Getting Started with ColdFusion®

MAX 2009 Lab

Josh Adams

Before you proceed, please note the following: throughout this document, the following styles are used to simplify instructions. **Menu items** and **field prompts** are formatted in **bold**. `Text you need to enter` is always in `monotype`. And any special notes are in boxes like this.

1: Setup and Configuration

This hands-on lab assumes that you have the following installed and running:

- Microsoft IIS
- Adobe® ColdFusion® Builder™ beta
- Adobe® ColdFusion® 9 Developer Edition installed in multi-server configuration and configured to use Microsoft IIS as the web server

Note that although the instructions in this lab are specific to the configuration noted above, the ColdFusion files you create write in this lab will run on any ColdFusion 9 installation. If you choose to work through these exercises with a different configuration, certain changes may be required to specifics regarding file and directory locations.

You can find more information and links to download these in the following locations:

- ColdFusion 9: http://www.adobe.com/support/coldfusion/downloads.html
- ColdFusion Builder beta: http://labs.adobe.com/technologies/coldfusionbuilder

2: Lab 1 – HelloWorld

This lab demonstrates how to create a ColdFusion project in ColdFusion Builder as well as how to configure that project to use a ColdFusion server. Additionally, this lab demonstrates how to create a simple ColdFusion application that dynamically generates output.

In this lab you will learn about the following:

- ColdFusion Builder projects
- ColdFusion pages
- ColdFusion tags
- ColdFusion functions
- The special significance of the # character for ColdFusion

Create New Project

To create a new project, perform the following steps:

1. If you have not yet opened ColdFusion Builder, please do so now.
2. Close any open projects in ColdFusion Builder.
3. To be certain you are in the ColdFusion perspective, open the **Window** menu and select **Perspective** and then **Other** to display the **Open Perspective** dialog box.
4. Choose **ColdFusion** and click the **OK** button.
5. Open the **File** menu and select **New** and then **ColdFusion Project** to display the **New ColdFusion Project** dialog.
6. Fill in the following fields:
    - **Project Name*** = `HelloWorld`
    - **Use Default Location** checkbox = unchecked
7. Click the **Browse** button, then expand **My Computer**, then expand **Local Disk (C:)**, then expand **Inetpub**, and then select **wwwroot**.
8. Click the **Make New Folder** button, then type `HelloWorld` followed by the Enter key, and then click the OK button.

9. In the **New ColdFusion Project** dialog you should now see the following:
    - **Project Name*** = `HelloWorld`
    - **Use Default Location** checkbox = unchecked
    - **Project Location*** = `C:\Inetpub\wwwroot\HelloWorld`
10. Click the **Next** button.
11. For **Server** in the **Select Server** area, choose **Add Server...** to display the **ColdFusion Server Setup** dialog.
12. Fill in the following fields:
    - **Server Name*** = `ColdFusion 9`
    - **Description** = `ColdFusion 9`
    - **Host Name*** = `localhost`
    - **Is Local** radio button = selected
    - **WebServer Port*** = `80`
    - **Context Root**** = leave this setting blank
    - **Application Server Name**** = `cfusion`
    - **RDS User Name** = `admin`
    - **RDS Password** = `admin`
13. Click the **Next** button.
14. Fill in the following fields:
    - **Server Home*** = `C:\JRun4`
    - **Document Root** = `C:\InetPub\wwwroot`
    - **Version*** = `U7`
15. Click the **Next** button.
16. On the next screen, click the **Finish** button to create and open the new project.

Remember the steps used here, they will be repeated for each project you'll create. However, the exact steps will not be repeated, and just the unique values for the specific project will be provided. You can refer back to these steps as needed.

Create a Page for Your Application

To create a new page for your application, perform the following steps:

1. Open the **File** menu and select **New** and then **ColdFusion Page** to display the **New ColdFusion Page** dialog.

2. Fill in the following fields:

    - **Name*** = `index.cfm`

3. Click the **Finish** button to create and open the new page.

> Remember the steps used here, they will be repeated for each project you'll create. However, the exact steps will not be repeated, and just the unique values for the specific project will be provided. You can refer back to these steps as needed.

Enter Your Code

The **index.cfm** file will be blank. Enter the following code:

```
Hello, World! <cfoutput>#Now()#</cfoutput>
```

> Note that in order to focus on the most essential code, this example, as well as many examples in this lab, does not use properly-formed HTML. Browser output should always be proper-formed HTML in production applications.

Test Your Code

Save your changes to **index.cfm** and run your code to test it. The expected result is **Hello, World!** as the page output. You can run your code by pressing the "play" button in the menu at the top of ColdFusion Builder. Each time you refresh the view in the browser you will see the time output change. This happens because each time the page is requested by the browser, ColdFusion dynamically generates the time value and inserts it into the output.

3: Lab 2 – Basic User Interaction

This lab demonstrates how to use ColdFusion for basic user interaction. One of the most common tasks for a ColdFusion application is interacting with a user by accepting user input and dynamically generating user-specific output.

In this lab you will learn about the following:

- ColdFusion conditional statements
- ColdFusion variables and variable scopes

Create New Project

Create a new project for this lab:

1. Close any open projects in ColdFusion Builder.
2. Create a new ColdFusion Builder project (using the same steps as used previously) in which you specify the following:
    - **Project Name*** = `HelloWhoever`
    - **Use Default Location** checkbox = unchecked
    - **Project Location*** = `C:\Inetpub\wwwroot\HelloWhoever`
    - **Servers** = `ColdFusion 9`

Note that because you defined your ColdFusion server in the last lab, you can simply select it for this project.

Create a Page for Your Application

Create the **index.cfm** page for your application:

1. Create a new page (using the same steps as used previously) for which you specify the following:
    - **Name*** = `index.cfm`

Enter Your Code

The **index.cfm** file will be blank. Enter the following code:

```
<form action="index.cfm" method="post">
     Your Name: <input name="myName" type="text">
     <input name="submitButton" type="submit" value="Submit">
</form>

<cfif IsDefined("Form.myName")>
     Hello, <cfoutput>#Form.myName#</cfoutput>!
</cfif>
```

Test Your Code

Save your changes to **index.cfm** and run your code to test it. Enter some text into the **Your Name:** text input then press the Submit button. You will see the page refresh and under the form you will see **Hello,** followed by the text you entered into the **Your Name:** text input.

4: Lab 3 – Working with a Database

This lab demonstrates how to use ColdFusion for interaction with a database. One of the most common tasks in ColdFusion is working with a database to read, create, update, and delete data. ColdFusion 9 offers multiple ways to interact with a database; in this lab, you will use the most traditional method.

In this lab you will learn about the following:

- Parameterized SQL queries in ColdFusion
- ColdFusion and Derby databases
- Sample databases included with ColdFusion

Create New Project

Create a new project for this lab:

1. Close any open projects in ColdFusion Builder.
2. Create a new ColdFusion Builder project (using the same steps as used previously) in which you specify the following:
    - **Project Name*** = `MyArtists`
    - **Use Default Location** checkbox = unchecked
    - **Project Location*** = `C:\Inetpub\wwwroot\MyArtists`
    - **Servers** = `ColdFusion 9`

Create a Page for Your Application

Create the **index.cfm** page for your application:

1. Create a new page (using the same steps as used previously) for which you specify the following:
    - **Name*** = `index.cfm`

Enter Your Code

The **index.cfm** file will be blank. Enter the following code:

```
<cfif IsDefined("Form.submitButton")>
     <cfquery datasource = "cfartgallery">
          <cfif Form.submitButton EQ "Add Artist">
               INSERT INTO Artists
               (
               FIRSTNAME, LASTNAME, EMAIL, PHONE, FAX,
               ADDRESS, CITY, STATE, POSTALCODE, THEPASSWORD
               )
               VALUES
               (
               <cfqueryparam value="#Left(FORM.artistFirstName, 20)#"
                    cfsqltype="cf_sql_varchar" maxlength="20">,
               <cfqueryparam value="#Left(FORM.artistLastName, 20)#"
                    cfsqltype="cf_sql_varchar" maxlength="20">,
               <cfqueryparam value="#Left(FORM.artistEmail, 50)#"
                    cfsqltype="cf_sql_varchar" maxlength="50">,
               <cfqueryparam value="#Left(FORM.artistPhone, 20)#"
                    cfsqltype="cf_sql_varchar" maxlength="20">,
               <cfqueryparam value="#Left(FORM.artistFax, 12)#"
                    cfsqltype="cf_sql_varchar" maxlength="12">,
               <cfqueryparam value="#Left(FORM.artistAddress, 50)#"
                    cfsqltype="cf_sql_varchar" maxlength="50">,
               <cfqueryparam value="#Left(FORM.artistCity, 20)#"
                    cfsqltype="cf_sql_varchar" maxlength="20">,
               <cfqueryparam value="#Left(FORM.artistState, 2)#"
                    cfsqltype="cf_sql_varchar" maxlength="2">,
               <cfqueryparam value="#Left(FORM.artistZIP,10)#"
                    cfsqltype="cf_sql_varchar" maxlength="10">,
               <cfqueryparam value="#Left(FORM.artistPassword, 20)#"
                    cfsqltype="cf_sql_varchar" maxlength="8">
               )
```

```
            <cfelseif Form.submitButton EQ "Update Artist">
                  UPDATE Artists SET
                  FIRSTNAME =
                  <cfqueryparam value="#Left(FORM.artistFirstName, 20)#"
                        cfsqltype="cf_sql_varchar" maxlength="20">,
                  LASTNAME =
                  <cfqueryparam value="#Left(FORM.artistLastName, 20)#"
                        cfsqltype="cf_sql_varchar" maxlength="20">,
                  EMAIL =
                  <cfqueryparam value="#Left(FORM.artistEmail, 50)#"
                        cfsqltype="cf_sql_varchar" maxlength="50">,
                  PHONE =
                  <cfqueryparam value="#Left(FORM.artistPhone, 20)#"
                        cfsqltype="cf_sql_varchar" maxlength="20">,
                  FAX = <cfqueryparam value="#Left(FORM.artistFax, 12)#"
                        cfsqltype="cf_sql_varchar" maxlength="12">,
                  ADDRESS =
                  <cfqueryparam value="#Left(FORM.artistAddress, 50)#"
                        cfsqltype="cf_sql_varchar" maxlength="50">,
                  CITY =
                  <cfqueryparam value="#Left(FORM.artistCity, 20)#"
                        cfsqltype="cf_sql_varchar" maxlength="20">,
                  STATE =
                  <cfqueryparam value="#Left(FORM.artistState, 2)#"
                        cfsqltype="cf_sql_varchar" maxlength="2">,
                  POSTALCODE =
                  <cfqueryparam value="#Left(FORM.artistZIP,10)#"
                        cfsqltype="cf_sql_varchar" maxlength="10">,
                  THEPASSWORD =
                  <cfqueryparam value="#Left(FORM.artistPassword, 20)#"
                        cfsqltype="cf_sql_varchar" maxlength="8">
                  WHERE ARTISTID = <cfqueryparam value="#FORM.artistID#"
                        cfsqltype="cf_sql_integer">
            <cfelseif Form.submitButton EQ "Delete Artist">
                  DELETE FROM Artists
                  WHERE ARTISTID = <cfqueryparam value="#FORM.artistID#"
                        cfsqltype="cf_sql_integer">
            </cfif>
      </cfquery>
</cfif>
```

```
<form action="index.cfm" method="post">
     Artist ID: <input name="artistID" type="text">
     (required for Update and Delete; ignored for Add)<br>
     Artist First Name: <input name="artistFirstName" type="text"><br>
     Artist Last Name: <input name="artistLastName" type="text"><br>
     Artist Email: <input name="artistEmail" type="text"><br>
     Artist Phone: <input name="artistPhone" type="text"><br>
     Artist Fax: <input name="artistFax" type="text"><br>
     Artist Address: <input name="artistAddress" type="text"><br>
     Artist City: <input name="artistCity" type="text"><br>
     Artist State: <input name="artistState" type="text"><br>
     Artist ZIP: <input name="artistZIP" type="text"><br>
     Artist Password: <input name="artistPassword" type="text"><br>
     <input name="submitButton" type="submit" value="Add Artist">
     <input name="submitButton" type="submit" value="Update Artist">
     <input name="submitButton" type="submit" value="Delete Artist">
</form>

<cfquery name="Variables.artists" datasource = "cfartgallery">
     SELECT * FROM Artists
</cfquery>

<cfdump var="#Variables.artists#">
```

Test Your Code

Save your changes to **index.cfm** and run your code to test it. Each time you run it, you should see your form at the top of the page and a dump of the data from the **Artists** table at the bottom of the page. Fill out the form completely and submit it; you should see the data you entered appear as the last row of data in the dumped recordset from the **Artists** table.

5: Lab 4 – Working with User Interface (UI) Controls

This lab demonstrates how to use some of ColdFusion's built-in UI controls. ColdFusion runs on a server and provides either data or UI or both to the client; in this lab, ColdFusion will provide both the UI and the data: you will use ColdFusion's powerful UI component generation features to provide a compelling interface for displaying data to users.

In this lab you will learn about the following:

- ColdFusion Ajax UI controls
- ColdFusion data binding
- ColdFusion Components (CFCs)
- CFC methods
- Application.cfc

Create New Project

Create a new project for this lab:

1. Close any open projects in ColdFusion Builder.
2. Create a new ColdFusion Builder project (using the same steps as used previously) in which you specify the following:
    - **Project Name*** = `PrettyArt`
    - **Use Default Location** checkbox = unchecked
    - **Project Location*** = `C:\Inetpub\wwwroot\PrettyArt`
    - **Servers** = `ColdFusion 9`

Create Application.cfc for Your Application

To create **Application.cfc** for your application, perform the following steps:

1. Open the **File** menu and select **New** and then **ColdFusion Page** to display the **New ColdFusion Component** dialog.
2. Fill in the following fields:
    - **Component Name*** = `Application`
3. Click the **Finish** button to create and open the new page.

The **Application.cfc** file will contain the following code:

```
<cfcomponent>

</cfcomponent>
```

Enter Your Code

Enter the following code in the blank line between the opening **<cfcomponent>** tag and the closing **</cfcomponent>** tag:

```
     <cfset this.datasource = "cfartgallery">
```

Create a Page for Your Application

Create the **index.cfm** page for your application:

1. Create a new page (using the same steps as used previously) for which you specify the following:
    - **Name*** = `index.cfm`

Enter Your Code

The **index.cfm** file will be blank. Enter the following code:

```
<cfform name="artForm">
     <cfgrid name="artGrid" format="html" insert="false"
          delete="false" selectmode="edit"
          bind="cfc:prettyArt.art.getAllArt({cfgridpage},
          {cfgridpagesize}, {cfgridsortcolumn}, {cfgridsortdirection})"
          onChange="cfc:prettyArt.art.changeArt({cfgridaction},
          {cfgridrow}, {cfgridchanged})">

          <cfgridcolumn name="artName" header="Name" width="200">
          <cfgridcolumn name="description" header="Description"
               width="400">
          <cfgridcolumn name="price" header="Price" width="100">
          <cfgridcolumn name="isSold" header="Sold" width="50">
     </cfgrid>
</cfform>
```

You will note that this code references a CFC named Art. You will now create this CFC.

Create a CFC for Your Application

Create **Art.cfc** for your application:

1. Create a new CFC (using the same steps as used previously) for which you specify the following:
    - **Component Name*** = `Art`

Enter Your Code

Enter the following code in the blank line between the opening **<cfcomponent>** tag and the closing **</cfcomponent>** tag:

```
<cffunction name="getAllArt" access="remote" returntype="struct">
     <cfargument name="page" type="numeric" required="true">
     <cfargument name="pageSize" type="numeric" required="true">
     <cfargument name="sortColumn" type="string" required="true">
     <cfargument name="sortDirection" type="string" required="true">

     <cfquery name="Local.artQuery">
          SELECT * FROM Art
     </cfquery>

     <cfset Local.returnVal = queryConvertForGrid(Local.artQuery,
          Arguments.page, Arguments.pageSize)>

     <cfreturn Local.returnVal>
</cffunction>

<cffunction name="changeArt" access="remote" returntype="boolean">
     <cfargument name="action" type="string" required="true">
     <cfargument name="row" type="struct" required="true">
     <cfargument name="changed" type="struct" required="true">

     <cfset Local.lCount = 0>
     <cfset Local.sCount = StructCount(Arguments.changed)>

     <cfif Arguments.action EQ "U">
          <cfquery>
               UPDATE Art SET
               <cfloop item="Local.key"
                    collection="#Arguments.changed#">

                    <cfset Local.lCount = Local.lCount + 1>

                    #Local.key# = <cfqueryparam
                    value="#Arguments.changed[Local.key]#">

                    <cfif Local.lCount NEQ Local.sCount>
                         ,
                    </cfif>
               </cfloop>
               WHERE ArtID = <cfqueryparam
                    value="#Arguments.row.ArtID#"
                    cfsqltype="cf_sql_integer">
          </cfquery>
     </cfif>

     <cfreturn True>
</cffunction>
```

Test Your Code

Save your changes to **index.cfm** and run your code to test it. When the grid loads, click on the navigation controls in the bar at the bottom of the grid; you will see the data in the grid change. Click on a value in one of the columns; you will notice that the field changes to become editable. Make a change to the value in the field and then click on another field; you will notice a red indicator appears in the upper left-hand corner of the field you changed. Press the refresh button the bottom navigation row to retrieve the data for the grid from the server; you will see that the field you changed still has the value to which you changed it. Try editing a value in the **Sold** column to something other than 0 or 1. What happens? Why?

This code can be enhanced to allow for the creation of new records as well as the deletion of existing records.

6: Lab 5 – Working with Presentations

This lab demonstrates how to use some of ColdFusion's features for working with presentations. ColdFusion contains powerful features for working with numerous file and document types, including presentations, that allow you to create compelling applications quickly.

In this lab you will learn about the following:

- ColdFusion's presentation features
- ColdFusion's features for working with files and directories

Create New Project

Create a new project for this lab:

1. Close any open projects in ColdFusion Builder.
2. Create a new ColdFusion Builder project (using the same steps as used previously) in which you specify the following:
    - **Project Name*** = `Presentastic`
    - **Use Default Location** checkbox = unchecked
    - **Project Location*** = `C:\Inetpub\wwwroot\Presentastic`
    - **Servers** = `ColdFusion 9`

Create a Page for Your Application

Create the **index.cfm** page for your application:

2. Create a new page (using the same steps as used previously) for which you specify the following:
    - **Name*** = `index.cfm`

Enter Your Code

The **index.cfm** file will be blank. Enter the following code:

```
<!---Set up--->
<cfset Variables.tDir = getDirectoryFromPath(GetCurrentTemplatePath())>
<cfset Variables.mainDir = "presentations">
<cfset Variables.pSubDir = Variables.mainDir & "\" & "ppt" & "\">
<cfset Variables.hSubDir = Variables.mainDir & "\" & "html" & "\">
<cfset Variables.cSubDir = Variables.mainDir & "\" & "connect" & "\">
<cfset Variables.pDir = Variables.tDir & Variables.pSubDir & "\">
<cfset Variables.hDir = Variables.tDir & Variables.hSubDir & "\">
<cfset Variables.cDir = Variables.tDir & Variables.cSubDir & "\">
<cfset Variables.pWebDir = replace(Variables.pSubDir, "\", "/", "all")>
<cfset Variables.hWebDir = replace(Variables.hSubDir, "\", "/", "all")>
<cfset Variables.cWebDir = replace(Variables.cSubDir, "\", "/", "all")>

<cfif isDefined("Form.submitButton") AND isDefined("Form.pFile")>
     <!---Handle uploaded presentations and do conversions here--->
     <!---Create directory for PPT file--->
     <cfif NOT directoryExists(Variables.pDir)>
          <cfdirectory action="create" directory="#Variables.pDir#">
     </cfif>

     <!---Upload file--->
     <cffile action="upload"
          fileField="Form.pFile" accept="application/vnd.ms-powerpoint"
          destination="#Variables.pDir#" nameConflict="overwrite">

     <!---Create directory for HTML version of PPT file--->
     <cfif NOT directoryExists(Variables.hDir & cffile.ClientFileName)>
          <cfdirectory action="create"
               directory="#Variables.hDir##cffile.ClientFileName#">
     </cfif>

     <!---Create HTML version of PPT file--->
     <cfpresentation format="html" overwrite="true"
          directory="#Variables.hDir##cffile.ClientFileName#">
          <cfpresentationslide
               src="#cffile.ServerDirectory#\#cffile.ClientFile#" />
     </cfpresentation>

     <!---Create directory for Connect version of PPT file--->
     <cfif NOT directoryExists(Variables.cDir & cffile.ClientFileName)>
          <cfdirectory action="create"
               directory="#Variables.cDir##cffile.ClientFileName#">
     </cfif>

     <!---Create Connect version of PPT file--->
     <cfpresentation title="#cffile.ClientFileName#"
          directory="#Variables.cDir##cffile.ClientFileName#"
          overwrite="true">

          <cfpresentationslide
               src="#cffile.ServerDirectory#\#cffile.ClientFile#" />
     </cfpresentation>

     <!---Redirect back to this page as a GET instead of a POST--->
     <cflocation url="#GetFileFromPath(CGI.Script_Name)#" addtoken="no">
</cfif>
```

This code is of course only half of the picture!

Enter Your Code

Enter the following code at the end of the **index.cfm** file:

```
<!---Get a list of all presentations for output--->
<cfdirectory action="list"
      directory="#Variables.hDir#"
      name="Variables.pQuery" type="dir">

<html>
<head><title>Presentastic</title></head>
<body>
<!---Display goes here--->
<h2>Welcome to Presentastic, Basic Edition!</h2>
<form name="pForm" method="post" enctype="multipart/form-data"
      action="<cfoutput>#GetFileFromPath(CGI.Script_Name)#</cfoutput>">
      Upload a presentation: <input type="file" name="pFile">
      <input type="submit" name="submitButton" value="Submit">
</form>
<table>
      <tr>

      <cfoutput query="Variables.pQuery">
      <cfif NOT Variables.pQuery.CurrentRow MOD 5>
            </tr>
            <tr>
      </cfif>
      <td>
      <table>
            <tr><td align="center">#Variables.pQuery.Name#</td></tr>
            <tr>
            <td align="center">
            <!---
            Note that in due to a planned change in ColdFusion 9,
            you may need to use index.htm instead of index.html below
            --->
            <a
href="#Variables.hWebDir##Variables.pQuery.Name#/index.html">HTML</a> |
            <a
href="#Variables.cWebDir##Variables.pQuery.Name#/index.htm">Connect</a>
            </td>
            </tr>
            <tr>
            <td align="center">
            <a
href="#Variables.pWebDir##Variables.pQuery.Name#.ppt">PowerPoint</a>
            </td>
            </tr>
      </table>
      </td>
      </cfoutput>

      </tr>
</table>

</body>
</html>
```

Test Your Code

Save your changes to **index.cfm** and run your code to test it. If you haven't already done so, open Microsoft PowerPoint and create a PPT file (make sure to create a PPT file and not a PPTX). Use the **Upload a presentation** control to upload your presentation. When the page reloads, you will see the name of your PowerPoint file and below it you will see three links: HTML, Connect, and PowerPoint. Click on each of these. What happens?

ColdFusion can do even more with PowerPoint files! If you install OpenOffice on the ColdFusion server, ColdFusion can also convert your PPT file to a PDF. And because of ColdFusion's ability to create thumbnail images from PDFs, once you have converted the PPT file to a PDF, you can create a thumbnail image of the PDF. But using these capabilities, you can enhance your Presentastic application so that it displays a thumbnail of the 1st page of the presentation and gives you a link to a PDF version of the presentation—and you can do all of this with only a few additional lines of code!