How to Install the Profiles RNS ORNG Extension

You really should read this document in its entirety before making any changes to your system. That might not be the fun way to do things, but it will save you time and frustration. The tricky part setting up IIS to be the front end for the Java server that is needed by ORNG, and fortunately you only have to do this once. The exact details of how to do this vary depending on what version of IIS and Windows you are using, but we have installed ORNG numerous times on a range of server installations so if you need help, reach out to either the Profiles user group or UCSF.

What you will need before starting

You need admin rights to the machines running Profiles RNS that you want to apply this to, and you need some amount of time and patience, but that's pretty much it. Presumably you know your way around .NET and SQLServer. If you know Java and Javascript that will help, but it is not required.

What you will NOT need

You will not need any new additional hardware.

You will not need any new purchased software licenses.

You will not need an additional IP address or DNS entries for your web server

The Process

1. Install the latest Java version on your Profiles RNS web server

The URL for finding the latest and greatest Java version is constantly changing, but if you Google for "java download" and go to an oracle.com website, you are good.

After installing, run "java -version" and confirm that you see something similar to:

C:\Users\emeeks>java -version

java version "1.7.0_10"

Java(TM) SE Runtime Environment (build 1.7.0_10-b18)

Java HotSpot(TM) 64-Bit Server VM (build23.6-b04, mixed mode)

2. Install Apache Tomcat on your Profiles RNS web server

Google for "apache tomcat download" and trust the sites from tomcat.apache.org. We currently run on Tomcat 7.x and strongly recommend that you do the same. The code requires that you at least have Tomcat 7.x, and we have not yet tested on the newer Tomcat 8.x versions.

Install Tomcat on the default port of 8080.

Confirm the install by navigating to http://localhost:8080/ on the server where you performed the installation.

You should see a tomcat admin screen.

3. Install shindigorng.war and configuration files.

Create a C:\shindig\conf directory*. Put shindigorng.properties into the directory and edit as needed for your environment. The four values that require editing have [PLACEHOLDER] values in their setting, they are at the bottom of the file and they are:

orng.dbURL

orng.dbUser

orng.dbPassword

orng.systemDomain

The other items can be left alone and you should only alter them if you take time to look through the code to understand how they are used. The orng.fuseki* items are for an optional advanced feature. It speeds up RDF access and for most gadgets, this is not necessary. It is critical for the ProfileListTool.xml gadget however, as it consumes a large amount of RDF. For now, leave those items as is. If you do wish to install the fuseki component at some later time, you should contact UCSF or the Profiles user group. Further documentation can be found in the shindigorng.properties file.

Put shindigorng.war into your webapps directory under Tomcat: C:\Program Files\Apache Software Foundation\Tomcat 7.0\webapps.

Please view the screenshot-apache.JPG file for an example of the following,

Add C:\shindig\conf to the BEGINNING of your Java Classpath in Tomcat*.

Add the following item to Java Options for Tomcat

-Dshindig.port=80

* Note that you can use a directory other than C:\shindig\conf, but whatever you use you must make sure it is in the beginning of the Java Classpath for Tomcat.

4. Create a secure key for Shindig.

The default location for this key is /shindig/openssl/securitytokenkey.txt, if you want to change this then edit the value for orng.securityTokenKeyFile in shindigorng.properties

The easiest way to create the key is to find a linux machine and issue the following command:

dd if=/dev/random bs=32 count=1 | openssl base64 > securitytokenkey.txt

You can then copy this over to your Profiles RNS windows machine, and place it in a /shindig/openssl directory that you will need to create.

5. Download and Install the Tomcat Connector

This allows Tomcat to work with IIS on the same machine. This is the tricky part, and the details will vary depending on what version of IIS and Windows you use.

http://tomcat.apache.org/connectors-doc/

From the URL above, you will want to click the "binaries" link in the "Download the binaries for selected platforms" for the most recently released version. Click through to grab "windows" binaries and be sure to grab a zip with either "i386-iis" (32 bit) or "64-iis" (64 bit) in the name.

The *.zip will contain an "isapi_redirect.dll" that you need to install.

Follow the directions under "Webserver HowTo" for IIS from http://tomcat.apache.org/connectors-doc/. We have included example worker.properties and uriworkermap.properties files to help you out. You can probably use the ones we have without modification. One thing to note!!!!!! Depending on your Windows version you might need to change the permissions where the isapi_redirect.dll and *.properties files are located so that IIS has rights to view them! We found this out the hard way with Windows 2008.

After you have finished the installation, restart IIS and Tomcat.

6. Test the Installation

Make sure the following URL works (Tomcat is serving this page directly):

http://[yoursite]:8080/shindigorng/gadgets/ifr?url=http://www.labpixies.com/campaigns/todo/todo.xml&view=canvas

If you do not see something that looks like a ToDo list, something went wrong earlier in the process

- Now try to have IIS pass the page request on to Tomcat by removing the :8080:

http://[yoursite]/shindigorng/gadgets/ifr?url=http://www.labpixies.com/campaigns/todo/todo.xml&view=canvas

If you see a 404 error, check step 3.

If you see a blank page, do a View Source. You will probably see a hyperlink into your server at the top of the source. Click it. If you get a Bad Request (Invalid URL) error then you need to fix IIS to work with the long URL's that Shindig generates.

7. Enable the Example Gadgets in the database

```sql
UPDATE [ORNG.].Apps set Enabled = 1 where AppID in (101, 103, 112, 114)
```

```sql
UPDATE [Ontology.].[ClassProperty] set EditExistingSecurityGroup = -20,
    CustomDisplay = 1, EditSecurityGroup = -20, EditPermissionsSecurityGroup = -20,
    EditAddNewSecurityGroup = -20, EditAddExistingSecurityGroup = -20,
    EditDeleteSecurityGroup = -20, ViewSecurityGroup = -1
    where Property in ('http://orng.info/ontology/orng#hasLinks',
    'http://orng.info/ontology/orng#hasSlideShare',
    'http://orng.info/ontology/orng#hasTwitter',
    'http://orng.info/ontology/orng#hasYouTube')
```

8. Edit the ORNG items in Web Config

By default the ORNG keys in the appSettings section of the web.config file are commented out. Uncomment these values to enable OpenSocial support in Profiles RNS.

It is probably best to leave the ORNG.Socket* ones alone. The ShindigURL one is clear, it should be the URL that IIS forwards to shindigorng. For the ORNG.TokenService you want to use the local machine name for your server, this might be something different than what shows up in the URL that you use. The way you can test is by first starting up Tomcat and then attempting to telnet to that name and port:

C:>telnet [LOCAL_MACHINE_NAME] 8777

If you get a connect failed message, then something is wrong. Anything else and you are OK. You will likely get stuck in telnet and need to close the terminal prompt. Oddly enough, this means success. If you want to change the port number from 8777 to something else you need to do so in both shindigorng.properties as well. Note that you want to use a high number port so that ONLY the machine itself can connect! Pick a port that is blocked by any reasonable firewall, as you do not want any outside traffic being able to communicate on this port.

The ORNG.SandboxPassword allows gadgets that are in development to be tested in your Profiles RNS environment. For production, you want to leave this commented out. For your development servers you should set it to something private but not to private, as you will want to share this to anyone who wants to work or test gadget changes. More details can be found in the ORNG_GadgetDevelopment documentation.

9. Join Us and Contribute

Go to http://orng.info to find more gadgets, and feel free to build some to share as well!

Hosting Gadgets

You will notice that the default gadgets are hosted on external web sites. Feel free to use these where they are, or to copy them to one of your own web servers where you can modify them if desired. At UCSF we host them on the same IIS web servers that we use for Profiles RNS.

Running in HTTPS

You can set the gadgets to run on either HTTPS or HTTP, but not both. For most installations, we expect Profiles RNS to run primarily in HTTP, and gadgets will only show up correctly if a page is rendered as HTTP. You can, however, have gadgets work in HTTPS if you want to run Profiles RNS in HTTPS only mode. To do this, set the following:

1. In Web.Config set ORNG.ShindigURL to an https URL

2. In shindigorng.properties, set orng.systemDomain to https for Profiles RNS

3. In Tomcat, configure –Dshindig.port=443 instead of 80

That's it! Note that if you have gadgets that include flash objects, you will need to make sure they use https based flash object for the browser security to let them show.

Upgrading From Profiles 2.5.1

Profiles 2.6.0 requires an upgraded version of Shindigorng.war. There is no need to make any changes to the Tomcat Connectors. If Apache Tomcat 7 is used there is no need to make any changes to Tomcat. If an older version of Apache Tomcat is in use, it must be upgraded to Version 7.

To upgrade Shindigorng.war:

1. Stop Apache Tomcat
2. Navigate to the Tomcat webapps folder
3. Delete the shindigorng folder and the existing sindigorng.war file
4. Copy the sindigorng.war file from the version 2.6.0 zip to the webapps folder.
5. Overwrite the existing shindigorng.properties with the file from the version 2.6.0 zip
6. Update the following values (orng.dbURL, orng.dbUser, orng.dbPassword and orng.systemDomain) in the shindigorng.properties file as described in in section 3 of this document.
7. Restart Apache Tomcat

The ORNG. database schema will be upgraded during the database upgrade scripts. This will also update the application URLs and data for the example gadgets supplied with version 2.5.1.