

Swiz RPC Library

Posted At : November 20, 2009 2:25 AM | Posted By : Nathan Mische

Related Categories: SwizRPC, Swiz, Frameworks, Flex

A couple of weeks ago [Joe Rinehart blogged about his RemoteMethodMediator](#) for Swiz. Like Joe, I've found that most of my executeSeviceCall callback methods were just dispatching events via Swiz, so I liked the idea of skipping callbacks completely. However, I didn't really like Joe's implementation, so I yesterday I did some playing around and came up with the a [Swiz RPC library](#). The library has swiz specific implementations of mx.rpc.remoting.mxml.RemoteObject, mx.rpc.remoting.mxml.Operation, and mx.rpc.CallResponder. These classes can be used in your bean loader in place of the standard mx:RemoteObject, mx:method, and mx:CallResponder tags.

I haven't done a whole lot of testing yet, but I thought I'd put this out and see what people thought. The swizrpc:RemoteObject and swizrpc:method tags are good if you want your remote method calls to always dispatch the same events. If you need to have different invocations of a remote object method fire different events, then the swizrpc:CallResponder may be what you need. Here is a rundown of each of the classes and their properties. There are also links to two small sample apps that show how to use these components.

RemoteObject

This extends the core mx.rpc.remoting.mxml.RemoteObject, and can be used via MXML with the swizrpc:RemoteObject tag. If you use this component and do not define any of the swizrpc specific properties it will work just like the standard RemoteObject component, with one exception: if a method with no fault handler is called and a fault occurs, a DynamicEvent of the type "org.riaforge.swizrpc.rpc.events.SwizFaultEvent" is dispatched by Swiz. This event has fault and faultEvent properties which hold the fault string and the entire faultEvent respectively.

Swiz Specific Properties

faultEvent (String)

the type of DynamicEvent to dispatch on fault. Defaults to "org.riaforge.swizrpc.rpc.events.SwizFaultEvent". This event type is available as a the SWIZ_FAULT_EVENT static constant of the RemoteObject class.

faultProperty (String)

the name of the property to put the FaultEvent.fault value into. Defaults to "fault".

resultEvent (String)

the type of DynamicEvent to dispatch on result.

resultProperty (String)

the name of the property to put the ResultEvent.result value into. Defaults to "result".

Operation

This extends the core mx.rpc.remoting.Operation, and can be used via MXML with the swizrpc:method tag. If you use this component and do not define any of the swizrpc specific properties it will work just like the standard Operation (mx:method) component.

Swiz Specific Properties

faultEvent (String)the type of DynamicEvent to dispatch on fault. faultProperty (String)the name of the property to put the FaultEvent.fault value into. Defaults to "fault". resultEvent (String)the type of DynamicEvent to dispatch on result. resultProperty (String)the name of the property to put the ResultEvent.result value into. Defaults to "result".

CallResponder

This extends the core mx.rpc.CallResponder, and can be used via MXML with the swizrpc:CallResponder tag. If you use this component and do not define any of the swizrpc specific properties it will work just like the standard CallResponder component.

Swiz Specific Properties

faultEvent (String)the type of DynamicEvent to dispatch on fault. faultProperty (String)the name of the property

to put the `FaultEvent.fault` value into. Defaults to "fault". `resultEvent (String)` the type of `DynamicEvent` to dispatch on result. `resultProperty (String)` the name of the property to put the `ResultEvent.result` value into. Defaults to "result".

Example Apps

Here are a couple of very simple sample apps. Right click to view source.

Using RemoteObject

Using CallResponder

If you use Swiz and think these components would be useful, please let me know.

In addition to [Joe Rinehart](#), thanks go to [Brian Kotek](#) for helping sort out an issue I was having getting my `RemoteObject` to play nicely with Swiz's `ExpressionUtils`.

Update: The SWC is now available for download from RIAForge. <http://swizrpc.riaforge.org>