Stratus™ Proxy RIA Developer Guide

October 2009

Table of Contents:

Introduction

A web application, containing any of the Stratus RIA Controls, will typically be hosted on the same server as the controls themselves. However, it is possible that a web designer/developer may wish to work on such a web application temporarily by hosting it on their local machine. This would normally cause cross-domain scripting problems, but these can be avoided by installing the RIA Proxy Web Service on the local machine, and having the local web application request the RIA controls through the proxy rather than from the server directly.

The RIA Proxy Web Service can proxy JavaScript, CSS, images and XML documents.

RIA Proxy Web Service

The proxy web service allows a web application to request JavaScript, JSON and xml files which are located on another domain. The proxy service solves the problem of cross site scripting as the request is performed server side and the response is forwarded back to the calling thread.

Requirements

- You have installed the Stratus RIA controls and RIA Controller on a network accessible location.
- You have some JavaScript skills.
- You have the ability to produce styling via CSS stylesheets for the RIA controls to be used.
- You have installed IIS (5.0 or 6.0+) and will be serving your page from within the same context as the location of the RIA Proxy Web Service.

Sample URL

An example of the practical usage of the Proxy web service is as follows:

http://localhost/RIAProxyService/Proxy.aspx?req=http://localhost:8080/test-harness/widgets/stratus

It is a good idea, once the Proxy Web Service has been setup with IIS, to test that the proxy is working using the following URL (for example):

http://localhost/RIAProxyService/Proxy.aspx?req=http://www.google.com

This should display the webpage specified after the URL parameter in your browser. It is not an issue if there appear to be images missing.

Parameters

The Proxy Web service currently only accepts a single parameter:

req   The desired URL to proxy

Description / Usage

The Proxy Web Service needs to be installed on the same context domain as the custom application that needs to access Stratus RIA controls and controller. The Proxy Web Service is utilized to act as a relay to Stratus RIA and/or the Stratus RIA controller which are located on a different context domain, and removes the issue of cross domain scripting.

There are a few different scenarios that the Proxy Web Service can be utilized with, including a partial cross-domain solution, and a fully cross-domain solution. It is important to note that use of the Proxy Web Service is intended only for testing. A final, customer-facing solution should involve the web application being hosted on the same server as the RIA Controls. Any scenario where the web application is hosted somewhere different to the RIA Controls would potentially require much more maintenance.

The following detailed scenarios are models of a set of Sample HTML files that demonstrate how the Proxy Web Service could be used.

Sample 1

This scenario deals with producing a web application that uses Stratus RIA controls cross-domain, while at the same time utilizing a local (or 'X-Domain') Dojo instance with optional third party controls. This scenario also provides the option for you to use your own local CSS and images for the Stratus RIA controls that ensures you retrieve images and styling for the controls in a more stable manner.
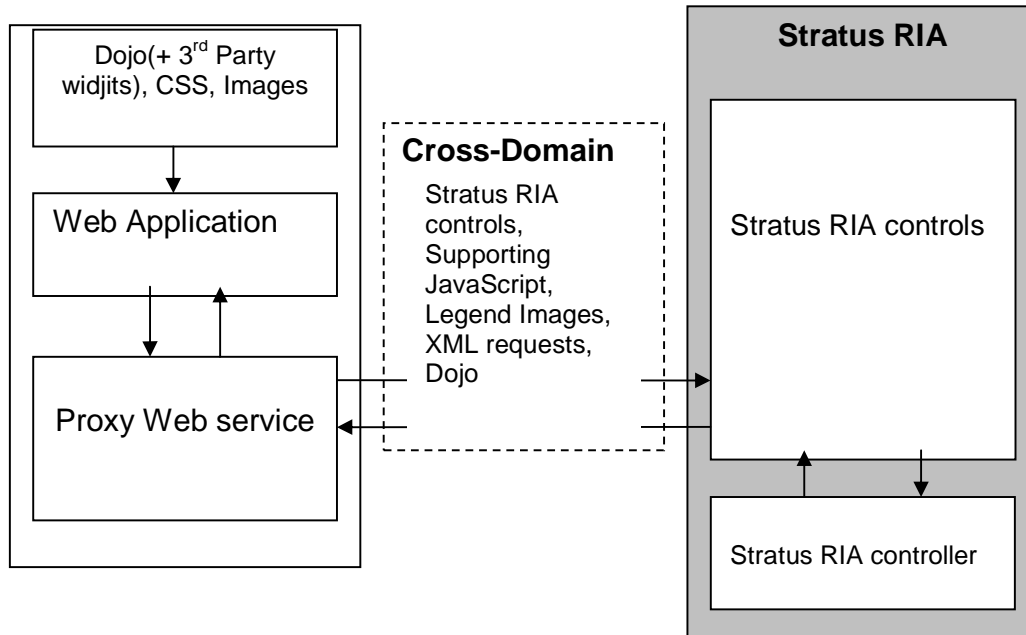
Sample 2

This scenario deals with a complete, independent web application with all components of the Stratus RIA controls (and their dependencies) being extracted from an alternate domain.
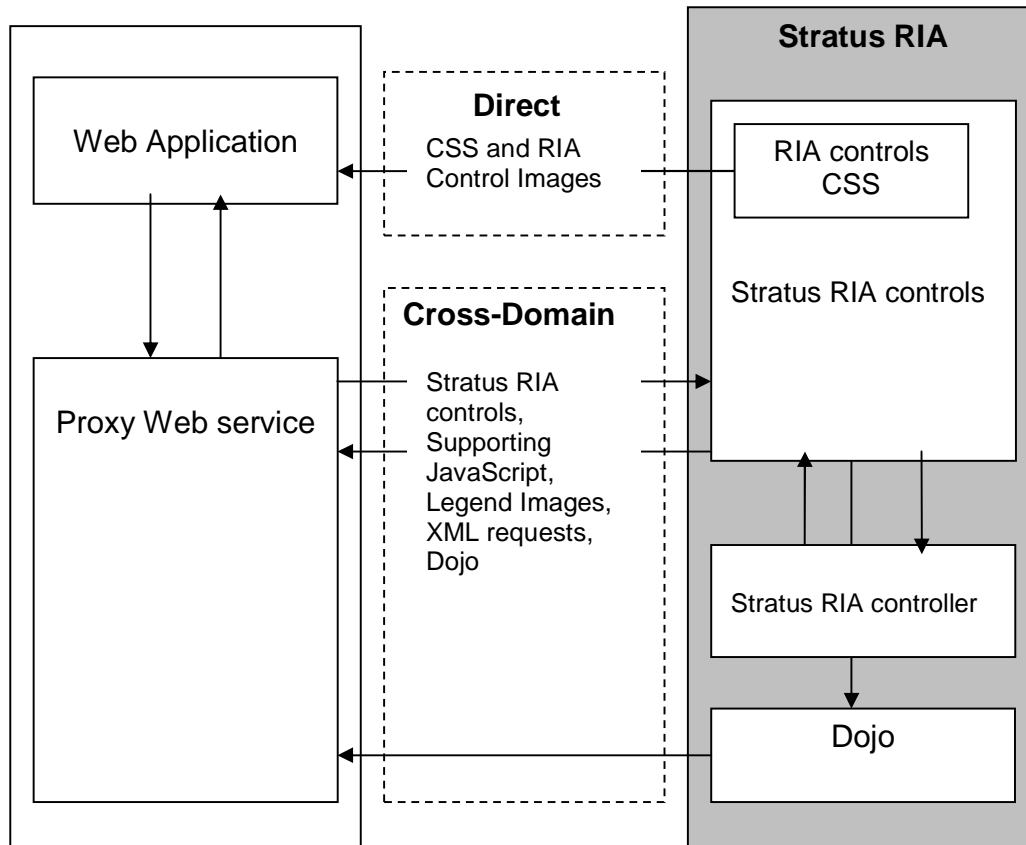
Sample 3

Sample 3 is a scenario very similar to Sample 1, albeit with the Stratus RIA controls on the same domain as the Web Application, though the Stratus RIA Controller is contacted cross-domain. This is implemented almost the same as Sample 1, though the Stratus RIA controls are registered with dojo using a relative path instead of a proxy path.

Limitations and Recommendations

Developing a completely cross-domain HTML page with RIA Controls requires more effort than producing a common-domain solution, and introduces a greater likelihood of errors occurring.

It is highly recommended that you initially develop with Firefox and Firebug in order to minimise issues and locate errors that may occur.

Ideally, **this solution should only be used for development purposes only**, with the eventual goal of the solution produced being ported into the same domain context as the RIA Controls and RIA Controller. Using this cross-domain solution as the final product will more than likely compromise the integrity of your web environment and the product itself.

Basic steps to create a Cross-Domain solution

Most of the preparation work for developing a cross-domain solution will involve additions to the `<head>` section of an HTML page.

Initially, however, we have to make a mandatory change to the contextpath.js file (found in the same directory as the proxy) in order to define the proxy and the location of the Stratus RIA controller.

Step One: Defining the proxy context

Contextpath.js initially may look similar to the following:

```
dojo.provide("dojo.contextpath");
contextPath =
"http://localhost/riaproxy/Proxy.aspx?req=http://localhost:8082/ria-
samples";
```

Modify the path to the Proxy web service (Proxy.aspx) so the hostname represents the domain context you are writing your web application for. This will usually be localhost or the network accessible name of your machine.

Finally, modify the URL that is passed to the Proxy web service (after the 'req' parameter), so it points to cross domain root location of the Stratus RIA controls.

Step Two: Declaring the Stratus RIA control dependencies

One of the first declarations you must make in the `<head>` section of the HTML is to declare Dojo. Dojo can be declared locally or via proxy. For example:

```
<script type="text/javascript"
src="http://localhost/riaproxy/Proxy.aspx?req=http://hostname:8082/ria-
samples/resource/dojo/dojo.js.uncompressed.js"></script>
```

Following this, we can load the contextpath.js file, which configures the proxy for the Stratus RIA controls. Please ensure that this is declared only after you have declared Dojo, otherwise the page will fail to load correctly.

```
<script type="text/javascript"
src="http://localhost/riaproxy/contextpath.js"></script>
```

Finally, if you intend on using the MapControl on your page, you must load its underlying framework using the following declaration:

```
<script type="text/javascript"
src="http://localhost/riaproxy/Proxy.aspx?req=http://hostname:8082/ria-
samples/openlayers/OpenLayers.js"></script>
```

You can alternatively declare OpenLayers locally if you have your own copy available.

Step Three: Declaring Explicit Dojo CSS Stylesheets

It's important at this point to also explicitly load the Dojo CSS stylesheets, as these are not automatically available via the proxy when Dojo is declared.

```
<style type="text/css">
                @import
"http://localhost/riaproxy/Proxy.aspx?req=http://hostname:8082/ria-
samples/resource/dijit/themes/dijit.css";
                @import
"http://localhost/riaproxy/Proxy.aspx?req=http://hostname:8082/ria-
samples/resource/dijit/themes/tundra/tundra.css";
                @import
"http://localhost/riaproxy/Proxy.aspx?req=http://hostname:8082/ria-
samples/resource/dojo/resources/dojo.css";

</style>
```

Step Four: Register Stratus RIA controls as a Dojo Module

It is at this point we can initialise Dojo and register the Stratus RIA controls as a Dojo module before using them.

Before registering the controls, ensure you have passed the appropriate Dojo initialisation options, a basic example of this is the following:

```
<script type="text/javascript">
var djConfig = { parseOnLoad: true, };
</script>
```

Once this has been declared, we can finally register the Stratus RIA control module – these can either be referenced via the proxy, or locally if you have them available in your own domain:

```
dojo.registerModulePath("stratus",
"http://localhost/riaproxy/Proxy.aspx?req=http://hostname:8082/ria-
samples/widgets/stratus")
```

It is at this point where errors can occur if your contextpath.js file or proxy is setup incorrectly. Ensure these are all working before proceeding.

After you have registered the Stratus module, you can proceed as normal in instantiating Stratus RIA controls in their usual manner. Once you have done so, there is a final step to ensure that they are displayed correctly.

Step Five: Assign Explicit CSS Stylesheets to Controls

Due to the proxy process, you must explicitly assign a corresponding CSS stylesheet to each (sub)control you use. This ensures that they are displayed correctly and function without error or unexpected results.

For example, if you are declaring a Map Control with PanControl, Zoom Slider and BaseMapPicker controls, you will have to declare the following after you have programmatically instantiated the MapControl itself:

```
        appMgr.loadStyleSheet("http://hostname:8082/ria-
samples/widgets/stratus/css/panControl.css");
        appMgr.loadStyleSheet("http://hostname:8082/ria-
samples/widgets/stratus/css/zoomSlider.css");
        appMgr.loadStyleSheet("http://hostname:8082/ria-
samples/widgets/stratus/css/legendControl.css");
```

These CSS stylesheets can be accessed via proxy, or referenced locally if you have the Stratus RIA controls available on the same domain.

Alternatively you can also use this opportunity to provide your own CSS stylesheets for the controls in order to effectively customise them.