF8 notes

- CF8 now has a built-in sleep() method
- General performance improvement
- CF8 can run on JRE 1.5, 1.6
  - ✓ CF8 installs with 1.6.0\_04
  - ✓ Performance and feature benefits of 1.6
  - ✓ Java JRE 1.6 classloader bug: on application startup, if loads lots of components, get an exponential performance issue
    - Update to 1.6 updater 10 or 11
    - [http://corfield.org/blog/index.cfm/do/blog.entry/entry/Java\\_6\\_and\\_ColdFusion\\_8](http://corfield.org/blog/index.cfm/do/blog.entry/entry/Java_6_and_ColdFusion_8)



# Other Resources

- [http://livedocs.adobe.com/coldfusion/8/htmldocs/AppEvents\\_01.html](http://livedocs.adobe.com/coldfusion/8/htmldocs/AppEvents_01.html)
  - ✓ Check here for usage / syntax of the Application.cfc events
- <http://www.coldfusionjedi.com/cfcresources.cfm>
- <http://www.coldfusionjedi.com/index.cfm?mode=entry&entry=ED9D4058-E661-02E9-E70A41706CD89724>
  - ✓ Warning about onError in 7.0 – fixed in 7.0.2
  - ✓ onRequest use when you need to call CFCs remotely
- [http://livedocs.adobe.com/coldfusion/8/htmldocs/appFramework\\_07.html#1191449](http://livedocs.adobe.com/coldfusion/8/htmldocs/appFramework_07.html#1191449)

