Coldfusion 9 with Coldbox V3 Framework

Application Development Team
Conyers Dill and Pearman, 2010
Faheem Khan

Outline of Presentation

1) Reasons for Initially choosing Coldbox
2) CF9 Framework Environment with Coldbox 3.0.
3) Example Code (OO Structure)
4) File Structure
5) SSO Authentication with IIS
6) Coding CFC Example CF ORM Hibernate
7) Example Screen CF ORM Hibernate Model
8) SOAP Call with CF9

Outline (cont)

9) Example Coding Handler

11) Future of Coldbox & CF

12) Current Desires for New CF Release

10) Main Benefits of CF9

11) Recommendations: Why we should not go with Coldbox 3.0 ?

12) QuestionTime

13) Resources

Reasons for Initially choosing Coldbox 3.0

- Convention Based Framework.
- An MVC Framework
- Supports CF ORM Hibernate's ORM Integration
- Pretty URLs
- An Object Oriented Framework Architecture moving away from procedural architecture such as Fusebox.

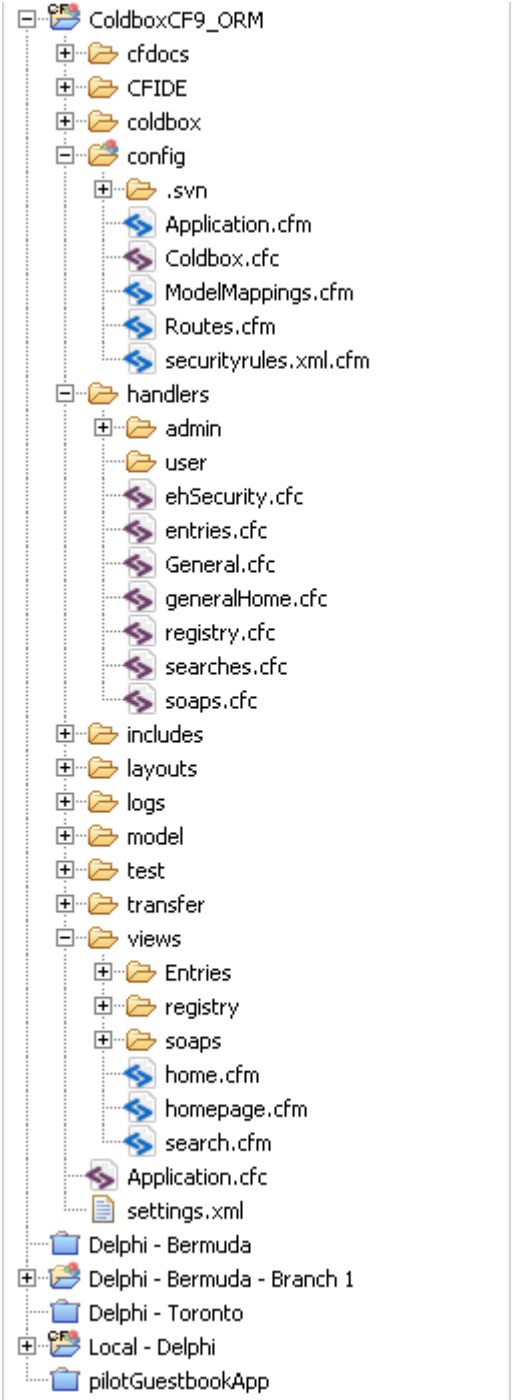
• CF9 Framework Environment with Coldbox 3.0.

- IIS 5.1
- Coldfusion 9 standalone (free developer version)
- Coldbox version 3.0
- MS SQL Server 2005
- Windows XP

Example code (OO Structure)

```
void function remove(entry_ID)
output=false{

transaction{

// get entry

var entry = getEntry(entry_ID);

if( NOT isNull(entry.getEntry_ID()) )
{

entityDelete(entry);

}

}
```

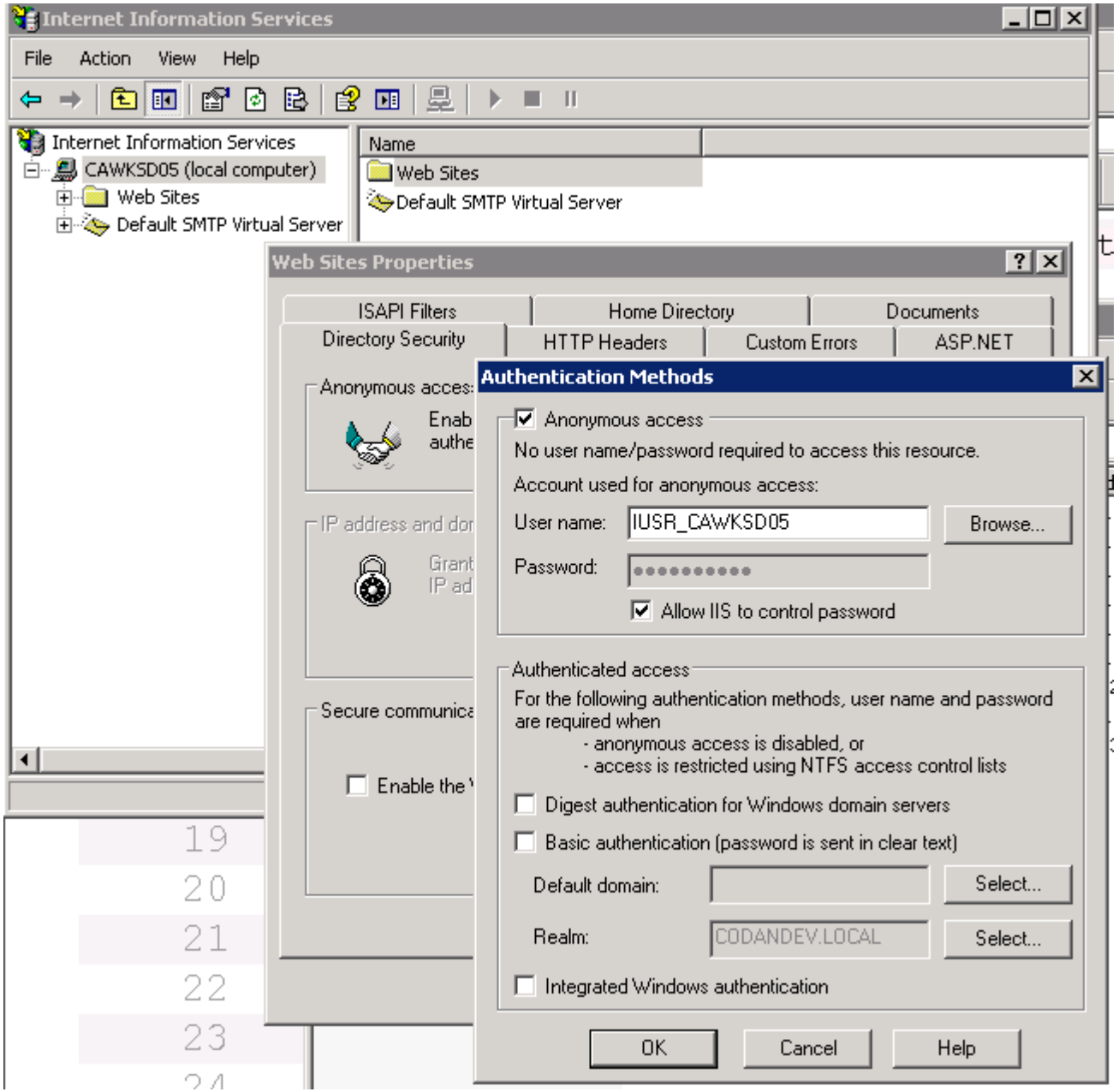
File Structure



SSO Authentication with IIS

- Authentication done in IIS 5.1
- User login credentials taken from cgi.remote_users and then matched against the SQL database
- If user details were matched then user is logged in
- If its the first time the user logs into the system the user gets automatically registered to the database.
- If there are no users registered to the pilot database the first user gets registered as an 'Administrator' and any subsequent users are registered as 'user'

SSO Authentication setup in IIS



Coding CFC Example CF ORM Hibernate

- CRUD Example – The create-read-update-delete operations were done using hibernate and no SQL was used.
- Hibernate also generated all the DDL scripts to create DB schema
- (tables, constraints, sequences) inside of SQL Sever
- By using hibernate which fully supports OO any change that was made to the DB was able to be persisted to the database until the hibernate session gets closed.
- The persistence for the database was persisted until a save or an update was made to the DB which then closes the hibernate session.

- http://hg.itweb/poc_coldbox/file/fa7388183320/model/entry.cfc
- http://hg.itweb/poc_coldbox/file/fa7388183320/model/user.cfc
- http://hg.itweb/poc_coldbox/file/fa7388183320/model/entryService.cfc
- http://hg.itweb/poc_coldbox/file/fa7388183320/model/userService.cfc
- http://hg.itweb/poc_coldbox/file/fa7388183320/views/Entries/editor.cfm

Example Screen CF ORM Hibernate Model

component questbook.model.entry

PROPERTIES	
entry_id	[empty string]
entryBody	This is a message to display stuff on CF ORM
title	CF ORM
time	[empty string]
isCompleted	false
user	component questbook.model.User

PROPERTIES	
userID	3
userName	khanf
lastName	Khan
firstName	Faheem
userType	admin
entry	array

1 component questbook.model.entry

PROPERTIES	
entry_id	2c9011152887b956012887c415fc0067
entryBody	Generators Generators are algorithms that are used to generate unique identifiers for instances of the persistent CFC. Generators can have any one of the following values: increment: This algorithm generates identifiers of type long, short, or int by incrementing a counter maintained by ORM. This is commonly used when auto-generation for the primary key is not enabled in the table and you want ORM to generate the primary key. This should be used when a single instance of ColdFusion is the only process to insert data into the table. identity: For databases such as DB2, MySQL, Sybase, and MS SQL, which support identity columns, you can use identity key generation. The key is automatically generated by the database and the object is populated with the generated key. This strategy requires ORM to execute two SQL queries to insert a new object.
title	ORM: Generators
time	2010-05-11 10:28:14.46
isCompleted	NO
user	[see cfc2 for User details]

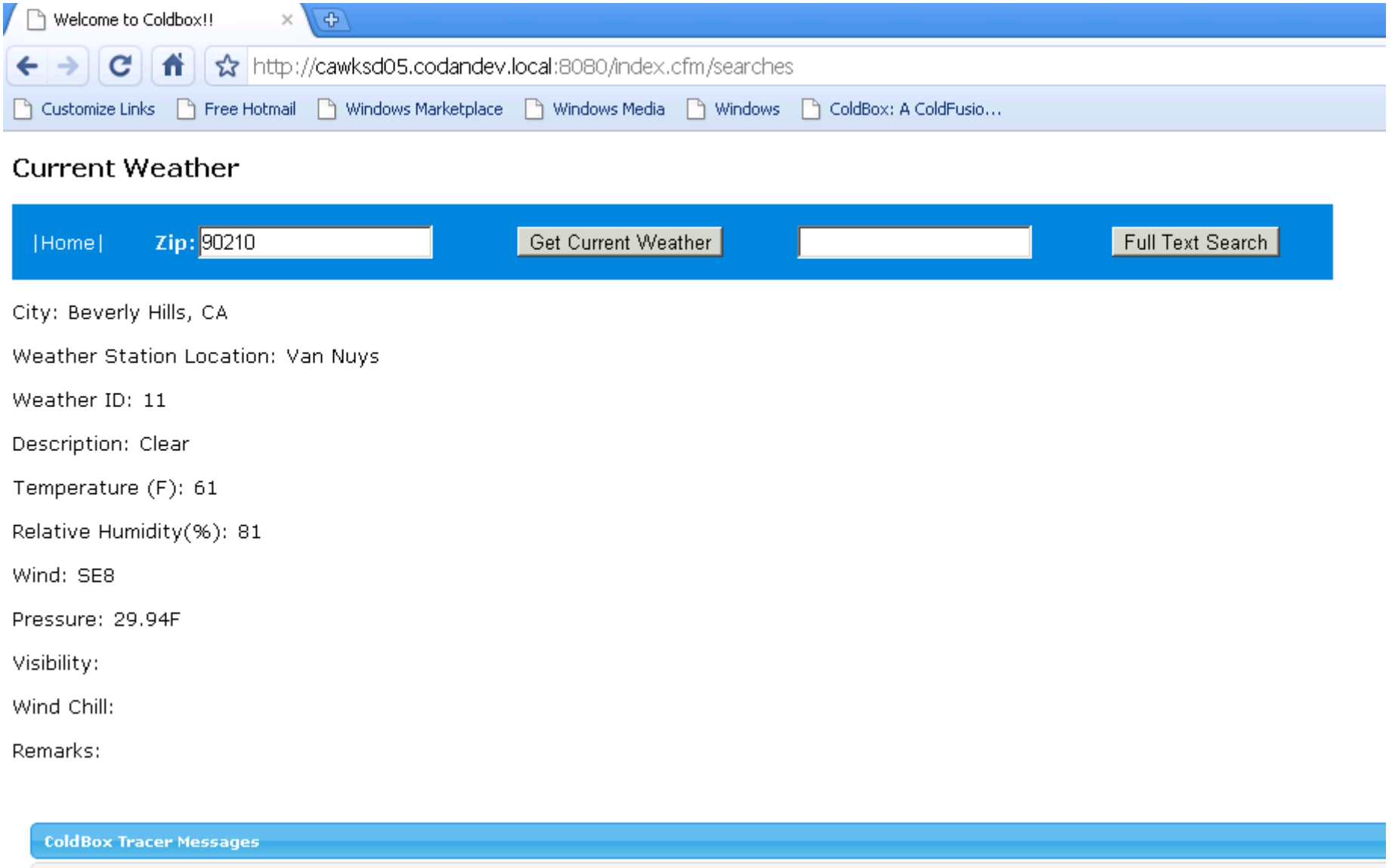
METHODS

METHODS

Example SOAP call with CF9

```
<cfinvoke webservice="http://ws.cdyne.com/WeatherWS/Weather.asmx?wsdl"
returnvariable="weather" method="GetCityWeatherByZIP"> <cfinvokeargument
name="zip" value="#rc.zipcode#">

</cfinvoke>

<h3>Current Weather</h3>

<cfinclude template="/includes/topbar.cfm">

<cfoutput>

<p>City: #weather.city#, #weather.state#</p>

<p>Weather Station Location: #weather.WeatherStationCity#</p>

<p>Weather ID: #weather.weatherID#</p>

<p>Description: #weather.Description#</p>

<p>Temperature (F): #weather.Temperature#</p>

<p>Relative Humidity(%): #weather.RelativeHumidity#</p>

<p>Wind: #weather.wind#</p>

<p>Pressure: #weather.pressure#</p>

<p>Visibility: #weather.visibility#</p>

<p>Wind Chill: #weather.windChill#</p>

<p>Remarks: #weather.Remarks#</p> </cfoutput>
```

Example SOAP call with CF9



Example Coding Handler

```
/*
*  Search handler
*/
component
extends="coldbox.system.EventHandler"
output="false"{

property name="entryService" inject;


function index(event) output=true{
var rc = event.getCollection();


if(rc.submit eq "Get Current Weather"){
event.setView("soaps/weather");

}

if(rc.submit eq "Full Text Search"){
rc.search =
entryService.search(event.getValue("searchText","
"));
event.setView("search");
}
}}
```

Future of Coldbox and CF

- Coldbox platform is in transition to becoming a POSS (professional open source software and services)
- Coldbox 3.0 is still relatively new and still needs to be more widely used and tested and enhanced in the future.
- There are currently talks of the following in the new version of Colfusion:

Current Desires for new CF Release

- Replace JRUN
- Fix server Monitoring
- Fix application deployment
- Make a free edition
- Update Web Services Engine
- Adding a workflow engine

Main Benefits of CF9

- Language specification Enhancements in CFSCRIPT –

  unlike previous versions of CF you can now code CF components entirely in CFSCRIPT and also run SQL queries from within CFSCRIPT as I have demonstrated inside my pilot.

- The ORM lets you access and update data through the application using the object model, without you needing to know anything about the details of the underlying database structure.

Why we should not go with Coldbox 3.0 ?

- Lack of documentation on CF ORM – CF ORM was a major part of my pilot.
- Lack of online support in forums
- Only 1 sample application for Coldbox available which shows CRUD on CF ORM.
- Installing MXUnit testing had destroyed some of my links to pages that I had in place.
- Only one screen video cast available online showing how to code CRUD app using coldbox 3.0.

Why we should not go with Coldbox 3.0 ? (Cont)

- Not widely used so people know little about this framework
- Not much out there on RBAC. - Checked forum and others experienced similar issues and stressed their frustration.
- The “Tons of samples”[1] found on coldbox, though when tested have to be manually configured to work and there is no documentation stating this

[1] http://www.coldboxframework.com/index.cfm/download/samples

Question Time

Any Questions ?

Resources

- http://www.coldboxframework.com/support/paid
- http://www.akbarsait.com/cf9tutorials.cfm#ORM
- http://www.bennadel.com/blog/1669-Learning-ColdFus
- http://www.bennadel.com/blog/1674-Learning-ColdFus
- http://forums.coldbox.org/index.cfm
- http://stackoverflow.com/
- http://www.coldboxframework.com/index.cfm/downloa
- http://wiki.coldbox.org/wiki/Models.cfm