



Flash Proxy Configuration Guide

Version 1.0.0

October 8, 2007

Introduction

The Flash Proxy is used with Flash under Javascript (FuJAX). The Flash Proxy is not a proxy at all, but is instead a small Flash player movie that will load into the browser to handle cross-domain browser-to-server communication. By using the Flash Proxy, the following benefits are achieved:

- Direct communications between the browser and MapQuest hosted servers for geocoding, routing, and search operations. You will not need to set up a server-based proxy on your own servers.
- Asynchronous communication, which means that you will need to provide a callback function on each operation.

The Flash Proxy is a small Flash movie file that is loaded with the Javascript. The JS API calls a function inside the Flash swf file. The Flash swf file can communicate directly to the server. It provides an asynchronous call to the server which allows the code to keep going to the next action — that is, it does not have to wait for a response from the server to continue. It is, however, necessary to provide a callback function when using the `Exec` object, in order to handle the response from the MapQuest server.

Configuring the Flash Proxy

To configure your Flash proxy, you need to follow some basic steps, including the following:

1. Download the correct package from the MapQuest Technical Resource Center (TRC).
2. Edit and compile the ActionScript file or create the XML configuration file.
3. Put together all of the files on the web server.
4. Configure your web page to include the Javascript files.

Downloading the Package from the Technical Resource Center (TRC)

Before you begin, you need to download the Javascript API package from the TRC. Inside the Javascript API package, look for two locations:

- The asynchronous Javascript that works with the Flash file is located within the `<FUJAX>` sub-directory.
- The uncompiled ActionScript file to make the Flash movie is located in the `<proxy>/<Flash>` directory tree

Editing and Compiling the ActionScript File

Note

You can either edit and compile this raw ActionScript file (**AAPIProxy.as**) or you can follow the next procedure to create the XML configuration file. In the download package, there is a default **AAPIProxy.swf** file that will be replaced if you modify and compile the **AAPIProxy.as** file and then save it as **AAPIProxy.swf** file when you compile that raw ActionScript file.

If you choose to create the XML configuration file, the JS API will use that configuration file and the default **AAPIProxy.swf** file while the raw ActionScript file will not be used.

1. Open the **AAPIProxy.as** file using any text editor of your choice.

You will need to modify the following two lines of code:

```
_clientConfigId = "YOUR_ID";  
_clientConfigPassword = "YOUR_PASSWORD";
```

2. In this code, you must change:
 - **YOUR_ID** to your client ID
 - **YOUR_PASSWORD** to your password (this information can be found either in your original Welcome Letter e-mail or on the “My Account” page on the TRC web site)
3. Compile this file using the Flex compiler included in the free Flex 2 SDK download found here: <http://www.adobe.com/products/flex/downloads/>.
4. Save the **AAPIProxy.as** file that you modified and then compile it using the downloaded compiler.

Note

Be sure to save the **AAPIProxy.as** file as **AAPIProxy.swf** when compiling.

Creating the XML Configuration File

You can choose to create the XML Configuration File rather than modifying and compiling the **AAPIProxy.as** file.

Note

A pre-compiled version of the **AAPIProxy.swf** file already exists in the package if you choose to use this XML Configuration File, which removes the necessity of you having to modify and compile the **AAPIProxy.as** file.

1. Using any text editing tool, create a file called **ClientConfig.xml**.
2. In this file, create the following lines of code:

```
<ClientConfig>
  <ClientConfigId>YOUR_ID</ClientConfigId>
  <ClientConfigPassword>YOUR_PASSWORD</ClientConfigPassword>
</ClientConfig>
```

3. In this code, you must change:
 - **YOUR_ID** to your customer ID
 - **YOUR_PASSWORD** to your customer ID password
4. Save this file with the name **ClientConfig.xml**.

Putting Together All of the Files on the Web Server

The last step is making sure that you have all of your files in the right location.

1. On your web server, locate the directory in which your web files are located.
2. Put all of the files from the **fujax** package, including the compiled ActionScript file or the XML configuration file that you created into the same directory in which your web files are located.

Configuring Your Web Page to Include the Javascript files

In your web page, you need to modify your code to have the script include for the following JS API files (this should match the list of Javascript files you have extracted from the **<fujax>** folder and placed on your web server):

Note

If you are using the JS API in conjunction with the Tilemap Toolkit, you do not need to include **mqcommon.js**.

- `<script type='text/javascript' src='swfobject.js'></script>`
- `<script type='text/javascript' src='AAPIProxy.js'></script>`
- `<script type='text/javascript' src='mqcommon.js'></script>`
- `<script type='text/javascript' src='mqutils.js'></script>`
- `<script type='text/javascript' src='mqobjects.js'></script>`
- `<script type='text/javascript' src='mqexec.js'></script>`

Note

Modify the SRC location to point to where you placed the files on your web server.
