Sam Farmer

Growing up I never imagined I would play bass guitar for the Dave Matthews Band. And indeed it never happened.

But I have become a passionate and pretty good web developer.

5 Ways To Improve Your App with ORM

>Me_

Web developer 15 years

ColdFusion (mainly) and HTML, JavaScript, jQuery

Live in Washington, DC with wife and two daughters

Adobe Community Professional

Work for FirstComp Insurance

Checklists (16applications.com/checklists)

>You_

1.: Versioning

Data Got Overriden

To: Sam

Cc:

Subject: Data Got Overriden

smtp.gmail.com

Sam,

It happened again. Both Jim and Deb where using the app at the same time and Deb overwrote Jim's work.

Jim is pissed as his was the correct record and Deb wants to stab her eyeballs out with a pencil!!!!!!

Please fix ASAP.

Thanks,

Jane

>1: Versioning_

- Gets updated when the data changes
- Allows for optimistic locking
- Either number incremented (version) or timestamp updated

>1: Versioning_

- Extra secure: Consider encrypting the version information in the form
- Other options: Possible to set optimistic-lock on the cfc

>1: Versioning_

Entity:

```
property name="ormVersion"
fieldtype="version|timestamp"
```

>1: Versioning_

Code:

```
<input type="hidden"
name="ormVersion" id="ormVersion"
value="#character.getOrmVersion()#"
>
```

>1: Versioning_

Code:

```
if ( form.ormVersion <
character.getOrmVersion() ) {
  writeOutput( "This record has been
edited elsewhere since you last opened
the page" );
  abort;
}
```

>2: Events_

- pre and post events for when interacting with entities

preInsert(entity)	postInsert(entity)
preUpdate(entity, Struct oldData)	postUpdate(entity)
preLoad(entity)	postLoad(entity)
preDelete(entity)	postDelete(entity)

>2: Events_

- Needs to turned on

```
component {
...

this.ormEnabled = true;
this.ormSettings = { eventHandling=true,
                     eventHandler="ORMEvents" };
}
```

turns it on

optional, points to a cfc that runs for every persistent cfc

>2: Events_

- eventHandler is great for global settings

```
component implements="CFIDE.orm.IEventHandler" {

    public void function preLoad ( any entity ) {}
    public void function postInsert ( any entity ) {}
    public void function postUpdate ( any entity ) {}
    public void function postLoad ( any entity ) {}
    public void function preDelete ( any entity ) {}

    public void function preUpdate ( any entity, Struct oldData ) {
        if ( structKeyExists( entity, "updateDate" ) ) {
            entity.setUpdateDate( now() );
        }
    }
    public void function postDelete ( any entity ) {}
    public void function preInsert ( any entity ) {
        if ( structKeyExists( entity, "createDate" ) ) {
            entity.setCreateDate( now() );
        }
    }

}
```

>2: Events_

- Use within a persistent cfc

```
component persistent="true" {
property name="orderId" ormtype="int" fieldtype="id" generated="always"
generator="native";
property name="product" fieldtype="many-to-one" cfc="product";
property name="quantity" ormtype="int";
property name="createDate" ormtype="timestamp";

function preInsert() {
    var inStock = getProduct().getInStock();
    getProduct().setInStock( inStock-getQuantity() );
    getProduct().setUpdateDate( now() );
}

}
```

>2: Events_

- When using both order is:

1. preXXX() in persistent cfc
2. preXXX() in global eventHandler
3. save to database
4. postXXX() in persistent cfc
5. postXXX() in global eventHandler

>2: Events_

- Prevent deletes
- preDelete(entity) { return true }

>2: Events_

Event Handling allows for the entity to define not just **what** is in the entity but **how** the entity can be interacted with

>3: MappedSuperClass_

- New in 9.0.1
- Is extended by persistent cfc
- And can contain properties
- As well as functions

>3: MappedSuperClass_

- Ensure good audit information

```
component mappedsuperclass="true" {

property name="createdDT" ormtype="timestamp";
property name="createdBy" ormtype="int";
property name="modifiedDT" ormtype="timestamp";
property name="modifiedBy" ormtype="int";

}
```

>3: MappedSuperClass_

- And have a cfc extend it:

```
component persistent="true" extends="base" {
properties..
}
```

>3: MappedSuperClass_

- Go big:

```
component mappedsuperclass="true" {

property name="id" ormtype="int" fieldtype="id"
generated="always" generator="native";

property name="isActive" ormtype="boolean" default="true";
property name="isDeleted" ormtype="boolean" default="false";

property name="createdDT" ormtype="timestamp";
property name="createdBy" ormtype="int";
property name="modifiedDT" ormtype="timestamp";
property name="modifiedBy" ormtype="int";

property name="ormVersion" fieldtype="version" ormtype="int";
}
```

>3: MappedSuperClass_

- And leave your actual entities clean:

```
component persistent="true" extends="base" {

property name="maxYear" ormtype="string";
property name="band" ormtype="string";
property name="mc" ormtype="string";

}
```

>4: Secondary Cache_

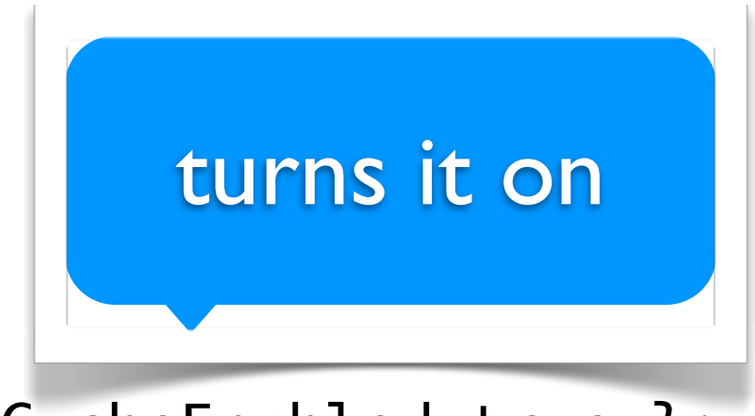
- Caching is a great way to improve performance
  - But remember some limits
- Secondary cache works with ORM
  - By default ehCache

>4: Secondary Cache_

- Needs to turned on

```
component {
...

this.ormEnabled = true;
this.ormSettings = { secondaryCacheEnabled=true };
}
```



>4: Secondary Cache_

Shut the server down

Shut it down now!

I had to fire Bob but he still has access until he logs out and is deleting records.

>4: Secondary Cache_

- Four types:
  - transactional
  - nonrestrict-read-write
  - read-write
  - read-only

>4: Secondary Cache_

- Transactional
- Creates key-value mapping for how you obtain the data
- When updating saves data to the database AND the cache
- Retrieval from ehCache is wicked fast
  - ~1 microsecond

>4: Secondary Cache_

```
component persistent="true" cachename="elmo"
cacheuse="transactional" {

property name="id" ormtype="int" fieldtype="id"
generated="always" generator="native";
...
}

entityLoad("User", id, true);
```

>4: Secondary Cache_

- read-only
- useful for look up data

```
component persistent="true" readonly="true"
cacheuse="read-only" cachename="stateCache" {
property name="stateId" fieldtype="id" ormtype="int"
generator="native" generated="always";
property name="name" ormtype="string";
}
```

>4: Secondary Cache_

- Three functions to evict:
  ORMEvictCollection()
  ORMEvictEntities()
  ORMEvictQueries()

>5: Struct Relationships_

- Perfect spot for a bad joke about relationships

>5: Struct Relationships_

- Set-up:

```
component persistent="true" entityname="user" {
property name="userId" fieldtype="id" generated="always" generator="native";
property name="name" ormtype="string";
property name="address" fieldtype="one-to-many" cfc="address" fkcolumn="userId"
inverse="true" type="struct" structkeycolumn="type";
}

component persistent="true" entityname="address"  {
property name="addressId" fieldtype="id" generated="always" generator="native";
property name="type" ormtype="string";
property name="user" fieldtype="many-to-one" cfc="user";
}
```

>5: Struct Relationships_

- The addXXX() takes a key value

```
work = entityLoad("address", 1, true );
user = entityNew("user", {name="Homer Simpson"} );
user.addAddress("work", work);

work.setUser(user);
```

there

>5: Struct Relationships_

- After load can use structKeyExists:

```
<cfif structKeyExists( user.getAddress(), "work" )>
    <cfdump var="#user.getAddress()["work"]#" top="1">
</cfif>
```

▶

>5: Struct Relationships_

- Performance better on one-to-many where the many, has, um, many records
- set lazy=”extra”

>Resources_

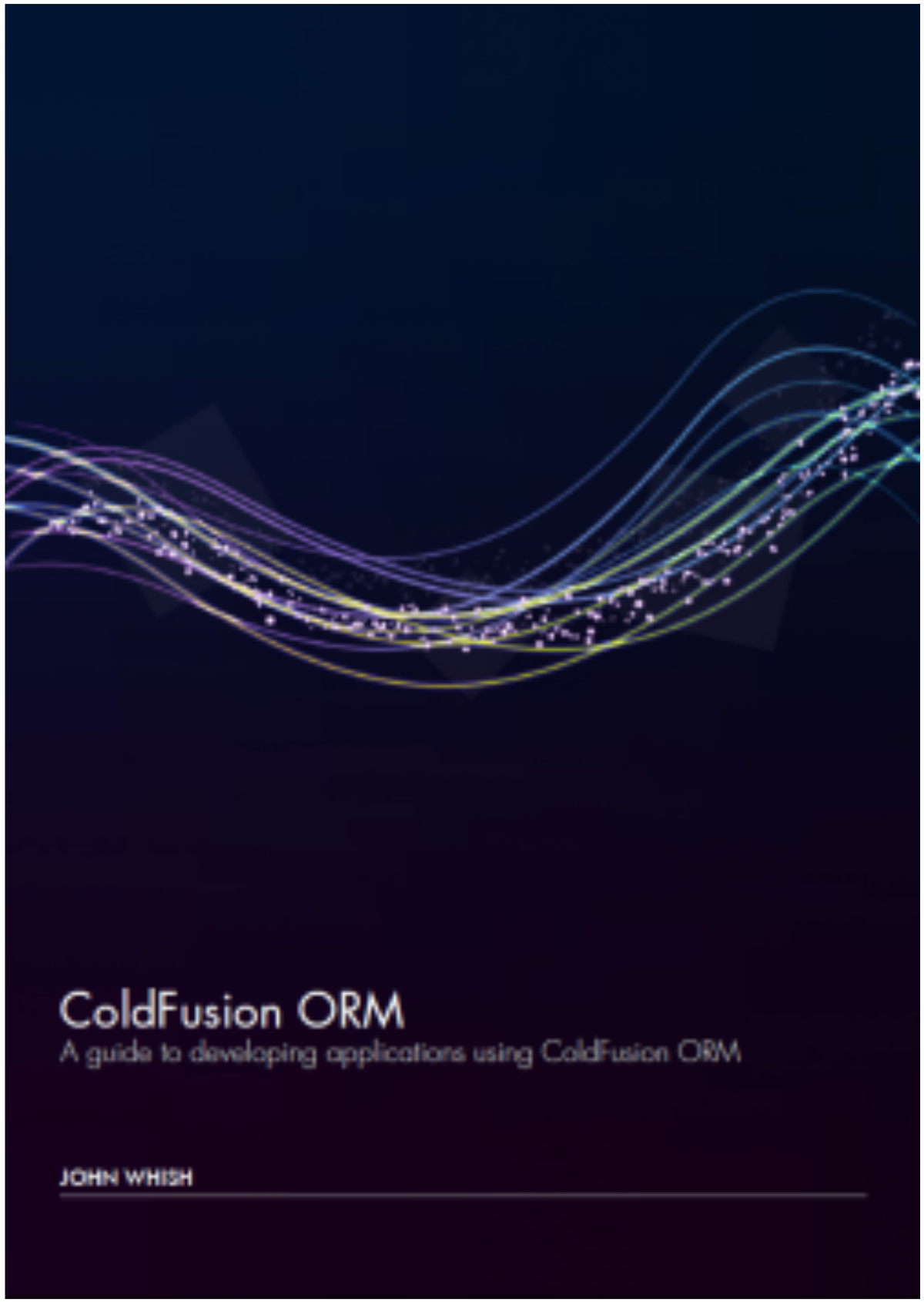
cf-orm-dev on Google Groups:

http://groups.google.com/group/cf-orm-dev

>Resources_

Book

http://tinyurl.com/cf-orm-book



>Resources_

ColdFusion 9.0 Resources

- Configuring and Administering Adobe ColdFusion 9
- Adobe ColdFusion 9 CFML Reference
- Developing Adobe ColdFusion 9 Applications
- Installing Adobe ColdFusion 9
- ColdFusion 9 ActionScript Language Reference

Developing ColdFusion 9 Applications

- What's New in ColdFusion 9
- Introduction
- Introducing ColdFusion
- The CFML Programming Language
- Building Blocks of ColdFusion Applications
- Developing CFML Applications
- Accessing and Using Data
- ColdFusion ORM

Developing Adobe ColdFusion 9 Applications

Web Help

Welcome to Adobe® ColdFusion® 9 Help on the web, part of the Adobe Community Help enviro comments from users and expert moderators provide additional tips and solutions. You can sea you are not connected to the Internet, a subset of Help opens.

Community Help

Community Help brings together Help, instruction, support articles, videos, blogs, and examples The Adobe custom search engine helps you find the best content available whether it's produce product Help are included whenever you search Community Help.

To access Community Help, go to http://www.adobe.com/go/lr_ColdFusion_community_en.

Q & A

samfarmer.com
@sam_farmer