

# Behold Model-Glue users! MG+AJAX Made easy

Posted At : March 20, 2006 12:14 AM | Posted By : Rob Gonda

Related Categories: [ajax](#), [ajaxCFC](#), [Coldfusion](#)

I just released a new improved version of [ajaxCFC](#) for the [Model Glue framework](#). Now it's even easier to add Ajax to your robust ColdFusion MVC applications. I modified the [Echo Example](#) and added a [Contact Manager Example](#) to show how it works. All open source and included in the [ajaxCFC download](#). The contact manager example is the same that comes with Model-Glue, using the same controller, models, and views ... slightly Ajaxified :)

*[Updated] IMPORTANT NOTE: The XML Socket works on port 1225, so if you're behind some firewall preventing that traffic you will not see the refreshes. I am still working on some port 80 proxy, but for this experiment just open that port.*

btw, I will be demonstrating all this at the upcoming [Spring Conference](#), this week.

If that wasn't enough, included is also the next level of Web 2.0 applications; a mash-up of ColdFusion / Model Glue / ajaxCFC / XML Binary Sockets / CF Event Gateways... This combinations allows you to only for your presentation layer to interact with the server in the background, but also for the server to push data to the clients. No more Ajax pulls every 3 seconds interval, now the server can just broadcast the command to all clients. I shall call it Apocalypse. ;)

I modified the [Contact Manager example](#) that comes with model glue to use Ajax and Socket servers... now every time anyone adds/edits/deletes a contact all clients' views will get updates... just imagine the possibilities. Just try it yourself... open two browsers and see them interact.

I will try to briefly explain the framework the best I can.

First, you need the BER version of the model glue framework; Joe still hasn't officially released MG v1.1.

Now, the framework consists of a model: `ajax.cfc`, some JavaScripts included in the `views/js` folder, and a rendering view `ajaxResponse.cfm`. That's it...

How do it work...

The examples that come with Model-Glue use a private Layout handler, so I just added two more: `AjaxLayout` and `SocketLayout`... Now every time you need to respond an Ajax request you use the `AjaxLayout`, and if you want to broadcast something through the socket, you use the `SocketLayout`.

I modified the controllers `init` and `OnRequestStart` to parse the Ajax request... once it gets to your controllers handler all Ajax vars will be sent to you as regular event values, which you can get with `getValue`. There's not much more you need to know there...

Your controller will do its thing and set vars to be used in `modelglue.xml` ... now, here's the beauty ... when you use the `AjaxResponse` renderer your views will modify DIV Layers, or anything with the view's ID ... For example, if your body has a container calls `contentForm`, once you render a view with that name, all content will get placed inside that DIV...

If you need to run scripts you can use a reserved view name: `'script'`. If you create a view with that name, all code

inside that view will get evaluated in the client side. I modified the see what I mean.

[Echo Example](#) to use this method so you can

The socket implementation is still in alpha mode... I just finished the prototype today... how does it work?!

I added a 1px Flash movie that weights only 1k that all it does is to open a socket connection to a ColdFusion Event Gateway. I rewrote the XML Binary sockets that comes with CFMX7 to better suite the Flash XML Socket protocol. I also added a few methods such as onConnect and onDisconnect. Anyways, when you need to talk to the server to do it through Ajax, but if the server needs to talk to you, it will do it through the XML Gateway. The beauty is that you do not need to modify anything or create any custom handler for it ... The flash movie uses FS Command to communicate with JavaScript and triggers the same calls the Ajax response would, so the same rendering in the server works for either method.

I included the org.gonda.socket.jar file, which would have to be registered as a custom event gateway to make the contact manager example to work. Just point your JVM class path where the file is, restart ColdFusion, and register a new gateway:

Name: FlashSocket

Class: org.gonda.socket.SocketGateway

Then create a new instance and call it 'flashSocket1'. I provided the cfg file in the example too.

In the main view, both IP and port are passed to the Flash file as FlashParams... will easily see how to modify it.

I know, you are thinking it's too much, but it's not THAT complicated ... and it's well worth it ...

Ok, so what is the catch? There are a couple of issue I still need to work on ... one) The socket binds itself to an IP and port, and I don't want to create a new socket for every application.. especially if I need to open a new IP on my firewall every time ... two) firewall... yes, I need to find a way to perform http tunneling, or to detect that it was blocked on run time to shift all socket calls to Ajax.

I know it's not crystal clear, but I wanted to get it out there and get people thinking ... I'll try to document it better this week before the conference.