The Experimental Server Service

Michael Gohde

June 9, 2015

1 Overview

The Experimental Server Service is the core component of RSSE. It parses XML files containing experiments and provides URLs and other such information to any clients that are able to connect to it. Due to the comparatively complex nature of its design, the ESS will be documented in as much detail as possible here.

Please note that all network connections and file formats (aside from those used for configuration) should be regarded as standard. As such, all other implementations of RSSE should follow the same such file formats and networking protocols to ensure maximum compatibility with a wide range of Experiment Clients.

2 The Experiment File

The first notable portion of the Experimental Server Service is entirely in the control of its users and managers. The *Experiment File* is an XML file containing all information relevant to as many experiments as necessary for the given application. This file is currently read on the startup of the Experimental Server Service, however it would be beneficial to continuously re-parse this file in order to continuously provide clients with updates and other such information relevant to changes in experiments.

The Experiment File may be composed of the following tags, though they will be described and several examples will be given in the following subsections.

Tag	Description
rsse	The root tag for the entire file.
experiment	Tag used to denote an individual experiment.
data	Tag used to mark a section for data within an experiment.
url	Denotes an individual member of the dataset accessible by URL.
description	Human-readable description of what the experiment is about.
title	Human-readable title used by the ESS and clients to differentiate between experiments.
report	Flag used to set whether or not experiment clients may post responses.
resserver	The server that each client should post responses to.
resport	The port the client should use while posting responses.

2.1 rsse

The <*rsse*> tag may have at most one attribute named *"version"*. This attribute is a string literal containing the integer representation of the current version of RSSE that the file expects the ESS to adhere to.
Example: <rsse version="1">

2.2 url

The <*url*> tag may also accept attributes describing the URL's class and label if appropriate. These attributes are named, quite appropriately, "class" and "label". The following is an example of such a URL tag with both a class and a label:
Example: <url class="cls" label="1">

If the label in is set to the constant value "-1000", then it will be regarded by the Experimental Server Service as unlabeled.

2.3 Example File

The following is a brief example file made to illustrate the hierarchy of tags implemented for Experiment Files:

```
<rsse version="1">
  <experiment>
      <data>
          <url>testurl1</url>
          <url class="test" label="1">testurl2</url>
      </data>
      <description>This is a test experiment.</description>
      <title>Title</title>
      <report>false</report>
```

```
        <resserver>addr</resserver>
        <resport>port</resport>
  </experiment>
</rsse>
```

3 XML Nuggets

Throughout the comments and related text within the Experimental Server Service itself, there are several references to "XML Nuggets." The concept behind XML Nuggets is that RSSE servers can pass individual small snippets of XML generated from the greater experiment file instead of explicitly using its own network protocol for such purposes.

3.1 Nuggets Sent by the Server

RSSE requires that servers be able to send nuggets containing URLs and other information necessary to define experiments.

The server can send any number of the following tags in each nugget:

Tag	**Description**
nugget	Root level tag for each nugget file.
title	Title of an experiment.
description	Description of the selected experiment.
url	An individual element in a dataset. Technically, this tag should support all of the attributes that it does in the experiment definitions file.
id	ID for the client.

3.2 The Snippet Protocol

Experiment Clients and the ESS communicate using a very simple protocol named *The Snippet Protocol*.

List of commands that may be sent by the client:

Command	Arguments	Description
GETINFO	An experiment name.	Causes the server to return a nugget containing information for a specific client.
LIST	Nothing	Causes the server to provide a nugget with a list of experiment names.
GETALL	A client ID.	Sends all URLs in an experiment.
REGISTER	An experiment name.	The ES returns a user ID.
UNREGISTER	A client ID	Deletes the user ID and all associated data except for responses.
GET	A client ID.	The server returns the next URL in the set or an empty nugget if there is none.

When "argument" is specified, the client prints each argument in the order stated on its own line following the command.

For example, to register, a client may send:

Client: REGISTER
Client: Title of an experiment
Server: (XML nugget with ID number)

4 Responses Sent by the Client

At the time of this writing, the ESS should be able to handle responses sent by clients, though this functionality has not yet been tested or implemented in the Client.

Because this feature is not yet complete, it will not be documented here and any implementation in the ESS is subject to change.

5 Installation and Usage

Installing the Experimental Server Service should be a fairly straightforward process. Each of the following subsections details one step in the process.

5.1 Creating Directories

The following directories should be created (Windows users should replace "var" with the name of their RSSE installation directories.)

1. /var/rsse/ess
2. /var/rsse/ess/db

3. /var/rsse/ess/tmp
4. /var/rsse/ess/resp

5.2 Writing an Experiment File

At this time there are unfortunately no automated utilities to help you write an experiment file. As such, you will have to manually write such a file according to the documentation provided and then place it in "/var/rsse/ess/experiment.xml"

5.3 Generating a Default Settings File

Like all of the other components of RSSE, the Experimental Server Service makes use of configuration files to avoid having to use an excessive number of command line arguments for each invocation. In order to generate the default configuration file, enter the following in a terminal:

java -jar ExperimentalServerService.jar –defconfig

Once the file is written, tweak it to suit your needs and install it in either "/var/rsse/ess/ec.conf" or the install location depending on your operating system.

5.4 Starting the Server

After all of the previous steps have been completed, please ensure that the Experimental Server Service can write to all of the directories you created as well as listen on the sockets specified in the server's configuration file.

To start the server, enter the following command:

java -jar ExperimentalServerService.jar

Unfortunately, a management interface for the server hasn't yet been implemented, so no changes can be made to the server as it runs.