
Unit 2: Getting Started

Unit Objectives

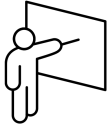
After completing this unit, you should be able to:

- Understand the basic architecture of dynamically generated web sites.
- Understand the features and capabilities of the ColdFusion server.
- Create a project using ColdFusion Builder.
- Use ColdFusion's built-in troubleshooting features to help you debug your code.

Unit Topics

- Reviewing ColdFusion's Features and Capabilities
- Introducing the ColdFusion Administrator
- Working with ColdFusion Builder
- Debugging and Troubleshooting your Apps

Reviewing ColdFusion's Features and Capabilities



The ColdFusion Server is a J2EE application server that is used to dynamically generate content from requests that are typically received via the Hypertext transfer protocol (http).

You'll create applications for the CF server using a combination of the ColdFusion Markup Language (CFML) which syntactically resembles HTML and ColdFusion Script (cfscript) which is similar to Javascript.

The response from a ColdFusion server is typically derived from a variety of inputs and data sources including, but not limited to:

- User input from HTML forms
- User input from PDF forms
- User credentials derived from a single sign-on architecture (LDAP / Active Directory/ Oauth)
- Content on the server's file system or network
- POP Mail Gateways
- Content stored in databases
- Content stored in PDF files
- Content stored in Microsoft Excel files
- Content stored in Microsoft Sharepoint
- Data pulled from other websites / web services.
- WebSocket requests

The ColdFusion server can generate every commonly used file format, including, but not limited to:

- HTML
- Javascript
- CSS
- PDF
- ZIP Files
- Microsoft Word
- JPEG
- PNG
- Microsoft Excel
- JSON / XML

Accessing Your Back Office Resources

ColdFusion is the “glue” that responds to your requests from your web client and accesses back-office resources. Features and capabilities include, but are not limited to the following:

- Enterprise relational databases
- SMTP/POP Mail Gateways
- PDF Generation
- PDF Form Filling & Data Extraction
- Authentication & Authorization
- Logging
- Deploy rich UI widgets
- SOAP web service creation/invocation
- REST web service creation/invocation
- Oauth (CF 11)
- Spreadsheet read/write
- Read/Write Zip Files
- LDAP Integration
- Sharepoint Integration
- Exchange Integration
- Output Caching
- File I/O
- FTP
- Invoke .NET Assemblies
- Invoke Java Servlets
- Output social media widgets (CF 11)

Understanding Dynamic Content Generation

As illustrated by figure 1, the ColdFusion server is “middleware” that receives requests from any http or websocket client and generates a dynamically created response from data that is marshaled from a variety of back-office resources.

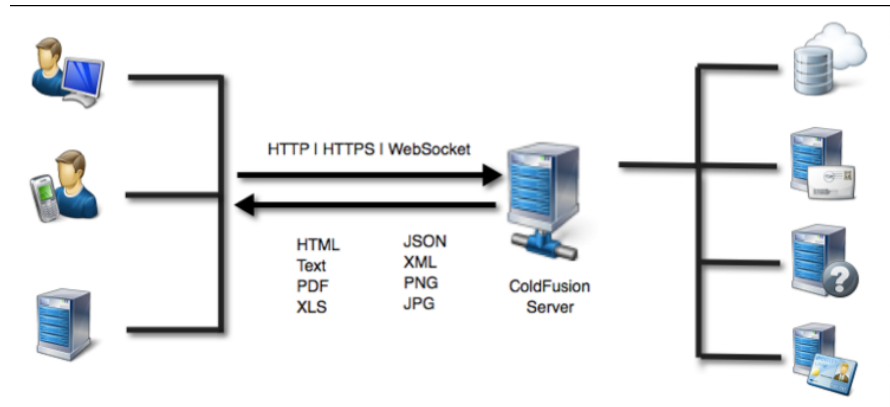


Figure 1: ColdFusion dynamic content generation

HTTP requests are handled by a separate web server process which has been configured to pass requests to the ColdFusion server for processing based on the following requested file name extensions:

File Extension	Description
.cfm	Implies that the generated output will be an HTML page.
.cfml	Implies that the generated output will be an HTML page.
.cfc	Implies that the returned data will be in JSON/XML format. CFC files are typically used for supporting AJAX/SOAP web services and to contain the business logic for fetching the raw data that is output in response to .cfm/.cfml requests.
.cfr	Executes a ColdFusion report file, generated by ColdFusion's Report Builder application.

Therefore, the following http requests would be passed from the web server to the ColdFusion server for processing:

- <http://localhost/index.cfm>
- <http://localhost/displayrecord.cfm?id=25>
- <http://localhost/index.cfc?method=GetSomeData>

Using ColdFusion in a Three-Tiered Architecture

In the pre-smartphone days, organizations could settle on creating a single front-end that supported only a limited number of desktop browsers. With the proliferation of mobile devices of all shapes, sizes, and capabilities, corporate IT must now consider developing multiple front-ends for their apps.

The best practice for building web-based applications is to organize your code into three distinct service layers:

1. The User-Interface
2. The Business Intelligence Tier
3. The Database Tier

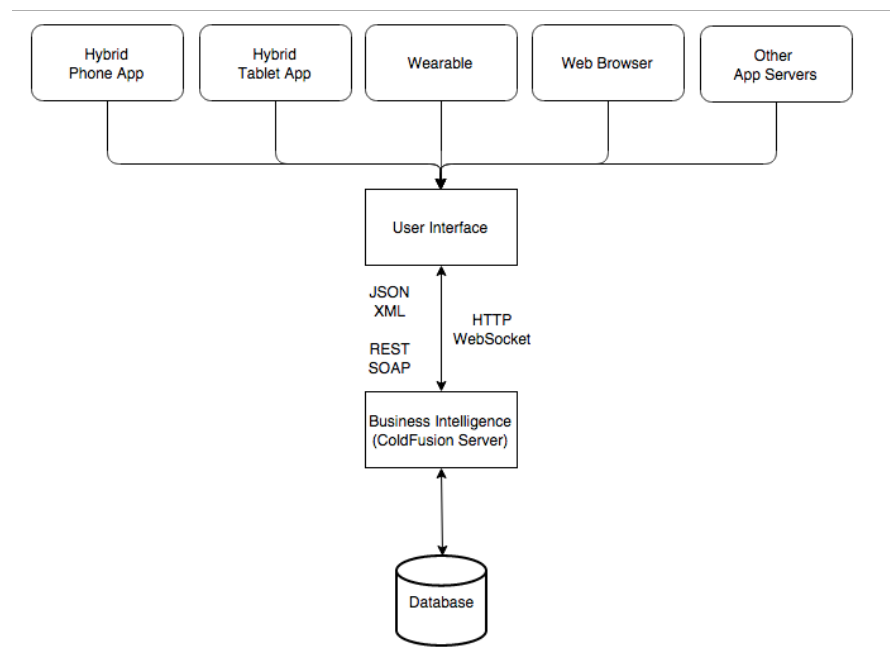


Figure 2: 3-Tiered Architecture

At an absolute minimum, line-of-business apps should support desktop and tablet with reduced functionality available for phones. This requirement frequently requires that multiple front-end apps be developed in parallel, each accessing a common REST-based webservices api implemented at the business intelligence tier.

ColdFusion 11 meets the requirements of the business intelligence tier by making it easy to:

- Connect to enterprise databases
- Implement RESTful web services
- Convert data to/from Javascript object notation (JSON) and XML.
- Partition your codebase into business rules and data handling stored in ColdFusion components (CFC) while handling html markup generation via ColdFusion pages (CFM).

Understanding the Challenges of Dynamically Serving Content

Dynamically serving web pages enables you to:

- Deliver personalized content
- Manage structured content in a database
- Marshal data from back-office servers or the cloud to produce a "mashup"
- Implement roles-based security

However, there are some potential “gotchas” associated with using this architecture that you should constantly evaluate during architecture and code reviews that include:

- Scalability
- Performance
- Robustness
- Security

Building a Scalable Infrastructure

The term “scalability” generally refers to how well your application supports multiple users simultaneously accessing your system. An optimally scalable application will provide a consistently acceptable response time for each concurrent user of your system. While no system is infinitely scalable, however, a well-built architecture’s response times should degrade linearly and predictably in proportion to the number of concurrent users.

Scalability can be measured by performing load testing whereby an automated program simulates concurrent users by running use-case scripts against your application.

Items affecting scalability include, but are not limited to, the following:

- ColdFusion server configuration
- Server hardware
- Database performance
- Server network infrastructure
- Source code efficiency under load

Building a Performant App

The term “performance” generally refers to the end-user's experience with using your application and the time that it takes for them to complete a single task. Multi-page applications that require full-page reloads, forcing the user to wait for interface redraws can be inherently difficult to make performant. Other issues affecting performance involve:

- User Bandwidth
- Server Bandwidth
- Data compression / minification
- Use of GPU intensive CSS transitions and animations
- Using html tables to layout the content of your pages into multiple columns.
- Failure to merge and compress CSS files
- Failure to use sprites and base-64 encoding to reduce the number of http requests needed to load images.

Single-page applications that only make http requests to transmit and receive data via AJAX requests generally tend to be more performant than multi-page apps.

Building a Robust Application

The term “robustness” generally refers to the ease in which an application can be modified to adapt to ever-changing business requirements and the addition of new features. Since labor costs typically represent the vast majority of an application's development budget, robustness considerations are typically the most important part of your decision-making processes. To create a robust app:

- Always comment your code.
- Separate your business logic from your presentation tier by implementing a 3-tiered architecture.
- Use code libraries and well-defined frameworks whenever possible.
- Group related functions into “packages”
- Adopt coding standards and stick to them.
- Participate in code reviews with your peers.
- Implement source control.
- Create a well-documented, automated build/test/deploy process that migrates your code from development to testing to production.

Building a Secure Application Architecture

Security is of paramount importance when developing any application. During this course, we'll discuss a series of features in ColdFusion that you should implement to prevent well-known exploits such as SQL injection and cross-site scripting.

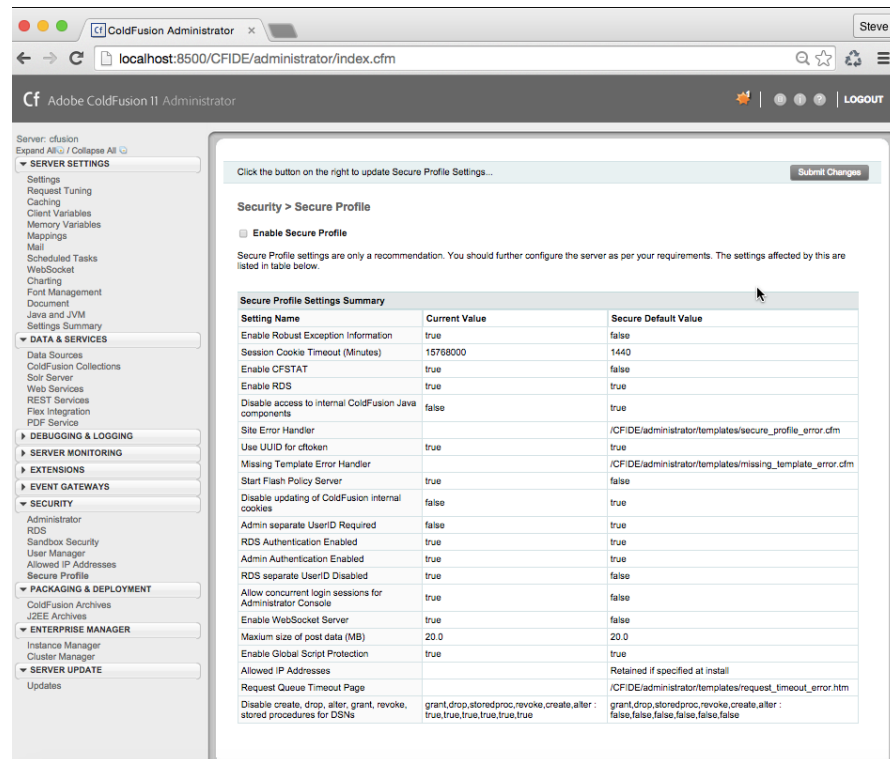


Figure 3: Using the ColdFusion 11 Secure Profile Feature

Other techniques to harden your implementation include, but are not limited to:

- Installing ColdFusion using a secure profile.
- Making the ColdFusion Administrator tool only accessible to users who have direct access to the server console.
- Staying current with CF server updates and operating system updates.
- Disabling ColdFusion features that give developers direct access to the server's file system.
- Encrypting sensitive data in your database using FIPS strong encryption (CF Enterprise)
- Making your app only available via the HTTPS protocol.

Choosing the Correct ColdFusion Version

ColdFusion has five different licenses:

- **Adobe ColdFusion Developer Edition**
This is a free, fully functional version of ColdFusion for local development of applications. The service will only respond to requests from two remote IP addresses.
- **Adobe ColdFusion Express**
Enables you to quickly setup a development or demonstration instance of ColdFusion without running a full installer or following the complete installation procedure. Essentially this is ColdFusion without the Apache Solr full text search engine, Jetty web server, PDF generation service, or Microsoft .NET services connector. These services can be installed later by using the ColdFusion Add On Services Installers, detailed later in this chapter.
- **ColdFusion 11 Standard**
Supports the same features as ColdFusion 11 Enterprise except has limited support for WebSockets, PDF generation, Microsoft Exchange integration, encryption, enterprise databases (ORACLE), J2EE deployment, Multisite REST services, event gateways, execution of scheduled tasks, and server management.
- **ColdFusion 11 Enterprise**
Supports all ColdFusion features and capabilities.
- **Amazon AWS**
ColdFusion is also available as a pay-as-you-go service on Amazon Web Services.

See the ColdFusion buyers guide at <http://adobe.ly/1GkWhz6> for full details regarding the differences between versions.

Installing ColdFusion

Adobe ColdFusion has installers for Windows, OS/X, Linux, and Solaris. However, other installers are also available that enable CF to communicate with other server-side processes, including:

- The **ColdFusion Add On Services Installer** that includes:
 - Apache Solr full-text search engine
 - Jetty web server
 - PDF generation service (Windows and Linux only)
- **PhoneGap Shell App**
This is the recommended approach to quickly test and debug hybrid ios/android applications that are generated by ColdFusion 11.
- **Adobe LiveCycle Data Services**
Typically used to push data in realtime to Adobe Flash/Flex/AIR applications.

- **ColdFusion .NET Integration Services**
Enables your ColdFusion server to execute Microsoft .NET assemblies.
- **ColdFusion 11 Sharepoint Integration WebParts**
Allows ColdFusion applications to be displayed as a Sharepoint webpart.
- **ColdFusion 11 Server Manager WAR files**
Application-server specific WAR files that enable you to run Start and Stop operations from the ColdFusion Server Manager.

Programming with the ColdFusion Markup Language

The ColdFusion Markup Language (CFML), uses an intuitive, case-insensitive, tag-based syntax to interact with server-side resources and dynamically generate content based on user's requests.

The following snippet outputs a web page containing the text “hello world”

```
<html>
<head>
<body>

<cfset out = "Hello World">
<cfoutput>
#variables.out#
</cfoutput>

</body>
</html>
```

Programming with CFSCRIPT

As illustrated by the following code snippet, CFScript resembles Javascript and generally requires less typing than CFML.

```
<html>
<head>
<body>

<cfscript>
    out = "Hello World";
    writeOutput(out);
</cfscript>

</body>
</html>
```

Note, however, that there are several key differences between CFScript and Javascript that include, but are not limited to the following:

- Variable declarations (`var` keyword) may only be used in user-defined functions and threads.
- Unlike Javascript, CFScript is not case sensitive.
- All CFScript statements must end with a semicolon.
- Line breaks in CFScript are ignored.
- CFScript does not have access to the browser's DOM.
- CFScript supports hundreds of server-side functions including methods for interfacing with databases.

Note: This course focuses on using CFML

Using Alternatives to ColdFusion

In addition to the Adobe ColdFusion server, other CFML engines are available, however, none are 100% compatible with Adobe ColdFusion's feature set. These include:

- **Railo (www.gettrilo.com)**
An open-source ColdFusion engine
- **Lucee (www.lucee.org)**
An open-source fork of Railo that uses a more community-driven model to evolve the ColdFusion language and feature set.
- **BlueDragon**
(<http://www.newatlanta.com/products/bluedragon/index.cfm>)
Enables you to run CFML as a .NET framework assembly and is generally compatible with ColdFusion 7 syntax.

Walkthrough 2-1: Getting Started with CFML



During this exercise you will verify that your ColdFusion server is processing requests.

In this walkthrough, you will perform the following tasks:

- Define a ColdFusion variable
- Output a ColdFusion variable

Steps

Create the Hello World Example

1. Using Notepad, Sublime Text, or any other text editor, open `/ftcf11/walk/walk2-1/index.html`
2. Between the `<body>` tags, use the `<cfset>` tag to define a variable named “myname” that is equal to your name.

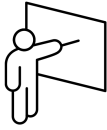
```
<cfset myname = "Steve Drucker">
```
3. After the code that you inserted in the prior step, insert a `<cfoutput>` block.

```
<cfoutput>  
  
</cfoutput>
```
4. Inside the `<cfoutput>` block, insert an `<h1>` block that outputs the string “Hello”, along with the variable that you defined in step 2.

```
<h1>Hello #variables.myname#</h1>
```
5. Save the file and browse the following url:
<http://localhost:8500/ftcf11/walk/walk2-1/index.html>
Note that your output contains the CFML code.
6. Return to your editor and save a copy of the file into the same folder, using a file name of `index.cfm`
7. Browse the following url:
<http://localhost:8500/ftcf11/walk/walk2-1/index.cfm>
Note that you should now see your name output to the web page.
8. Use your browser's View Source feature to view the generated html source code. Note that ColdFusion stripped all of the CFML from the generated output. Why?

– End of Walkthrough --

Introducing the ColdFusion Administrator



The ColdFusion Administrator, depicted in Figure 4, is a web-based application that enables you to configure ColdFusion to connect to other enterprise servers and gateways, allocate system resources for optimal scalability, download and apply server patches, and more.

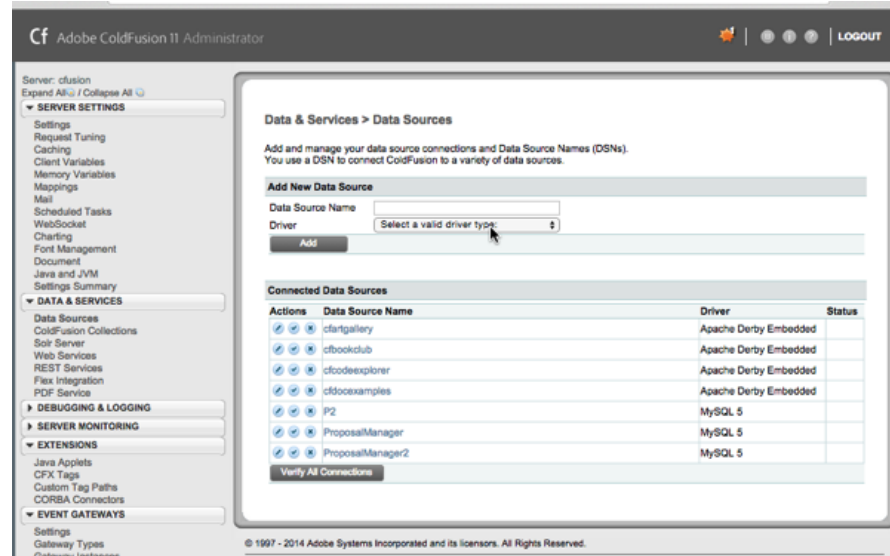


Figure 4: ColdFusion Administrator

Configuration settings generally fall into the following categories:

- Throughput Settings
- Caching
- Variable Persistence*
- SMTP Mail Settings*
- Task Scheduling
- Datasources*
- REST services*
- Debugging*
- Custom Tag Paths*
- Server Mappings*
- Event Gateways
- Security
- Packaging & Deployment
- Clustering
- Server Monitoring
- Server Updates*

Items marked with an asterisk will be covered in this course.

Every feature of the ColdFusion Administrator is available for you to invoke programmatically via the CF Admin API.

Note: For a complete description of how to properly secure and configure your ColdFusion server, please see our follow-on class, [Administering ColdFusion 10/11](#).

Connecting to a Database

Writing code to interface with a database first requires that you register that database with the ColdFusion Administrator.

ColdFusion Enterprise supports the following databases:

- Apache Derby
- DB2
- Microsoft SQL Server
- MySQL
- ORACLE
- Postgress
- Sybase

You can also connect to any other database that has an available JDBC driver available.

Note: You will use MySQL and Apache Derby during this course.

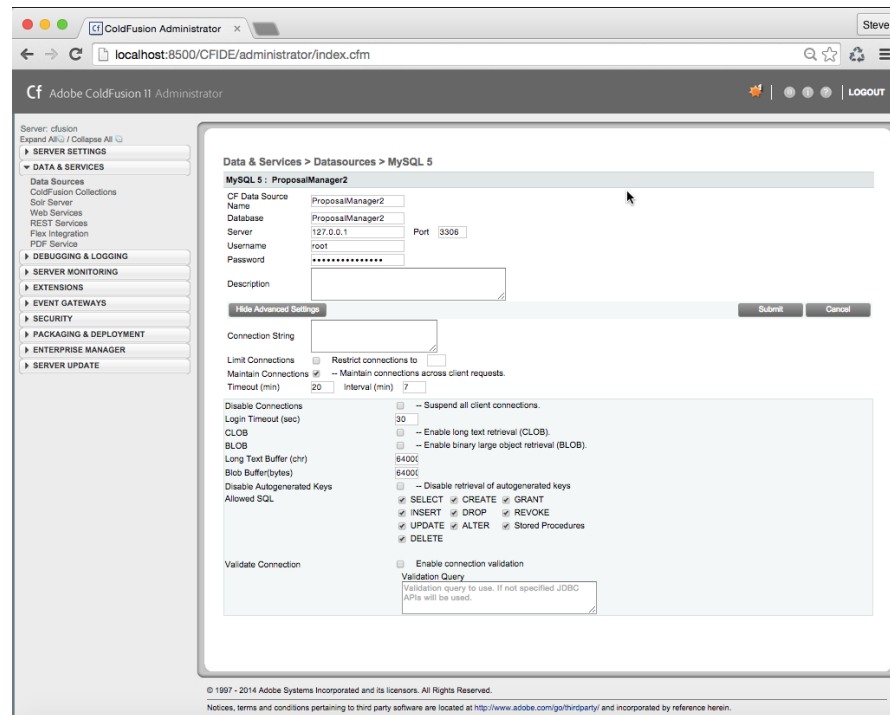


Figure 5: Connecting to a MySQL5 Database

As illustrated in Figure 5, connecting CF to your database generally requires you to enter the following information:

- **Datasource Name**
This is the name by which you will refer to your data schema from within your ColdFusion application. While it can be any string, the best practice is to not embed any spaces or special characters in the name.
- **Database**
The name of the database schema that you intend to access.

- **Server**
The TCP/IP address or host name of your database server
- **Port**
The TCP/IP port on which your database server is listening for requests.
- **Username**
The database account under which all requests will be issued, unless an alternative username is supplied in your code. Care should be taken to create a database user account that is specific to your ColdFusion application and that has limited rights to the specified schema.
- **Password**
The password for the database account under which all requests will be issued unless an alternative password is supplied in your code.
- **CLOB**
This box should be checked if your database has any TEXT/Memo fields that can exceed 64K characters in length.
- **Allowed SQL**
This feature enables you to restrict access to specific SQL commands. Most ColdFusion applications typically only issue SELECT/INSERT/UPDATE/DELETE commands as well as invoke Stored Procedures.

Connecting to an SMTP Mail Gateway

ColdFusion's <cfmail> tag enables you to dynamically generate and send email messages. Before you can use <cfmail>, however, you must set a default SMTP mail gateway as illustrated in figure 6.

The screenshot shows the ColdFusion Administrator web interface. The browser address bar indicates the URL is localhost:8500/CFIDE/administrator/index.cfm. The page title is "Adobe ColdFusion II Administrator". On the left, a navigation menu is visible with categories like "SERVER SETTINGS", "DATA & SERVICES", "DEBUGGING & LOGGING", "SERVER MONITORING", "EXTENSIONS", "EVENT GATEWAYS", "SECURITY", "PACKAGING & DEPLOYMENT", "ENTERPRISE MANAGER", and "SERVER UPDATE". The "SERVER SETTINGS" section is expanded, and the "Mail" sub-section is selected. The main content area displays the "Mail Server Settings" configuration page. At the top of this page, there is a "Submit Changes" button. Below that, the "Server Settings > Mail" section is shown. The "Mail Server Settings" section includes the following fields and options:

- Mail Server:** localhost
- User name:** (empty)
- Password:** (empty)
- Sign the mail**
Select this check box to have ColdFusion Sign the Mail.
- Keystore:** (empty) with a "Browse Server" button
- KeystorePassword:** (empty)
- KeyAlias:** (empty)
- KeyPassword:** (empty)
- Verify mail server connection**
Select this check box to have ColdFusion verify that it can connect to this mail server when you submit this form.
- Server Port:** 25
- Connection Timeout(in seconds):** 60

Figure 6: Connecting to a Mail Server

Downloading and Applying Server Updates

As illustrated by Figure 7, the ColdFusion Administrator alerts you when new product updates are available.

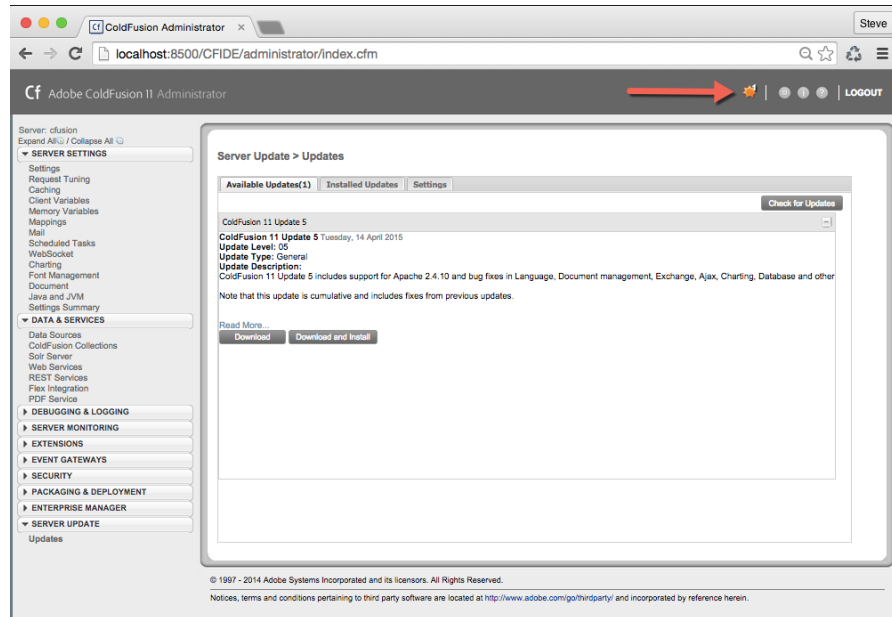


Figure 7: Patching the ColdFusion Service

Clicking on the *Server Updates* link takes you to a screen where you can download and install the latest updates at the click of a button. Note that server updates should always be tested on your development and staging infrastructure before being moved into production.

Walkthrough 2-2: Using the ColdFusion Administrator



During this exercise you will configure your ColdFusion server for development and apply any available patches.

In this lab, you will perform the following tasks:

- Install any available ColdFusion patches
- Define a MySQL Datasource
- Connect to an SMTP mail gateway

Steps

Install ColdFusion Patches

1. Open your web browser to **`http://localhost:8500/CFIDE/administrator/`**
2. Enter your admin password and click the **Login** button.
3. Click on **Server Update > Updates**. You should see at least one update that is available.
4. Click on the **Download and Install** button.
5. At the prompt notifying you that the CF service will be restarted, click the **Continue** button.
6. Wait for the CF service to restart.
7. Click the **OK** button to view installed updates.
8. Enter your password and click the **Login** button.
9. Verify that the latest update was installed.

Define a MySQL Datasource

10. Click on **Data & Services > Data Sources**
11. Enter the following information:
 - Data Source Name: ProposalManager
 - Driver: MySQL 5
12. Click **Add**

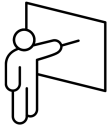
13. Enter the following information:
 - Database: ProposalManager
 - Server: 127.0.0.1
 - UserName: cfuser-ftcf
 - password: learncf!
14. Click the **Submit** button. The ColdFusion administrator should report that your connection to the database succeeded.
15. Click on the **ProposalManager** datasource.
16. Click on the **Show Advanced Settings** button.
17. Turn on the checkbox to **Enable long text retrieval (CLOB)**
18. Turn off the checkboxes to allow **CREATE, DROP, ALTER, GRANT,** and **REVOKE** operations.
19. Click the **Submit** button.

Connect to an SMTP Mail Gateway

20. Click on Server Settings > Mail
21. Enter the following information:
 - Mail Server: localhost
 - Verify Mail Server Connection: checked
22. Click the **Submit Changes** button. ColdFusion should report that it can connect to your local James mail server.

-- End of Walkthrough --

Working with ColdFusion Builder



Since ColdFusion is a script-based language, you can use any editor that outputs plain text to develop your codebase. However, there are a number of integrated development environments (IDE) available that can help facilitate coding and debugging including:

- Sublime Text
- JetBrains IntelliJ IDEA
- Adobe Brackets
- ColdFusion Builder
- ColdFusion Report Builder

Using Sublime Text

Sublime Text 2 is a nearly-free text editor with a plugin architecture. Language helpers are available for ColdFusion, HTML, CSS, and nearly every other web framework.

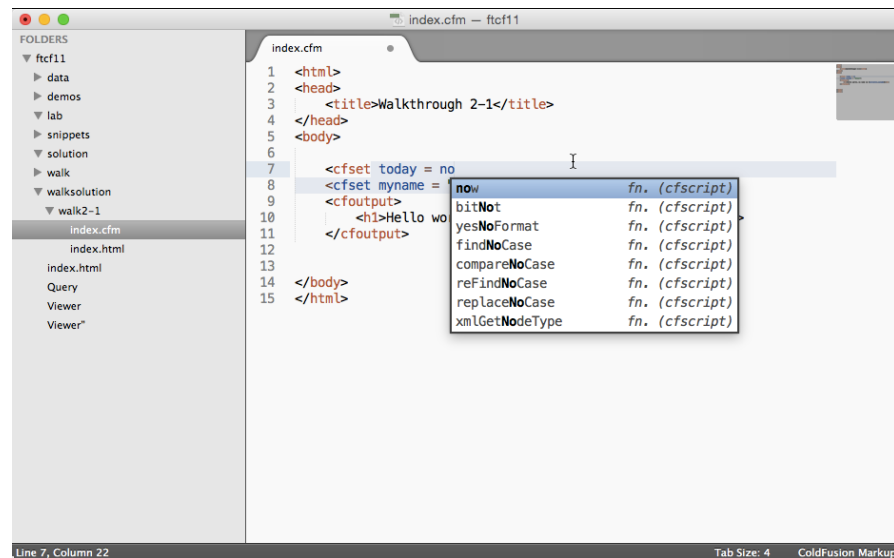


Figure 8: Sublime Text 2

Developers who have achieved a good working knowledge of ColdFusion and web scripting tend to gravitate towards using Sublime because of its speed and laser-focus on writing code.

ColdFusion-specific plugins for sublime can be installed via Sublime's package manager.

Using JetBrains IntelliJ IDEA

JetBrains IntelliJ IDEA is a well-respected, commercial integrated development environment for developing Java applications. As illustrated in Figure 9, IntelliJ IDEA has built-in code hinting for CFML.

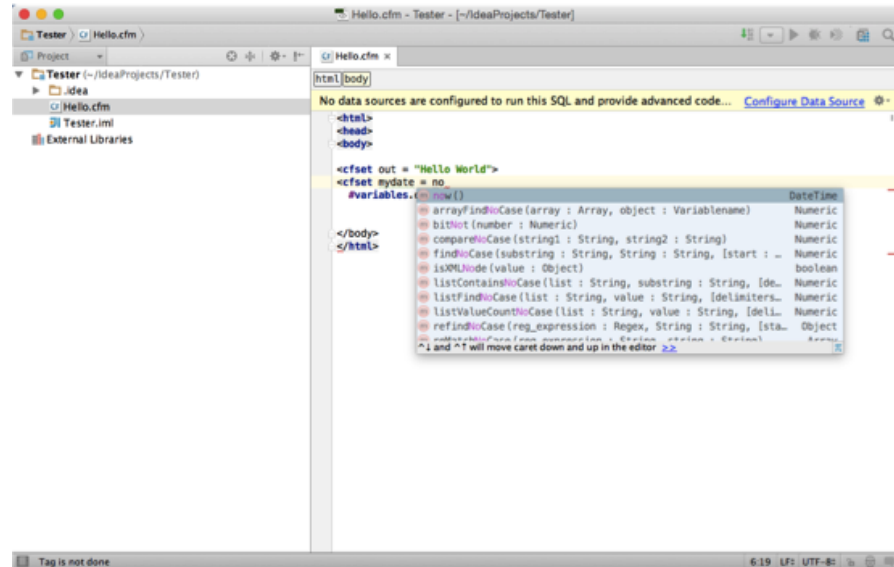


Figure 9: JetBrains IntelliJ IDEA

Using Adobe Brackets

Adobe Brackets is a free, open-source editor geared towards web development. It features a plugin architecture similar to the one found in Sublime Text that supports CFML code hinting.

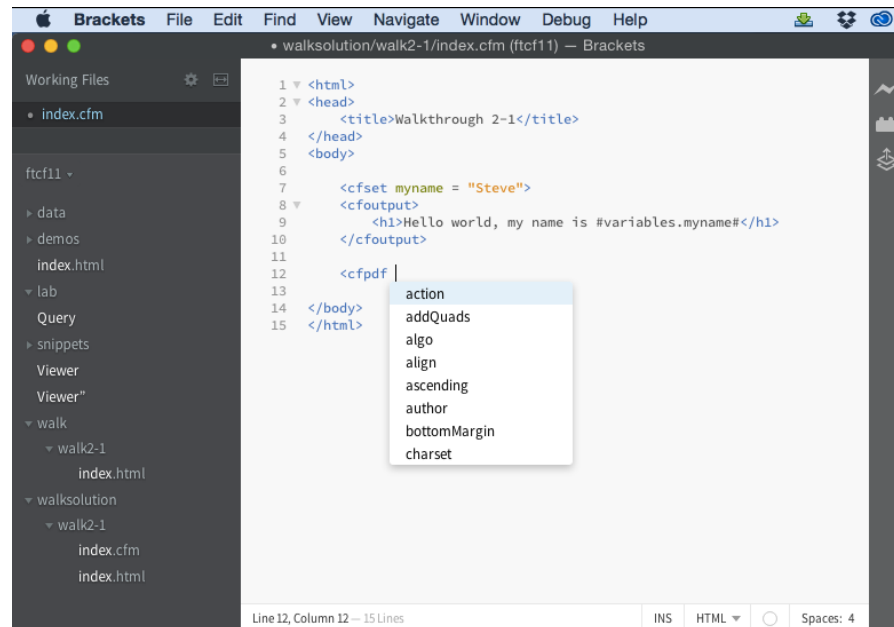


Figure 10: Adobe Brackets has a CFML plugin for limited code hinting.

Using ColdFusion Builder

ColdFusion Builder 3.x is the only fully-featured integrated development environment for building ColdFusion applications. It is based on Eclipse and packaged in either a standalone app or as an Eclipse plugin.

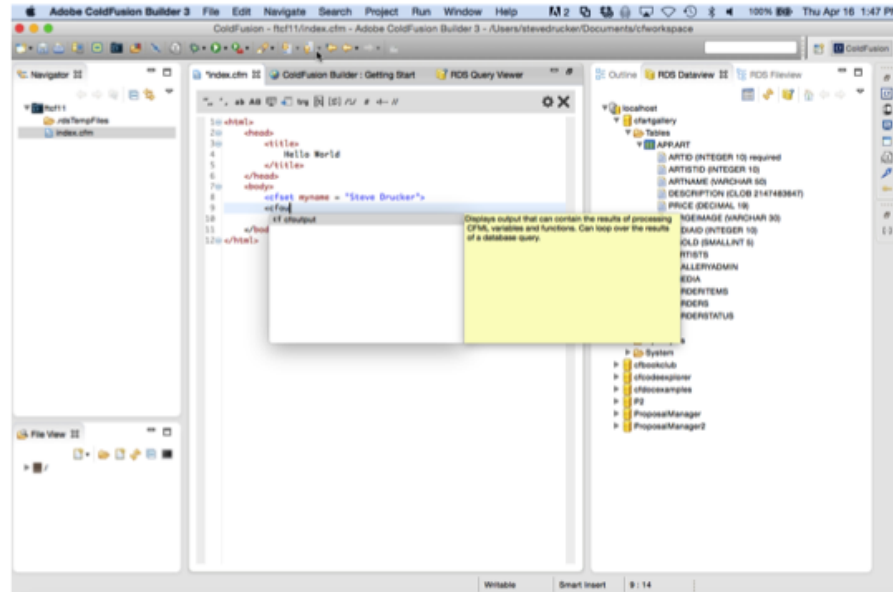


Figure 11: ColdFusion Builder v3

Features include, but are not limited to the following:

- Code Completion
- Automatic Code Formatting
- RDS Connection to Server
- Integrated Task Management
- Query Builder
- Code Insight
- ColdFusion Framework Support
- Interactive Debugging
- Tailview Log Viewer
- Integration with Phoneygap Build
- CF Server Stop/Start
- Multidevice Inspection
- Advanced Find/Replace
- Snippets

Creating a ColdFusion Project

Create a new ColdFusion project in ColdFusion builder by selecting File>New>ColdFusion project or, if you have an existing codebase, File>Import>ColdFusion>Import existing Project.

As illustrated in Figure 12, defining your project is simply a matter of assigning it a name, location, and ColdFusion version number.

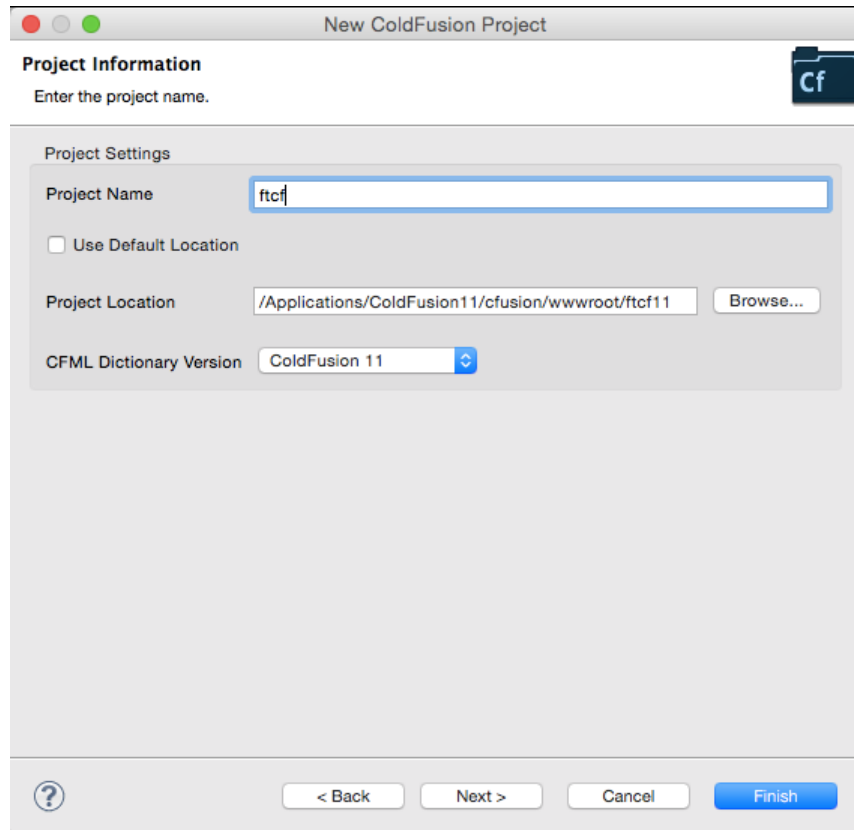


Figure 12: Creating a New ColdFusion Project

Configuring the RDS Connection

RDS is a protocol whereby CFBuilder communicates interactively with your ColdFusion server over http/https in order to display information about datasources, edit files on the server, and use the step-through interactive debugger.

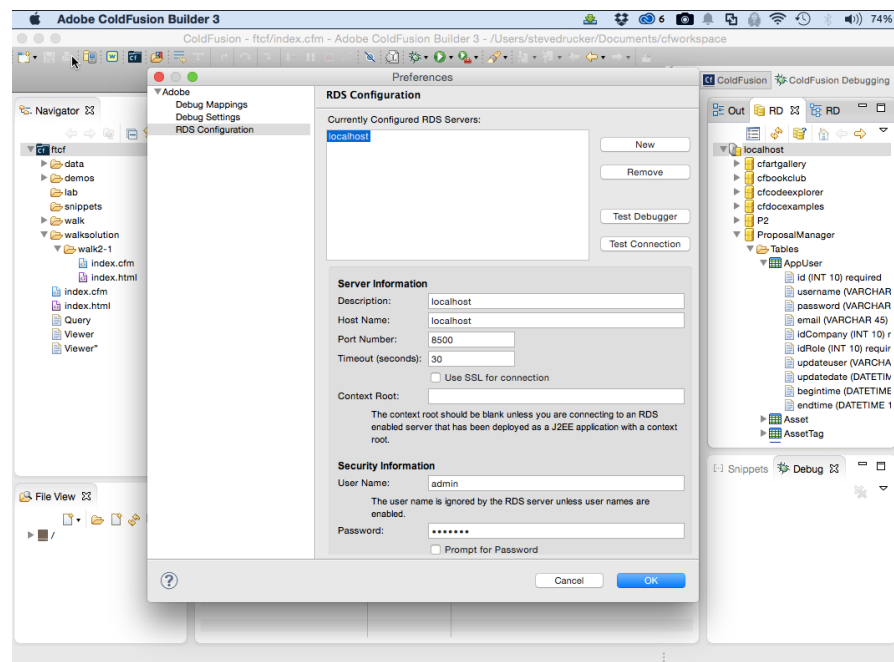


Figure 13: Configuring RDS

To link your project to a server via RDS, open the RDS Dataview view, right-click on your server, and select RDS configuration. The RDS configuration dialog should appear as illustrated in Figure 13.

Enabling RDS

The username and password fields of the RDS configuration in ColdFusion Builder map to the RDS section of the ColdFusion Administrator as illustrated in Figure 14.

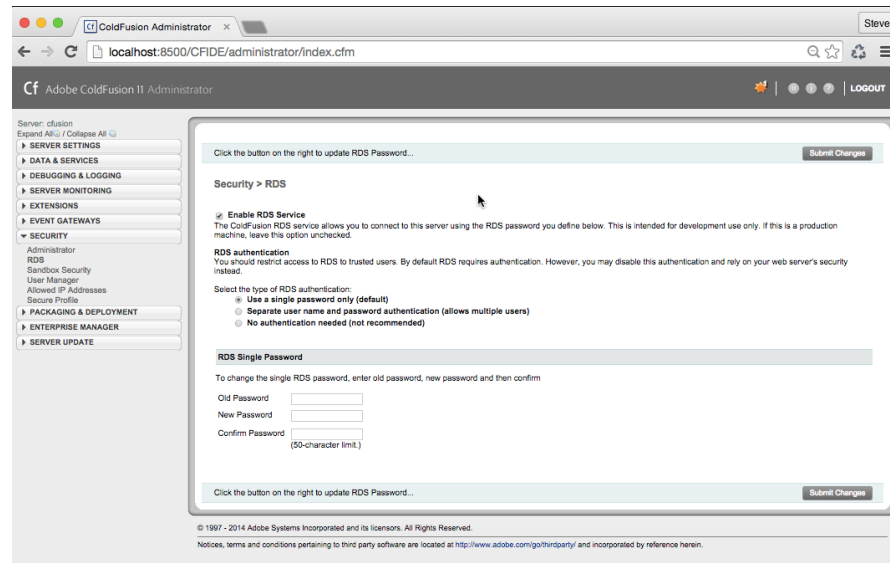


Figure 14: Configuring RDS Logins

Note that you can create separate username/passwords for each of your developers, or give them a single RDS password to use.

Adding RDS Accounts for Developers

You can define additional RDS logins by accessing the User Manager section of the ColdFusion Administrator, as illustrated in Figure 15.

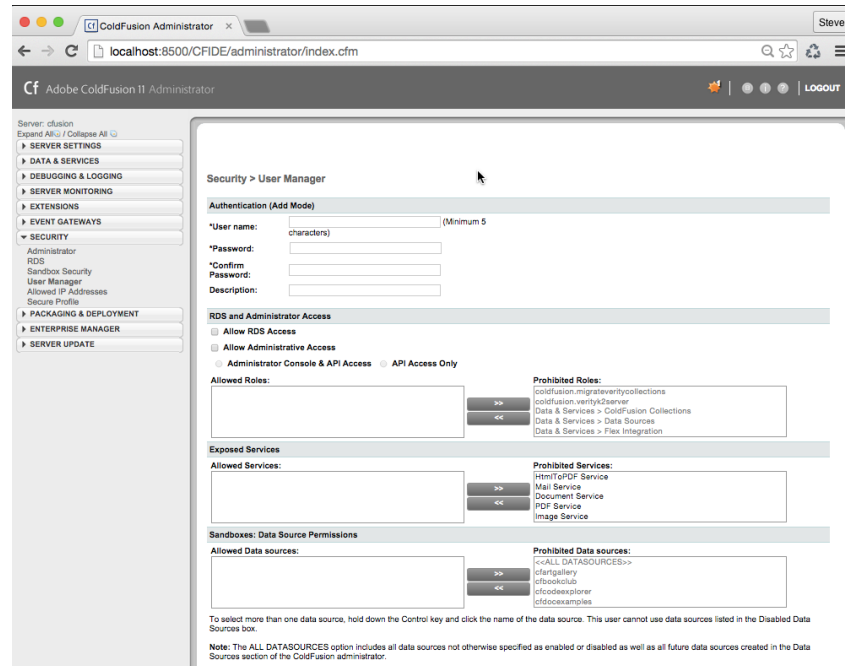


Figure 15: Adding RDS User Accounts

Note that each developer can be granted restricted access to specific datasources, directories on the server's file system, and features of the ColdFusion Administrator.

Inspecting Datasources From ColdFusion Builder

ColdFusion Builder's RDS DataView view communicates over RDS to the ColdFusion server, making critical information about your datasources available for you to view.

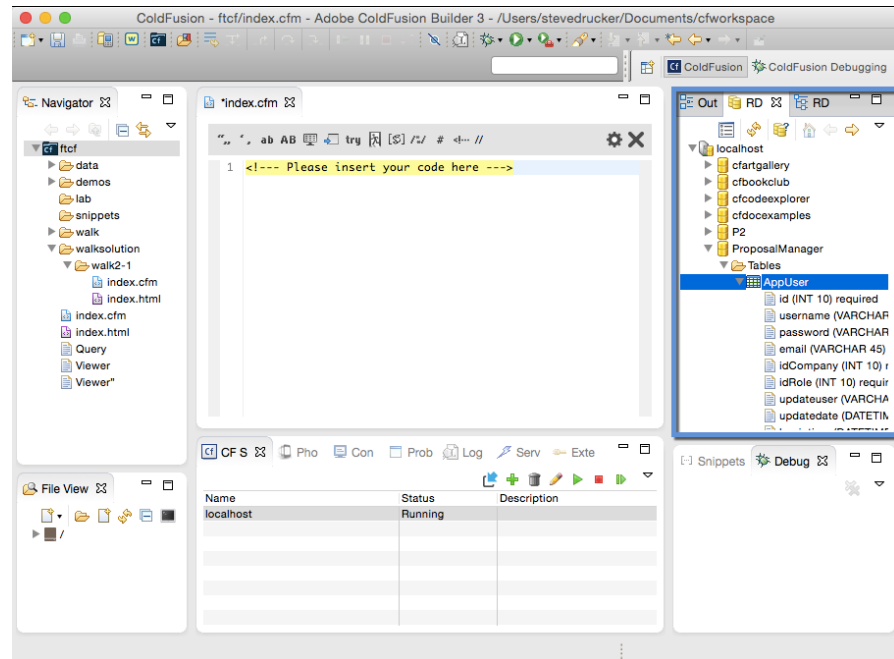


Figure 16: RDS DataView

From RDS DataView you can:

- Inspect tables, columns, views, and stored procedures for any registered datasource.
- View the contents of any database table.
- Graphically create a SQL query.
- Drag and drop datasource names, table names, and column names from the DataView panel into your source code.

Using the ColdFusion Report Builder

The Adobe ColdFusion Report Builder is a visual integrated development environment for creating reports from ColdFusion datasources.

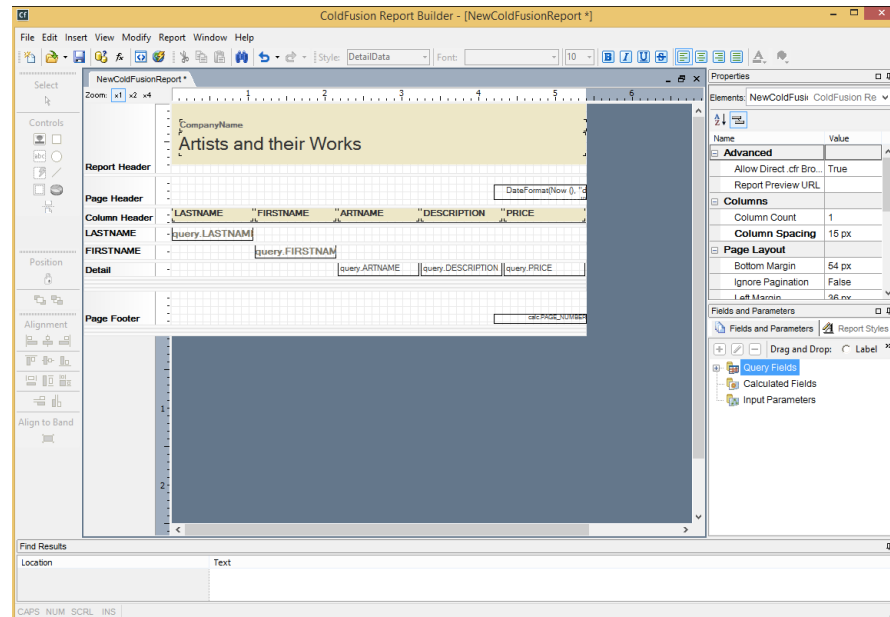


Figure 17: The ColdFusion Report Builder

In order to function properly, the report builder must be able to access your ColdFusion server via RDS. Using the report builder, you define a single query that supplies the core data for your report. You can then drag and drop the resulting fields into any position that you'd like.

The report builder generates .CFR files, which can be accessed either directly by URL or invoked via ColdFusion's `<cfreport>` command. Output from reports can be translated into any of the following file formats:

- PDF
- Excel
- RTF
- HTML
- XML

The ColdFusion Report Builder is available as a free download for Windows operating systems from:

<https://www.adobe.com/support/coldfusion/downloads.html>

Walkthrough 2-3: Using ColdFusion Builder



During this exercise you will configure a new project in ColdFusion builder by performing the following tasks:

- Enable RDS
- Create a new project
- Configure the CF Builder RDS connection
- Inspect datasources
- Browse a project file

Steps

Verify that RDS is Enabled

1. Open the ColdFusion Administrator at <http://localhost:8500/CFIDE/administrator/>
2. Click on **Security > RDS**
3. Verify that **Enable RDS** is checked.

Create a New CF Builder Project

4. Launch ColdFusion Builder
5. Select **File > New > ColdFusion Project**
6. Click the **Next** button.
7. Enter a project name of **FTCF**
8. Uncheck the checkbox labeled **Use Default Location**
9. Click the Browse button and select **/ColdFusion11/cfusion/wwwroot/ftcf11**
10. Click **Next**.
11. Click the **Servers** select box and choose **Add New Server**
12. Enter the following information:
 - Server name: **localhost**
 - Host Name: **localhost**
 - WebServer Port: **8500**
 - RDS User Name: **Admin**
 - RDS Password: **learnCF!**
13. Click the **Next** button.

14. Click the **Browse** button adjacent to **Server Home** and select your ColdFusion installation folder, e.g. [C:\ColdFusion11](#). The Document Root field should automatically populate.
15. Click the **version** select box and choose 11.0
16. Click on the **URL Prefix** tab.
17. Click the Browse button adjacent to the **Local Path** property and choose your ftcf11 folder, e.g.
C:\ColdFusion11\cfusion\wwwroot\ftcf11
18. Enter a URL prefix of <http://localhost:8500/ftcf11>
19. Click the **Add** button.
20. On the **Install Extensions** dialog box, click the **Next** button.
21. Click **Finish**.
22. On the **Server Details** dialog box, click the **Finish** button.

Enable RDS Query View

23. In the Navigator view, confirm that there is a directory named **.rdsTempFiles**. If no such directory exists, create one by right-clicking on the ftcf project and selecting **New > Folder**.
24. Inside the **.rdsTempFiles** folder, confirm that there exists a file named "RDS Query Viewer". Create the empty file if none exists.

Use RDS Query View

25. Click on the **RDS Dataview** view.
26. Confirm that you can inspect the ColdFusion datasources.
27. In the RDS Dataview, select **ProposalManager > Tables > AppUser**
28. Right-click on the **AppUser** table and select **Show Table Contents**. The contents of the data table should appear on screen.

Browse a Project File

29. Open `/ftcf11/walksolution/walk2-1/index.cfm`
30. Select **Run > Run As > ColdFusion Application**. Your default browser should activate and run the currently selected file.

-- End of Walkthrough --

Debugging and Troubleshooting your Apps



ColdFusion has a number of features to help you debug and troubleshoot your code including:

- Outputting diagnostic information at the bottom of every page.
- Interactive, step-through debugging with the ColdFusion Administrator.
- Logging error messages in text files.
- Using the `<cflog>` tag to output custom messages into log files.

Outputting Diagnostic Information

The ColdFusion Administrator as depicted in Figure 18 contains a feature that outputs diagnostic information at the bottom of every dynamically generated page.

There is also a feature that you can use to output very specific diagnostic messages whenever an error occurs.

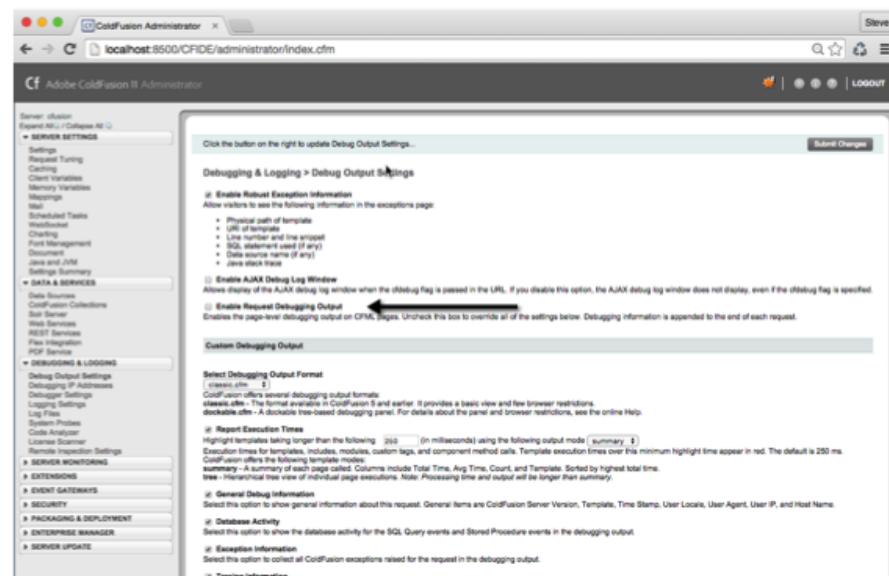


Figure 18: Enabling Request Debug Output

Enabling Robust Exception Information

Turning on the Enable Robust Exception Information feature in the ColdFusion Administrator causes the ColdFusion server to output very specific and highly detailed error messages in the format illustrated by Figure 19.

The web site you are accessing has experienced an unexpected error.
Please contact the website administrator.

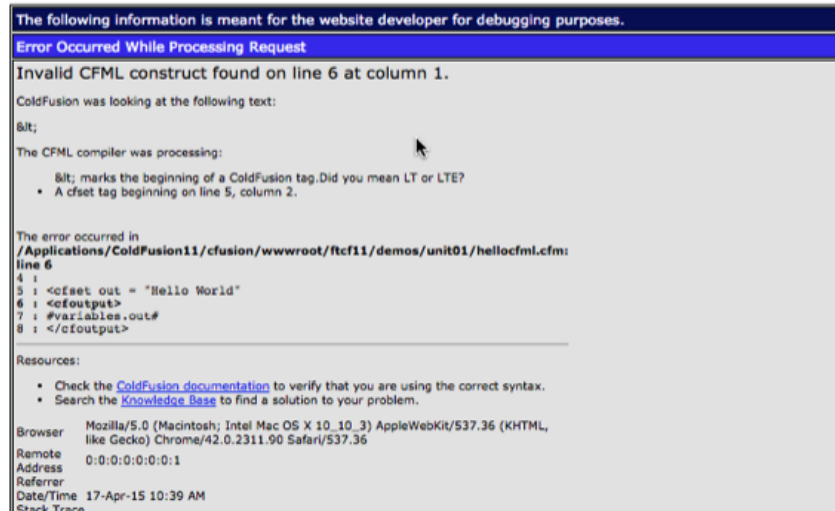


Figure 19: Robust Exception Information

When an error occurs, ColdFusion will show the line number and file name where the problem occurred, your application's source code in the vicinity of the trouble, a full stack trace, and other pertinent details.

Note: Since this feature potentially exposes source code to users of your application, it should be disabled on production servers.

Enabling Request Debug Output

The Request Debug Output, as illustrated in figure 20, outputs the following diagnostic information at the end of each CFM page request:

- Report Execution Times
- General Debug Information
- Database Activity
- Exception Information
- Tracing Information
- Timer Information
- Variables

Debugging Information
ColdFusion Server Developer 11,0,04,293328
Template /fctf11/demos/unit01/hellocfml.cfm
Time Stamp 17-Apr-15 10:42 AM
Locale English (US)
User Agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.90 Safari/537.36
Remote IP 0.0.0.0.0.0.1
Host Name 0.0.0.0.0.0.1

Execution Time

Total Time	Avg Time	Count	Template
3 ms	3 ms	1	/Applications/ColdFusion11/cfusion/wwwroot/fctf11/demos/unit01/hellocfml.cfm
5 ms			STARTUP, PARSING, COMPILING, LOADING, & SHUTDOWN
8 ms			TOTAL EXECUTION TIME

red = over 250 ms average execution time

SQL Queries
q (Datasource=ProposalManager2, Time=2ms, Records=7) in /Applications/ColdFusion11/cfusion/wwwroot/fctf11/demos/unit01/hellocfml.cfm
select * from client

Scope Variables
CGI Variables:
AUTH_PASSWORD=
AUTH_TYPE=
AUTH_USER=
CERT_COOKIE=
CERT_FLAGS=
CERT_ISSUER=
CERT_KEYSIZE=

Figure 20: Request Debug Output

Use this feature to identify poorly performing code that may cause your application to exhibit scalability problems. Note that while there is a feature of the ColdFusion Administrator that enables you to restrict the output of this diagnostic information to specific IP addresses, activation of this feature will degrade overall application performance for all users, not just the ones who see it appear in their browsers.

Starting a Debugging Session

ColdFusion Builder communicates via RDS to the ColdFusion debugging service to enable you to interactively step through and trace your code as its running on the server. Before you can use this application feature, however, you must first start the debugger server as illustrated in Figure 21.

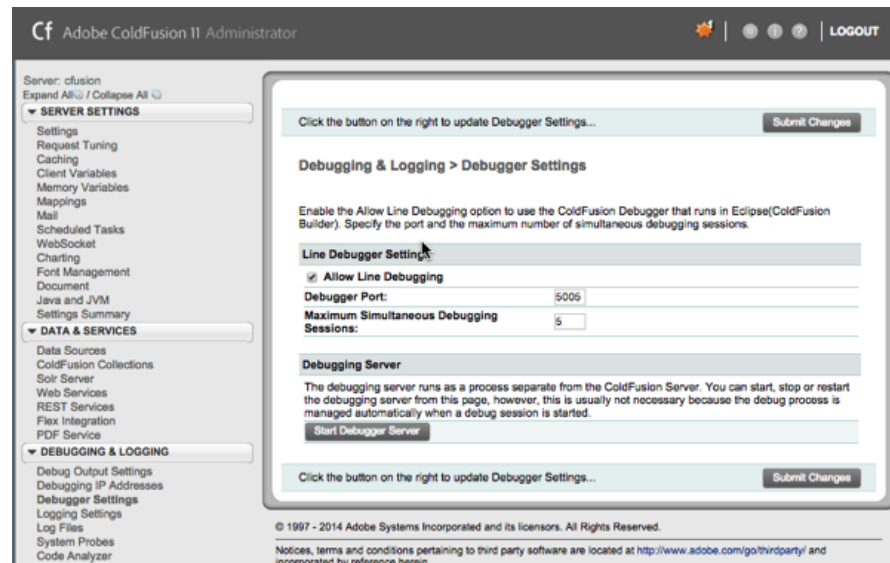


Figure 21: Starting the Debugging Server

After the debugging server has come online you can use ColdFusion Builder to set breakpoints in your code by double clicking in the code editor's margin. Small blue dots in the margin indicate where breakpoints have been set and execution will be paused. You then begin the debugging session by launching the debugging perspective as illustrated in Figure 22.

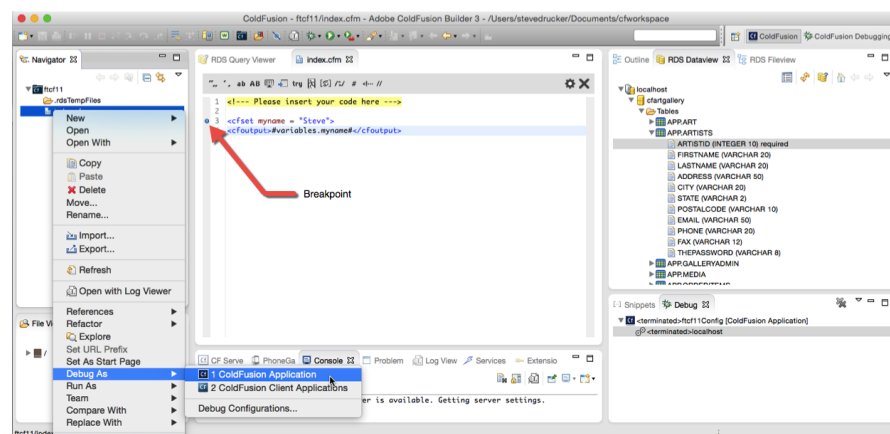


Figure 22: Setting a breakpoint and starting a debug session

Using the Interactive Debugger

The debugging perspective, as illustrated in Figure 23, segments the interface into the following views:

- The **Debug View** indicates the file that you are currently debugging.
- The **Variables View** enables you to inspect and change the contents of any variable.
- The **Debug Output Buffer** indicates the output generated from your code up to the currently active line.
- The **Breakpoints View** shows a listing of all breakpoints in your code.
- The **Expressions View** enables you to output the value from any valid ColdFusion expression.

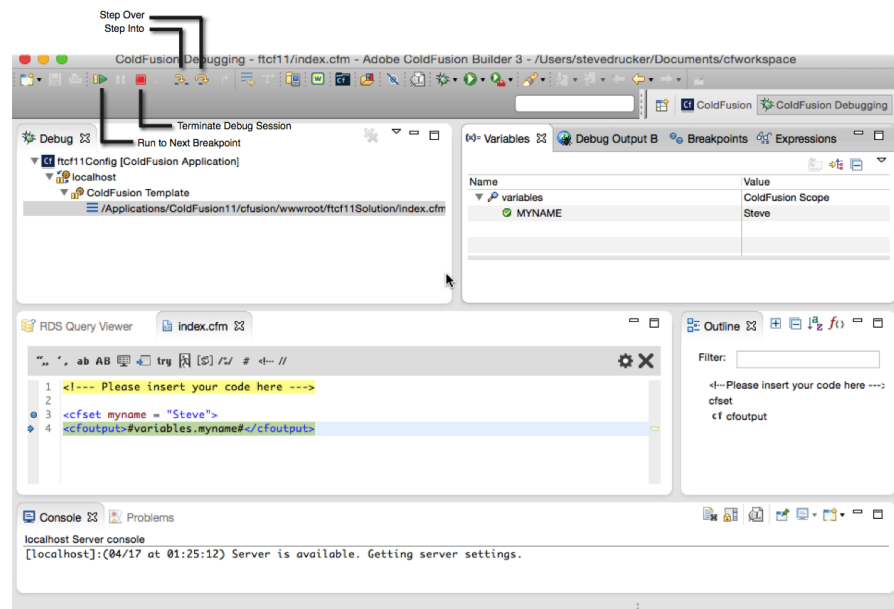


Figure 23: Using the Debugging Perspective

Stepping Through Your Code

Once a breakpoint has become active, you can proceed to execute the following lines of code by using the controls in the top toolbar as illustrated by Figure 24.

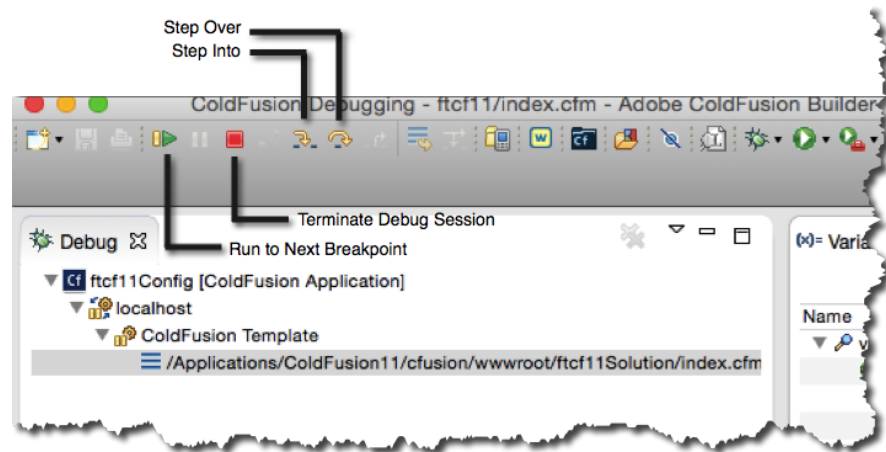


Figure 24: Using the Debugging Controls

Configuring the Request Timeout

Note that even though you are in a debugging session, your page remains subject to the ColdFusion Server's timeout settings as illustrated in Figure 25. Before you debug, consider increasing the request timeout value to give yourself enough time to complete your debugging task.

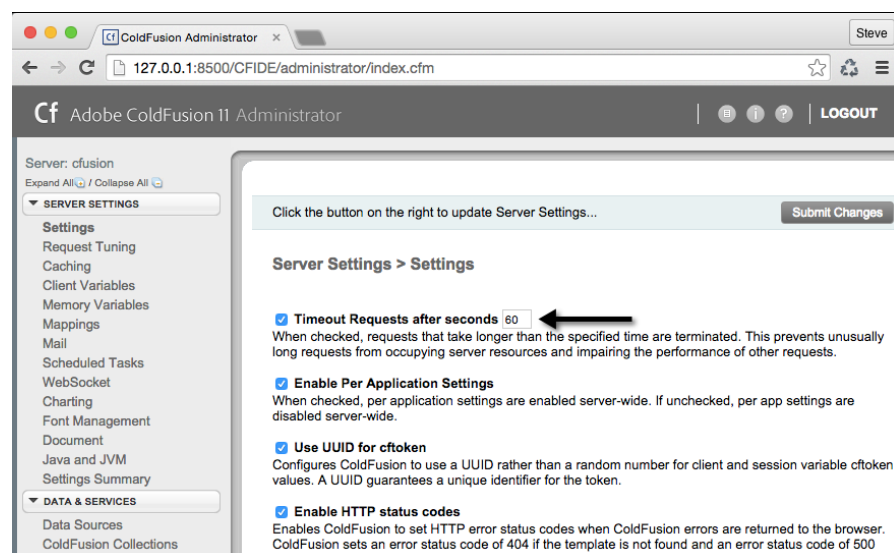


Figure 25: Configuring the Request Timeout

Reviewing the Log Files

Periodically reviewing your ColdFusion server's log files is a best practice to ensure that your applications are continuing to perform correctly under all conditions and use-cases.

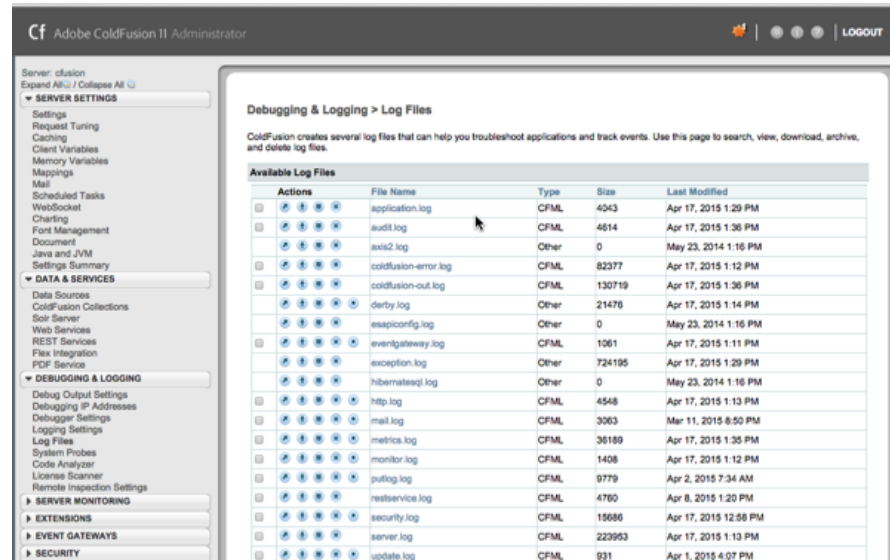






Figure 26: ColdFusion Log Files

In the event that an error occurs, ColdFusion will output details to a series of log files that you can inspect from within the ColdFusion Administrator and ColdFusion Builder. Typically, each application error is written to the application.log file and a complete stack trace is written to exception.log.

You can review all of the ColdFusion logs by accessing **Debugging & Logging > Log Files** from the ColdFusion Administrator as depicted in Figure 26.

Log files can get quite lengthy so you might want to set a business rule to archive the logs at various intervals. In addition, you can manually activate the following features for each log file:

Button	Description
	Search/View Log File
	Download Log File
	Archive Log File
	Delete Log File

Using the ColdFusion Administrator Log Viewer

The ColdFusion Administrator log viewer, depicted in Figure 27, parses the comma-delimited log file, displaying entries in a tabular format. Clicking on the Launch Filter button spawns a new browser window where you can enter filter criteria in order to pare down the list.

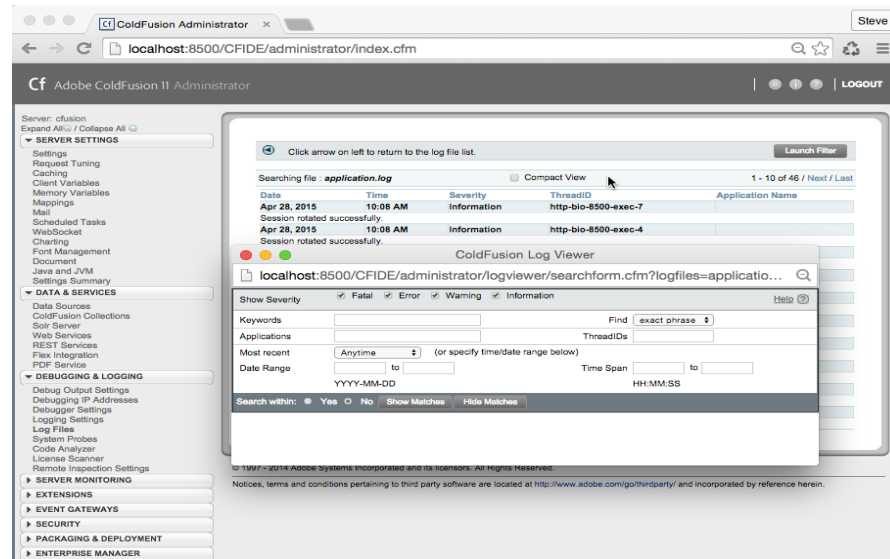


Figure 27: The ColdFusion Log Viewer

Using the Integrated Baretail Viewer in ColdFusion Builder

ColdFusion Builder's integrated Baretail Viewer, depicted in Figure 28, enables you to monitor the contents of your log files in real-time from within your development environment. You can also create rules to color-code log entries based on their contents.

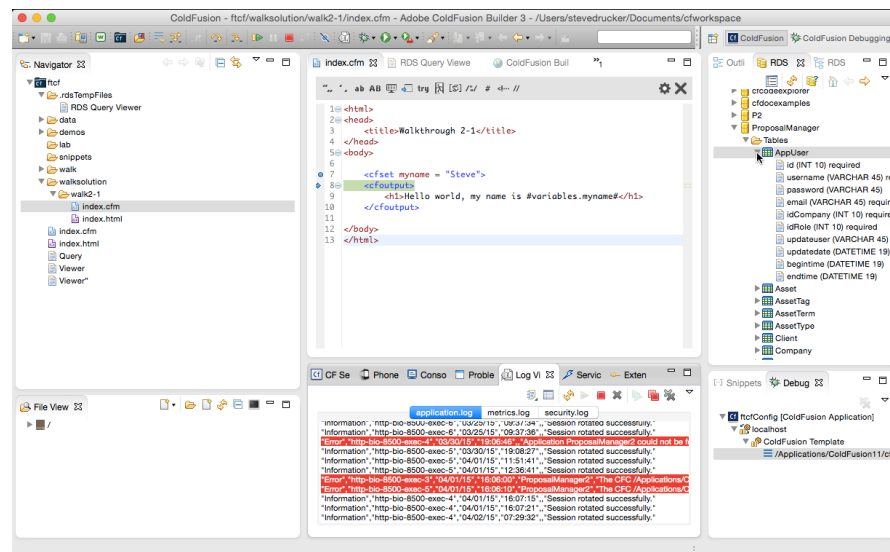


Figure 28: The Baretail Viewer

Programmatically Writing to a Log File with <cflog>

Use the <cflog> tag to output messages to your own custom logs, or ColdFusion's default log files. The basic syntax is as follows:

```
<cflog
  text = "text"
  type = "information|warning|error|fatal"
  application = "yes|no"
  file = "filename"
  log = "log type">
```

Writing data to logs can help you trace problems with business logic at runtime, without significantly impacting performance.

The following example illustrates how to dump the contents of a variable to a custom log file, resulting in the log entry depicted in Figure 29:

```
<cfset myname = "Steve">

<cfoutput>
  <h1>Hello world, my name is #variables.myname#</h1>
</cfoutput>

<cflog
  file="ProposalManager"
  type="information"
  text="MyName: #variables.myname#">
```

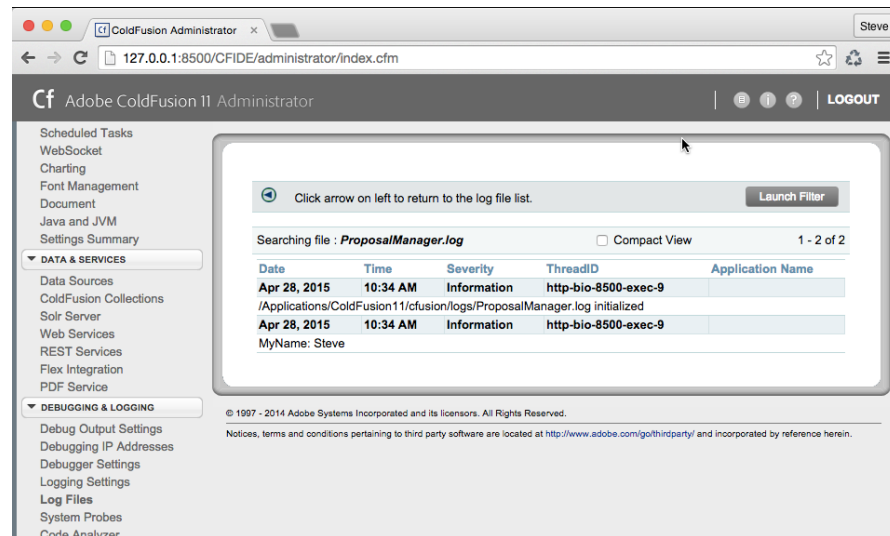


Figure 29: Custom log entry

Walkthrough 2-4: Debugging your Code



During this exercise you will experiment with debugging and logging

In this lab, you will perform the following tasks:

- Log the contents of a variable to the debugger
- Step through your code interactively

Steps

Write Entries to a Log File

1. Open ColdFusion Builder
2. Open `/walk/walk2-4/index.cfm`
3. At the bottom of the file, insert a `<cflog>` tag to output the contents of the `myname` variable to a log file named “ProposalManager” with an severity level of “information”

```
<cflog
  file="ProposalManager"
  type="information"
  text="MyName: #variables.myname#">
```

4. Save the file.
5. Browse the file.

View Log Entries

6. Open the ColdFusion Administrator by browsing <http://localhost:8500/CFIDE/administrator/index.cfm>
7. Click on Debugging & Logging > Log Files
8. Click on ProposalManager.Log. You should see your custom log entry.
9. Press the browser's back button.

Filter a Log File

10. Click on `application.log` and review its contents
11. Click the **Launch Filter** button.
12. Uncheck the **Warning** and **Information** severity checkboxes and press the **Show Matches** button. Note if any errors were displayed.
13. Close the **Filter** window.

Turn on Debug Output

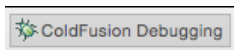
14. In the ColdFusion Administrator, click on **Debugging & Logging > Debug Output Settings**.
15. Check the following boxes if they are not already checked:
 - Enable Robust Exception Handling
 - Enable Request Debugging Output
16. Click the **Submit Changes** button.
17. Return to your browser and reload `/walk/walk2-4/index.cfm`. You should see that diagnostic information information was output to the bottom of the page.

Enable the Interactive Debugging Service

18. Return to the ColdFusion Administrator.
19. Click on **Debugging & Logging > Debugger Settings**
20. Click on the **Start Debugger Server** button.
21. Click on Server Settings > Settings
22. Uncheck the “**Timeout Requests after seconds**” checkbox and press **Submit Changes**.

Interactively Step Through Your Code

23. Return to ColdFusion Builder and `/walk2-4/index.cfm`
24. Double-Click in the gray left margin of the code view on the line that sets the myname variable. A small blue dot should appear.
25. Launch the debugger by right-clicking on the `index.cfm` file in the Navigator view and selecting **Debug As > ColdFusion Application**. Google Chrome should launch and you should see an arrow appear next to your breakpoint.



26. In the top-right corner of ColdFusion Builder, click on the ColdFusion Debugging button to enter the ColdFusion Debugger Perspective.



27. Click the **Step Over** button to advance execution to the next line. You should see that the variable myname appears in the Variables view.



28. Click the **Step Into** button to advance into the `<cfoutput>` block.

29. Click **Step Over** to advance past the `<h1>` element.

30. Click on the **Debug Output Buffer** view and inspect its contents.

31. Step through the code until you reach the end of the file.



32. Click on the ColdFusion perspective button to return back to your coding workspace.

– End of Exercise –

Unit Summary



- ColdFusion Server is a J2EE application server that is used to dynamically generate content from requests that are typically received via the Hypertext transfer protocol.
- You can program the server using a combination of CFML and CFScript.
- ColdFusion can dynamically generate every commonly used file format.
- Requests are passed from the webserver to the ColdFusion server based on the requested filename's extension.
- ColdFusion supports the creation of RESTful web services for participation in a three-tiered architecture.
- As you develop your application, consider your code's impact on scalability, performance, robustness, and security.
- ColdFusion is available in multiple editions and on Amazon Web Services.
- Use the ColdFusion Administrator to link your server to external back-office resources such as databases and mail gateways.
- You can develop apps using any text editor.
- ColdFusion Builder communicates with your server over RDS and has features that can help you rapidly build your applications.
- Many features are available to help you troubleshoot your apps.
- The interactive debugger in ColdFusion Builder enables you to step through your code one line at a time.
- You should regularly inspect your server logs to verify that your applications are performing as expected.

Unit Review



1. List three differences between ColdFusion Standard and ColdFusion Enterprise.
2. List three differences between CFScript and Javascript.
3. Describe a three-tiered architecture.
4. ColdFusion requires a connection to a web server in order to respond to http requests (true/false)?
5. ColdFusion can only respond to http requests (true / false)?
6. What are the advantages to using the step-through debugger to troubleshoot your code?
7. Which ColdFusion features discussed in this unit affect scalability?
8. Which ColdFusion features discussed in this unit affect security?