

Advanced Model-Glue



alagad

120 Nelson Lane, Clayton, NC 27527 | p 1 888 ALAGADA | f 1 888 248 7836 | www.alagad.com | info@alagad.com

Doug Hughes, President

What You Will Learn

- Generic database messages
- Scaffolding your UI using database metadata
- The include tag and ActionPacks
- How to tweak Model-Glue using ColdSpring

120 Nelson Lane, Clayton, NC 27527 | p 1 888 ALAGADA | f 1 888 248 7836 | www.alagad.com | info@alagad.com



Generic Database Messages

- Model-Glue comes with generic messages you can broadcast to interact with your database
- Works with Reactor or Transfer ORMs
- Very handy and time-saving

120 Nelson Lane, Clayton, NC 27527 | p 1 888 ALAGADA | f 1 888 248 7836 | www.alagad.com | info@alagad.com



ModelGlue.GenericList

- Lists records for the specified table

```
<message name="ModelGlue.GenericList">  
  <argument name="object" value="Customer" />  
  <argument name="orderBy" value="LastName" />  
</message>
```

- This will create a query in the event and viewstate named CustomerQuery ordered by the LastName field.

120 Nelson Lane, Clayton, NC 27527 | p 1 888 ALAGADA | f 1 888 248 7836 | www.alagad.com | info@alagad.com



ModelGlue.GenericList Arguments

Object

- The object type to list

QueryName (optional)

- The name of the query.
- Defaults to Object & Query (CustomerQuery)

Criteria (optional)

- A list of values from the event to use as where criteria

OrderBy (optional)

- A column to order the query by

Ascending (optional)

- If true, indicates if the order is ascending

ModelGlue.GenericRead

- Reads a Record for the specified table

```
<message name="ModelGlue.GenericRead">
  <argument name="object" value="Customer" />
  <argument name="criteria" value="customerId" />
</message>
```

- This will create a Record in the event and viewstate named CustomerRecord loaded based on the customerId event value.

ModelGlue.GenericRead Arguments

Object

- The object type to read

RecordName (optional)

- The name of the resulting Record.
- Defaults to Object & Record (CustomerRecord)

Criteria (optional)

- A list of values from the event to use as criteria when loading the Record
- If not provided or not matched a new Record will be created

ModelGlue.GenericCommit

- Commits a Record to the database

```
<message name="ModelGlue.GenericCommit">
  <argument name="object" value="Customer" />
  <argument name="criteria" value="customerId" />
</message>
```

- This will save a CustomerRecord to the database based on the customerId event value.

ModelGlue.GenericCommit Arguments

Object

- The object type to commit

Criteria (optional)

- A list of values from the event to use as criteria when committing the Record
- If not provided or not matched a new Record will be created

RecordName (optional)

- The name of the resulting Record.
- Defaults to Object & Record (CustomerRecord)

ValidationName (optional)

- The name of the resulting validation structure.
- Defaults to Object & Validation (CustomerValidation)

Properties (optional)

- A comma separated list of properties to try to commit from the Event.
- Defaults to all properties in a Record

ModelGlue.GenericCommit Results

- ModelGlue.GenericCommit will add one of two results. You can map them to events as you want:

commit

- This is added when the Record is successfully committed.

validationError

- This is added when there are errors validating the Record before committing it.

ModelGlue.GenericDelete

- Deletes a Record from the specified table

```
<message name="ModelGlue.GenericDelete">
  <argument name="object" value="Customer" />
  <argument name="criteria" value="customerId" />
</message>
```

- This will delete a Record from the Customer table based on the customerId event value.

ModelGlue.GenericDelete Arguments

Object

- The object type to delete

Criteria (optional)

- A list of values from the event to use as criteria when deleting the Record

Scaffolding CRUD

- Model-Glue can automatically generate basic user interfaces for CRUD functions
- The term Scaffolding comes from construction
- A rough outline of your application
- Useful for quick CRUD forms

<scaffold>

- Defines event handlers for the specified object
- The <scaffold> tag is an extension of the <event-handlers> tag and can hold all the same children.
- Generates List, View, Edit, Commit, Delete events in the format Object.Event. For example: Customer.List

```
<scaffold object="Customer" />
```

Viewing Scaffolds

- To view a scaffold go to a scaffold event such as Customer.List
- <http://localhost/index.cfm?event=Customer.List>

Broadcasts, Results and Views

- Because the <scaffold> tag simply is an extension on the <event-handler> tag it can hold all of the same child tags.
- Easily allows use of results for templates, etc.
- Easily allows for use of messages for security, etc.

Scaffold Only What You Want

- Configure Model-Glue to generate only the scaffolds you want:

```
<property name="defaultScaffolds">
  <value>list,edit,view,commit,delete</value>
</property>
```

- Or, use the Scaffold tag type attribute to indicate what should be generated:

```
<scaffold object="Customer" type="List,View">
```

Customize Scaffolds

- Scaffolds are generated into that path configured in ColdSpring.xml
 - generatedViewMapping
- This is always the last directory Model-Glue looks in for a view
- Copy the generated view to another view directory and modify it
- Changes to your database will not be reflected

The <include> Tag

- Allows developers to include entire Model-Glue applications.
- Applications are merged together.
 - Unlike using ModelGlue_APP_KEY
- Allows for creation and usage of "ActionPacks".

The <include> Tag

- Used directly under the <model-glue> tag.
- Includes a ModelGlue.xml file.
- Imports all of the event handlers and configuration as if, combined, they were one big XML file.

```
<include template="//ActionPackExample/Fortune/config/ModelGlue.xml" />
```


Include Tips

- Create one root application that includes additional XML files.
- Prefix event names and views with the included application name.
(Fortune.ShowFortune or Fortune.dspFortune.cfm)
- Use <config> tags to add additional ColdSpring.XML files and view mappings.

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



<config> Tag

- Used only with ActionPacks
- To include additional ColdSpring configuration
- To include additional view mappings.

```
<config>
  <setting name="beanMappings"
    value="/ActionPackExample/Fortune/config/ColdSpring.xml" />
  <setting name="viewMappings"
    value="/ActionPackExample/Fortune/views" />
</config>
```

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



ActionPacks

- There are a few ActionPacks available in the wild.
- Model-Glue comes with one ActionPack
 - Email
- There may be a market for commercial ActionPacks

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



Customize The Model-Glue Framework

- Model-Glue is configured by ColdSpring
 - /modelglue/unity/config/Configuration.xml
- Model-Glue's Configuration.xml is loaded before your ColdSpring.xml
- Settings in your Configuration.xml override those in the Configuration.xml
- Allows you to wire in your own versions of Model-Glue core files

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



Customize The Model-Glue Framework

- Because you can customize the core framework you can change how parts of the framework behave.
- For example, you can change the generic lists and deletes to use soft deletes.
- Difficult, but can theoretically work for any part of Model-Glue
- May or may not be forward compatible through major versions

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



Things You Didn't Learn

- How to integrate Model-Glue with Flex
 - The Flex integration stuff is currently checked into Subversion
- How to announce asynchronous messages
 - Uses Sean Corfield's concurrency library

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



What You Learned

- How to use Generic Database Messages
- How to use Scaffolding
- Ways to customize Model-Glue
- Some more areas for study

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



Questions and Answers

120 Nelson Lane - Clayton, NC 27527 | p 1 888 ALAGADn | f 1 888 248.7636 | www.alagad.com | info@alagad.com



The End!

- Thank you very much for attending Enterprise ColdFusion with Model-Glue.
- I hope it was all you expected and more!