

Tardis Framework

MVC Pattern applied to ColdFusion MX

About Tardis

Shortly after the first release of CFMX (6.0) in 2002 I developed a reusable CF component-based (CFC) MVC Framework called MVCF-CF. At the time there were no CFC-based frameworks of any kind available. Those were still the days of Fusebox, when most CF developers might have thought MVC was a dubbed Japanese game show on SpikeTV;)

Once it was fully tested and refined, it became the basis of all my development and I started putting together documentation suitable for public consumption. Putting your code & ideas out for public critique or ridicule is kind of scary (especially the first time you do it, kind of like blogging), so it took me until August of 2004 to take the leap (grow some balls). By that time there were a couple of other MVC frameworks out there (Mach-II, etc), but I was far enough down my own path that it wasn't worth looking into a course change. Since the code was stable there didn't seem to be much reason to mess with something which worked, so there were no features added (1 bugfix) since then.

MVCF-CF was originally inspired by Struts (and the ideas of Benoit Hediard), but I'd never actually used that code, nor trained on it, so it was more conceptual inspiration rather than functional. Since the 2004 release I attended some classes on JSP & Servlet (with Struts) and EJB, which gave me some ideas for a feature-add release to the framework. I filed away those ideas for a day when I actually had time to work on them.

Goal

The goal of this effort is to provide a framework for building ColdFusion MX applications.

The core of the Tardis framework is a ColdFusion Components/Extensible Markup Language (CFC/XML) based control layer for developing Model-View-Controller 2 (MVC-2) architecture applications.

Tardis provides the Controller component and integrates with other technologies to provide the Model and the View. For the Model, Tardis can interact with other CFC's and data access technologies, like JDBC and EJB. For the View, Tardis works with ColdFusion pages (.cfm), JavaServer Pages (.jsp) and Flash (.swf).

This framework provides the underpinnings for developing web applications developed using standards and proven design patterns.

Overview of MVC

The MVC pattern is an architectural design pattern which lends itself well to developing n-tier web applications. MVC organizes an application into three modules, the Model which is responsible for data persistence and business logic, the View for providing data presentation and accepting user input, and the Controller which controls application flow.

MVC allows for the separation of data persistence and logic, presentation and control which promotes code reuse, decreased code duplication and more extensible and maintainable application development.

Tardis Framework

Overview of Tardis

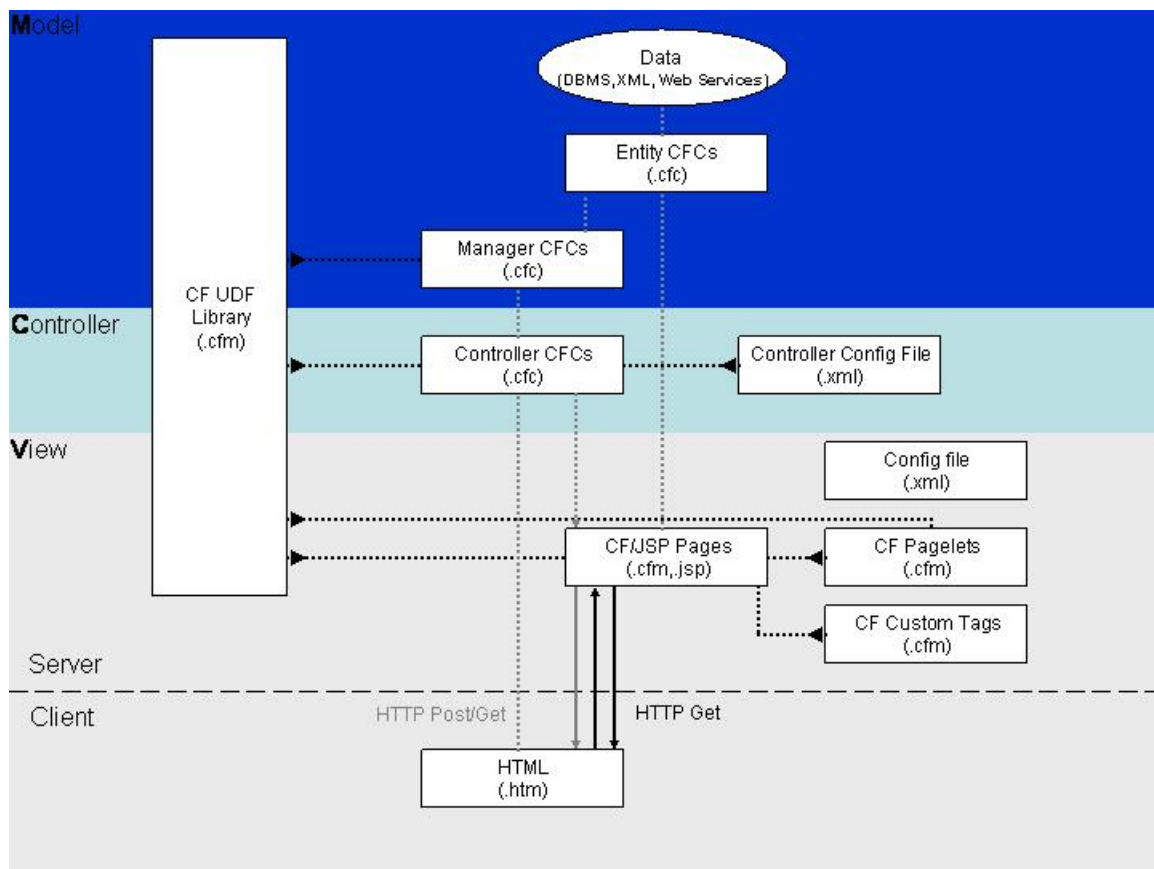
Tardis is an framework for developing MVC-2 web applications. The main purpose of this framework is to decouple presentation layer code and business layer code. The benefits of using this framework are:

- Application partitioning and modularity
- Increased code manageability
- Increased code extensibility

The reason why I developed this framework instead of using STRUTS was to shield CF developers for the perceived complexity of the underlying J2EE platform. For CF developers that are comfortable with using J2EE components, Tardis is flexible enough to allow the use of Flex, JSP, Servlets, Java Classes and EJB.

Prior to ColdFusion MX and CFC's the structure to provide a real MVC implementation was not available. There were other development frameworks available, but they promoted mixing the business and presentation layers, and were unwieldy for larger applications. This mixture makes code reuse difficult and even simple changes to the database layer can cause far reaching ripples of problems (brittleness).

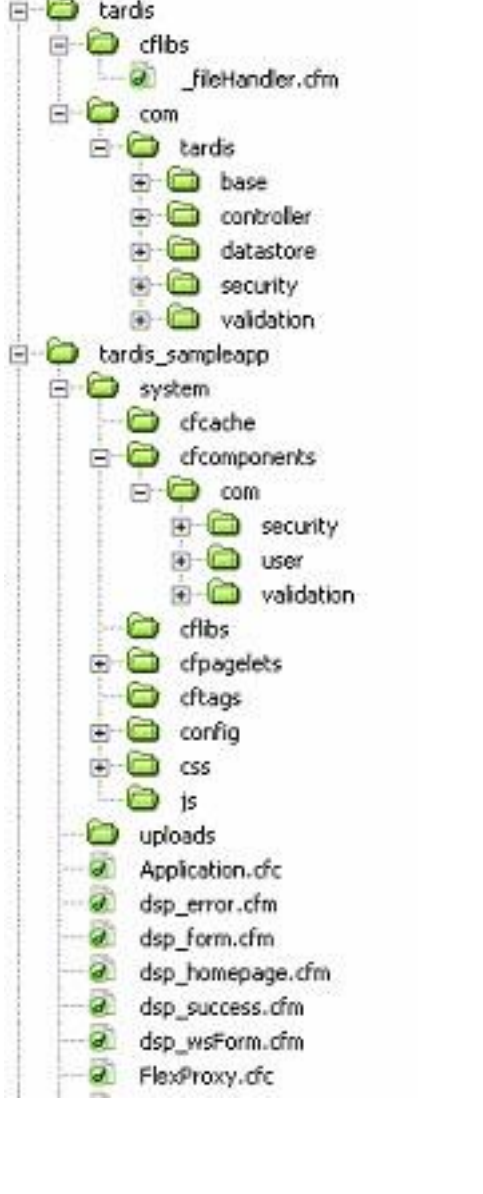
Below is a visual representation of Tardis



Tardis Framework

Organization of code

Tardis follows a modified version of Hediard's "distributed application" code organization convention, which is depicted below.

	<p>The top level folder is the application name and a CF and Web server mappings need to be added for the application.</p> <p>The root application folder will contain application.cfc and all view pages.</p> <p>The system folder contains the following subfolders:</p> <ul style="list-style-type: none">- cfcache: presentation cache,- cfcomponents: CFC's, including any model components or extended components- cflibs: UDF libraries,- cfpagelets: pagelet scripts- cftags: custom tags,- config: configuration files- css: stylesheet files,- js: Javascript files <p>Inside the cfcomponents folder, scripts are organized by package. In this example, the updateUser component is placed in a folder called "user" and would be accessed by /app/system/cfcomponents/com/user/userManager.cfm or alternatively like app.system.cfcomponents.com.user.userManager.</p> <p>The uploads folder is used for file uploading by the controller.</p>
--	--

An alternative to the method of component access detailed above, you may use a variable prefix in the place of the full path. The variable prefix is set in the application configuration file (cfapp-config.xml), which is used to initialize the application.

Example of component call with a variable prefix:

```
<cfobject type="component" name="userObj"  
component="#request.application.basecomponentpath#.com.user.User">
```

Tardis Framework

The corresponding configuration file entry (basecomponentpath):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE globals>
<globals>
  <!-- These variables are required for the controller framework -->
  <!-- These variables are paths to the controller and datastore -->
  <controllervirtualpath>/mvcframework/system/cfcomponents/com/controller/</controllervirtualpath>
  <controllercomponentpath>mvcframework.system.cfcomponents.com.controller</controllercomponentpath>
  <datastorevirtualpath>/mvcframework/system/cfcomponents/com/datastore/</datastorevirtualpath>
  <datastorecomponentpath>mvcframework.system.cfcomponents.com.datastore.datastore</datastorecomponentpath>
  <!-- This is the max number of rows of error data stored in Datastore -->
  <datastoremaxerrorsstored>20</datastoremaxerrorsstored>
  <!-- These variables are required for file uploading -->
  <!-- This is the file upload directory-->
  <fileuploaddirectory>D:\apps\CFapps\mvcframework\uploads</fileuploaddirectory>
  <!-- This is where you set your upload nameconflict value (OVERWRITE,MAKEUNIQUE)-->
  <fileconflict>OVERWRITE</fileconflict>
  <!-- These variables are required for the application framework-->
  <!-- These variables are paths to the components -->
  <basecomponentpath>mvcframework.system.cfcomponents</basecomponentpath>
  <cfcomponents>/mvcframework/system/cfcomponents/</cfcomponents>
  <!-- These variables are paths to the application -->
  <physicalpath>D:\apps\CFapps\mvcframework</physicalpath>
  <virtualpath>/mvcframework/</virtualpath>
  <!-- This is where you would store UDF's -->
  <cflibs>/mvcframework/system/cflibs/</cflibs>
  <!-- This is where you would store customtags -->
  <cftags>/mvcframework/system/cftags/</cftags>
  <!-- This is where you would store layout pagelets -->
  <cfpagelets>/mvcframework/system/cfpagelets/</cfpagelets>
  <!-- This is where you would store cached files -->
  <cfcache>/mvcframework/system/cfcache/</cfcache>
  <!-- This is where you would store cascading stylesheets -->
  <css>/mvcframework/system/css/</css>
  <!-- This is where you would store javascript files -->
  <js>/mvcframework/system/js/</js>
  <!-- Optional (list) - Variables to be excluded from being passed out of a callback on the URL query string-->
  <urlexcludes>APPSECINITIALIZE,LOGOUT,INSTANCEINITIALIZE</urlexcludes>

  <!-- These are application specific variables-->
  <appsecdsn>CF_APP_SEC</appsecdsn>
  <dsn>CF_CONNECTOR_DSN</dsn>
</globals>
```

XML configuration files

All of the application/session configuration parameters are kept in the cfapp-config.xml file which allows an application to be moved to another server and configured without changing any code. The top portion of the file is reserved for Tardis specific configuration parameters, the lower portion is reserved for application specific parameters.

Tardis Framework

In addition, each of the tardis components depends on XML configuration files and are initialized with the `init()` method during object creation.

The configuration file is initialized in application cfm using the `<cf_globals>` custom tag.

```
<cf_globals scope="APPLICATION" filename="cfapp-config.xml"
            configDirectory="d:/apps/system/config/"
            bRefresh="FALSE">
```

`<cf_globals>` Arguments

scope	APPLICATION SESSION (or both with another use of the tag).
filename	Name of the configuration file.
configDirectory	Directory path to the configuration file.
bCopyToRequest	TRUE FALSE, if true copies to the request scope.
bRefresh	TRUE FALSE Re-initializes configuration.

Dump of the application scope:

Tardis Framework

application - struct																																											
applicationname	MVC_Framework																																										
bisappinitialized	TRUE																																										
frontcontroller	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>userManager</td> <td> <table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>addusererror</td> <td>/mvcframework/dsp_error.cfm</td> </tr> <tr> <td>addusersuccess</td> <td>/mvcframework/dsp_success.cfm</td> </tr> <tr> <td>componentpath</td> <td>mvcframework.system.cfcomponents.com.user.userManager</td> </tr> <tr> <td>updateusererror</td> <td>/mvcframework/dsp_error.cfm</td> </tr> <tr> <td>updateusersuccess</td> <td>/mvcframework/dsp_success.cfm</td> </tr> <tr> <td>virtualpath</td> <td>/mvcframework/system/cfcomponents/com/user/</td> </tr> </tbody> </table> </td> </tr> </tbody> </table>	struct		userManager	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>addusererror</td> <td>/mvcframework/dsp_error.cfm</td> </tr> <tr> <td>addusersuccess</td> <td>/mvcframework/dsp_success.cfm</td> </tr> <tr> <td>componentpath</td> <td>mvcframework.system.cfcomponents.com.user.userManager</td> </tr> <tr> <td>updateusererror</td> <td>/mvcframework/dsp_error.cfm</td> </tr> <tr> <td>updateusersuccess</td> <td>/mvcframework/dsp_success.cfm</td> </tr> <tr> <td>virtualpath</td> <td>/mvcframework/system/cfcomponents/com/user/</td> </tr> </tbody> </table>	struct		addusererror	/mvcframework/dsp_error.cfm	addusersuccess	/mvcframework/dsp_success.cfm	componentpath	mvcframework.system.cfcomponents.com.user.userManager	updateusererror	/mvcframework/dsp_error.cfm	updateusersuccess	/mvcframework/dsp_success.cfm	virtualpath	/mvcframework/system/cfcomponents/com/user/																								
struct																																											
userManager	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>addusererror</td> <td>/mvcframework/dsp_error.cfm</td> </tr> <tr> <td>addusersuccess</td> <td>/mvcframework/dsp_success.cfm</td> </tr> <tr> <td>componentpath</td> <td>mvcframework.system.cfcomponents.com.user.userManager</td> </tr> <tr> <td>updateusererror</td> <td>/mvcframework/dsp_error.cfm</td> </tr> <tr> <td>updateusersuccess</td> <td>/mvcframework/dsp_success.cfm</td> </tr> <tr> <td>virtualpath</td> <td>/mvcframework/system/cfcomponents/com/user/</td> </tr> </tbody> </table>	struct		addusererror	/mvcframework/dsp_error.cfm	addusersuccess	/mvcframework/dsp_success.cfm	componentpath	mvcframework.system.cfcomponents.com.user.userManager	updateusererror	/mvcframework/dsp_error.cfm	updateusersuccess	/mvcframework/dsp_success.cfm	virtualpath	/mvcframework/system/cfcomponents/com/user/																												
struct																																											
addusererror	/mvcframework/dsp_error.cfm																																										
addusersuccess	/mvcframework/dsp_success.cfm																																										
componentpath	mvcframework.system.cfcomponents.com.user.userManager																																										
updateusererror	/mvcframework/dsp_error.cfm																																										
updateusersuccess	/mvcframework/dsp_success.cfm																																										
virtualpath	/mvcframework/system/cfcomponents/com/user/																																										
frontcontroller_scopelock	false																																										
globals	<table border="1"> <thead> <tr> <th colspan="2">struct</th> </tr> </thead> <tbody> <tr> <td>appsecdsn</td> <td>CF_APP_SEC</td> </tr> <tr> <td>basecomponentpath</td> <td>mvcframework.system.cfcomponents</td> </tr> <tr> <td>cfcache</td> <td>/mvcframework/system/cfcache/</td> </tr> <tr> <td>cfcomponents</td> <td>/mvcframework/system/cfcomponents/</td> </tr> <tr> <td>cflibs</td> <td>/mvcframework/system/cflibs/</td> </tr> <tr> <td>cfpagelets</td> <td>/mvcframework/system/cfpagelets/</td> </tr> <tr> <td>cftags</td> <td>/mvcframework/system/cftags/</td> </tr> <tr> <td>controllercomponentpath</td> <td>mvcframework.system.cfcomponents.com.controller</td> </tr> <tr> <td>controllervirtualpath</td> <td>/mvcframework/system/cfcomponents/com/controller/</td> </tr> <tr> <td>css</td> <td>/mvcframework/system/css/</td> </tr> <tr> <td>datastorecomponentpath</td> <td>mvcframework.system.cfcomponents.com.datastore.datastore</td> </tr> <tr> <td>datastoremaxerrorsstored</td> <td>20</td> </tr> <tr> <td>datastorevirtualpath</td> <td>/mvcframework/system/cfcomponents/com/datastore/</td> </tr> <tr> <td>dsn</td> <td>CF_CONNECTOR_DSN</td> </tr> <tr> <td>fileconflict</td> <td>OVERWRITE</td> </tr> <tr> <td>fileuploaddirectory</td> <td>D:\apps\CFapps\mvcframework\uploads\</td> </tr> <tr> <td>js</td> <td>/mvcframework/system/js/</td> </tr> <tr> <td>physicalpath</td> <td>D:\apps\CFapps\mvcframework</td> </tr> <tr> <td>urlexcludes</td> <td>APPSECINITIALIZE,LOGOUT,INSTANCEINITIALIZE</td> </tr> <tr> <td>virtualpath</td> <td>/mvcframework/</td> </tr> </tbody> </table>	struct		appsecdsn	CF_APP_SEC	basecomponentpath	mvcframework.system.cfcomponents	cfcache	/mvcframework/system/cfcache/	cfcomponents	/mvcframework/system/cfcomponents/	cflibs	/mvcframework/system/cflibs/	cfpagelets	/mvcframework/system/cfpagelets/	cftags	/mvcframework/system/cftags/	controllercomponentpath	mvcframework.system.cfcomponents.com.controller	controllervirtualpath	/mvcframework/system/cfcomponents/com/controller/	css	/mvcframework/system/css/	datastorecomponentpath	mvcframework.system.cfcomponents.com.datastore.datastore	datastoremaxerrorsstored	20	datastorevirtualpath	/mvcframework/system/cfcomponents/com/datastore/	dsn	CF_CONNECTOR_DSN	fileconflict	OVERWRITE	fileuploaddirectory	D:\apps\CFapps\mvcframework\uploads\	js	/mvcframework/system/js/	physicalpath	D:\apps\CFapps\mvcframework	urlexcludes	APPSECINITIALIZE,LOGOUT,INSTANCEINITIALIZE	virtualpath	/mvcframework/
struct																																											
appsecdsn	CF_APP_SEC																																										
basecomponentpath	mvcframework.system.cfcomponents																																										
cfcache	/mvcframework/system/cfcache/																																										
cfcomponents	/mvcframework/system/cfcomponents/																																										
cflibs	/mvcframework/system/cflibs/																																										
cfpagelets	/mvcframework/system/cfpagelets/																																										
cftags	/mvcframework/system/cftags/																																										
controllercomponentpath	mvcframework.system.cfcomponents.com.controller																																										
controllervirtualpath	/mvcframework/system/cfcomponents/com/controller/																																										
css	/mvcframework/system/css/																																										
datastorecomponentpath	mvcframework.system.cfcomponents.com.datastore.datastore																																										
datastoremaxerrorsstored	20																																										
datastorevirtualpath	/mvcframework/system/cfcomponents/com/datastore/																																										
dsn	CF_CONNECTOR_DSN																																										
fileconflict	OVERWRITE																																										
fileuploaddirectory	D:\apps\CFapps\mvcframework\uploads\																																										
js	/mvcframework/system/js/																																										
physicalpath	D:\apps\CFapps\mvcframework																																										
urlexcludes	APPSECINITIALIZE,LOGOUT,INSTANCEINITIALIZE																																										
virtualpath	/mvcframework/																																										

Example of UDF library include using configuration parameter:

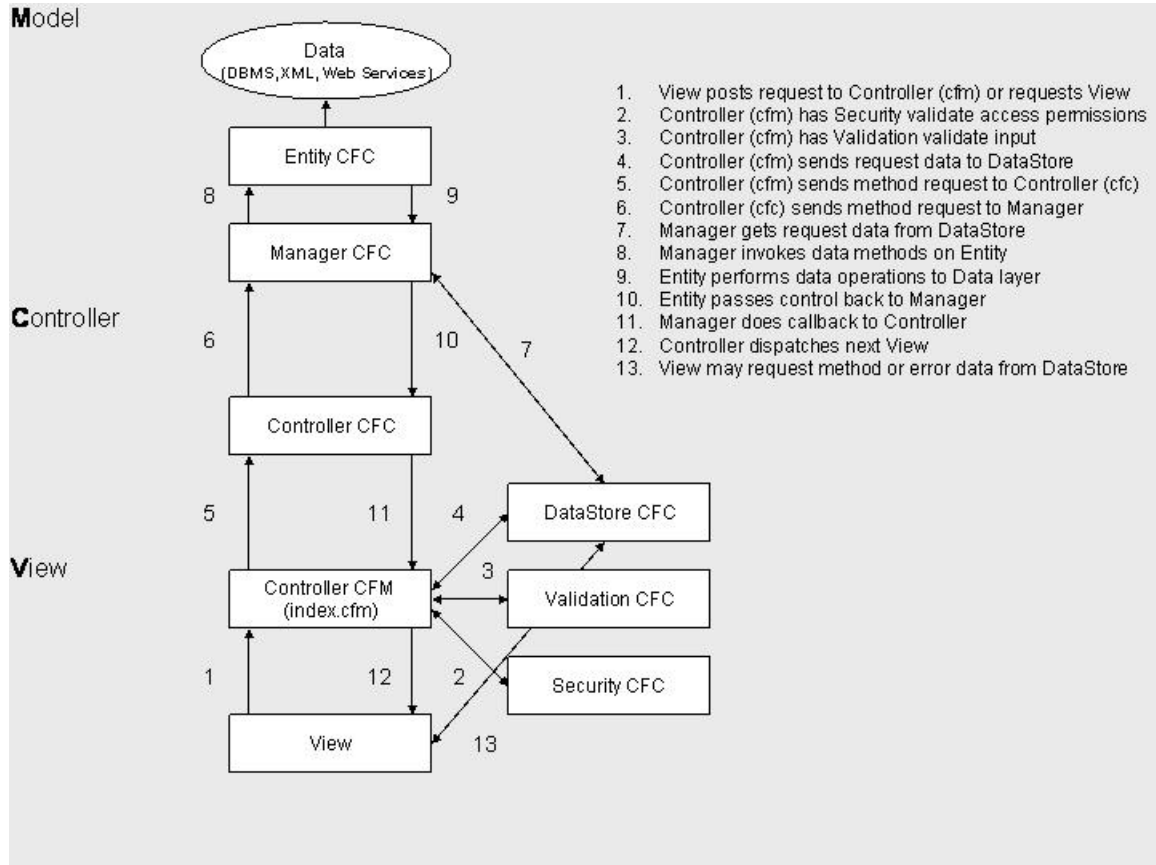
```
<cfinclude template="#application.globals.cflibs#isEmail.cfm">
```

Example of UDF library include using configuration parameter if bCopyToRequest is true in <cf_globals>

```
<cfinclude template="#request.application.cflibs#isEmail.cfm">
```

Tardis Framework

Functional Overview



Refer to the `tardis_sampleapp` for examples of application implementation.