

[Documentation Home](#)

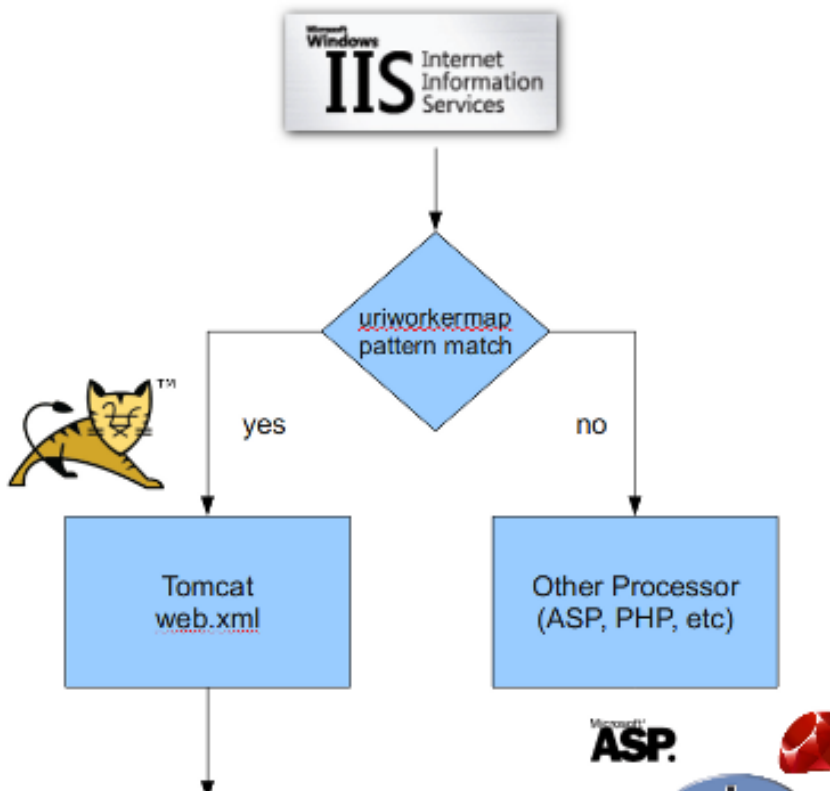
Contents

- [Configuring SES URL's on Windows OS's](#)
- [Configuring IIS](#)
- [Configuring Tomcat](#)

Configuring SES URL's on Windows OS's

The following describes how to configure Search Engine Safe (SES) URL's working on Windows systems and the installer. I would say 95% of the problems that are encountered with setting up SES URL's have to do with a service not knowing where to hand off the request.

When a request comes in on a Windows System, it hits IIS, which looks at the `uriworkermap.properties` file and (theoretically) gets passed to Tomcat. From there, Tomcat checks it's `web.xml` file, and (theoretically) the request gets passed to the Railo servlet, from there, the CFML is read from the file system and the request is processed accordingly. The following flow chart may exemplify it better:





Configuring IIS

IIS configuration is governed by the `uriworkermap.properties` file. The IIS connector knows what to hand off to Tomcat based on the URL patterns that are laid out in this file.

So, with that in mind: **If you're setting up SES URL's and you're getting a 404 from IIS, chances are very good that you need to check your `uriworkermap.properties` file and make sure the URL you're trying to hit is being passed off to Tomcat.**

Updating the `uriworkermap.properties` file is talked about in detail [here](#).

The general idea, though, is to identify a pattern that will always be used in the URL that needs to get passed off to Tomcat (Identify the "key"). So, if your SES URL looks something like this:

```
http://www.mysite.com/mysite/content/newcontent/
```

Then the "Key" would be the `"/mysite/"` in the URL. That means that you could edit your `uriworkermap.properties` file and add the following line:

```
/mysite/*=ajp13
```

If your site doesn't have something in the URL to key off of, then you're going to have to pass EVERYTHING off to Tomcat and Railo. This is not very efficient, as threads by the processing engine will be wasted on mundane tasks, but it will get the job done. Something like this will work in your `uriworkermap.properties` file:

```
/*=ajp13
```

However, at this point IIS will be pretty useless. The only thing it will be doing is handing off requests to Tomcat. You may want to consider removing IIS and just [Utilize Tomcat's Built-In Web Server](#).

Configuring Tomcat

Now that Tomcat is getting the request passed to it from IIS, we now have to tell Tomcat how to handle it. In general, this just means that we want to tell Tomcat to pass the request off to Railo.

There are actually several different ways to configure Tomcat, but we recommend what's generally referred to as the "Site Specific web.xml" method, as it gives you greater control and flexibility over how Tomcat handles requests that are sent to it. To do this, you will need to create the web.xml file within the WEB-INF file that railo creates for a new site. The process goes something like this:

1. The first step is to modify the server.xml file with the information for the new site [explained here](#).
2. Restart Tomcat so that the WEB-INF directory is created in the "DocBase" path, or create the directory yourself
3. Once the WEB-INF directory is created, create the web.xml file inside it. IE:
C:\websites\mysite.com\WEB-INF\web.xml

Once you've created the file, add something similar to the following to create the site-specific mappings:

For Mango Blog

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
CFMLServlet
/blog/post.cfm/*
/blog/page.cfm/*
/blog/archives.cfm/*
/blog/feeds/rss.cfm/*
```

For Mura

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
CFMLServlet
/index.cfm/*
/default/index.cfm/*
```

You can add and remove additional attributes as needed by your specific application.

Once you've added the web.xml files to the site, you may need to restart Tomcat for the updated configs to take full effect.

[Documentation Home](#)