Weather Web service (flex) - Part 2

Posted At : 22 May 2008 13:34 | Posted By : Shaun McCran
Related Categories: Flex Remoting, RIA, Flex

In the first part of this article, we created the coldfusion back end (cfc's). The first thing we'll do in the Flex section is setup our initialize function which will call a Remote Object, to interface with the cfcs.

```
/* Init the app
*/
private function init():void
{
RO.populateCombo();
}
```

I prefer using Remote Object, as it fits very nicely with coldfusion, and its very easy to setup, for great results. There is supposed to be a speed difference too, but I can't honestly say I'd notice.

```
<mx:RemoteObject destination="ColdFusion"
id="RO"
fault="Alert.show(event.fault.message)"
source="cfc.weather"
showBusyCursor="true">

ethod name="populateCombo" result="handleCombo(event)" fault="Alert.show(event.fau
thod name="getForecast" result="handleForecast(event)" fault="Alert.show(event.fau

</mx:RemoteObject>
```

Above we have a Remote Object call. Specify a destination of ColdFusion, and point the source at your cfc. Remember that its dot notation. (IE folder.folder.cfcName).

Then we have a separate method call for each of the cfc methods. Each pointing at their own actionScript handler.

So our init function will call the Remote Object, which will populate the combo box. When an item is selected in the combo, we fire off the callForecast() method.

```
                    /* Handles the combo results event
                                    */
                       private function handleCombo(event:ResultEvent):void
                                                    {
                                locationCombo.dataProvider = event.result;
                                                    }

                           /* Function to call forecast RO from combobox
                                                    */
                                private function callForecast():void
                                                    {
                               RO.getForecast(locationCombo.selectedItem.data, _uni
                                                    }


="103" y="38" id="locationCombo" change="callForecast()" prompt="Select Location">
```

The Remote Object passes the country code back to our cfc, which actions the cfhttp call. This is returned as xml to our Flex app, and handled in this function:

```
                     /* Handles the forecast result
                                                    */
                    private function handleForecast(event:ResultEvent):void
                                                    {
                                    _xmlBlock = XML(event.result);
                                                // label data
                             var labelText:String = _xmlBlock..item.title;
                                     returnLabel.text = labelText

                                                // get link url
                              var link:String = _xmlBlock..channel.link;
                                                    _link = link

                                var desc:String = _xmlBlock..item.description;
                                     returnDescription.htmlText = desc;
```

```
                                        // turn on hidden elements
                                        linkButton.enabled = true;
```

This formats the response as xml, and maps it out to display elements on screen.

Lastly we have a button that will take a user through to the full forecast, this is a standard Flex UrlRequest:

```
                    /* url handler for button
                                          */
                            private function goToUrl():void
                                          {
                                var url:URLRequest = new URLRequest(_link);
                                      navigateToURL(url,"_blank");
                                          }

on x="10" y="497" id="linkButton" label="Full Forecast" click="goToUrl()" enabled="
```

You can download the full source here .