

Configuration File Parsing in Genesis II

Mark Morgan

1 Abstract

The purpose of this document is to indicate the places in the Genesis II codebase where various configuration files are parsed. This document is a live, growing document and more information will be added as time permits. In the interim, I intend to at least cover where various things are parsed.

Each configuration file will be given it's own numbered section and it is possible that other configuration files will potentially reference previous (or subsequent) sections. Further, each section is identified by the name (or group of files if using wildcards) to which the section pertains. Also, all configuration file paths are assumed to be relative to a Genesis II development directory. Finally, some configuration files refer to configuration files that pertain to the container, some pertain to creating resources of specific services, and some are bundled in jar files for other purposes.

2 QAsBESConstructionProperties/*

A set of configuration files that are construction properties¹ for creating or configuring grid queues. These files are all JAXB parsed by **edu.virginia.vcgr.genii.client.queue.QueueConstructionParameters**.

3 deployments/\${deployment-name}/deployment-conf.xml

The deployment-conf.xml configuration file is a simple XML configuration file that describes how a deployment relates to another deployment. Deployments can be hierarchical referring to other deployments when the system asks for a configuration file (or directory) that doesn't exist in a current deployment. This file is JAXB parsed by **edu.virginia.vcgr.genii.client.configuration.DeploymentConf**.

¹ Construction properties are special XML configuration files that are used when creating a grid resource to affect the behavior of that resource.

4 deployments/\${deployment-name}/configuration/client-config.xml

The client-config.xml file is the main configuration file for all Genesis II clients. It includes configuration information such as:

- Accounts and machines to use for sending email
- Code to use when handling data staging URIs (URI handlers)
- How to map a port type interface to the Axis “locator” that can create proxies for it
- Which set of command line tools to load
- Which Axis processing interceptors to run

The parsing of this file is somewhat complex, but is basically handled by the following Java classes:

- **edu.virginia.vcgr.genii.client.configuration.LocatorRegistrySectionHandler**
- **edu.virginia.vcgr.genii.client.configuration.URIHandlerSectionHandler**
- **org.morgan.util.configuration.PropertiesConfigurationSectionHandler**
- **org.morgan.util.configuration.InstanceConfigurationSectionHandler**
- **org.morgan.util.configuration.ClassConfigurationSectionHandler**
- **edu.virginia.vcgr.genii.client.invoke.ClientPipelineSectionHandler**
- **org.morgan.util.configuration.***

5 deployments/\${deployment-name}/configuration/client-socket.properties

A configuration file that allows a user to “tune” various TCP/IP parameters having to do with sockets (things like tcp-no-delay, reuse-address, traffic-class, etc.). This is a Java properties file that is use in

edu.virginia.vcgr.genii.client.comm.socket.SocketConfigurer.

6 deployments/\${deployment-name}/configuration/cservices/*.xml

XML files located in the cservices directory correspond to special “container services” that are not web services but are rather pieces of code loaded and managed by the Genesis II web container that are accessible inside the container for the purpose of performing various activities and jobs. All of these services are configured by an individual .xml configuration file JAXB parsed by

edu.virginia.vcgr.genii.container.cservices.conf.ContainerServiceConfiguration.

7 deployments/\${deployment-name}/configuration/filesystems.xml

The filesystems.xml configuration file configures a special file system “watcher” built into Genesis II that can take various actions depending on space availability changes in various file systems on the machine. This watcher is JAXB parsed by **edu.virginia.vcgr.genii.client.filesystems.FilesystemsConfiguration**.

8 deployments/\${deployment-name}/configuration/global-bes-config.xml

BES's can be configured to export additional environment variables to their activities. This configuration file tells the container what variables all contained BES containers will export (there are additional ways of exporting on a per BES basis). The file is JAXB parsed by **edu.virginia.vcgr.genii.client.bes.envvarexp.EnvironmentExport**.

9 deployments/\${deployment-name}/configuration/secure-runner.properties

Secure Runnables are bundled jar files that contain a descriptor and a set of implemented interfaces. These bundles get hooked into the startup sequence of a grid container and allow for arbitrary code to be run successfully exactly once. This mechanism can be used for patching or even for bootstrapping. Each secure runnable jar bundle is signed to ensure security of the container. This configuration file configures the containers view of secure runnables (who they should be signed by, etc.). This Java properties file is parsed mostly by **edu.virginia.vcgr.secrun.SecureRunnerManager**.

10 deployments/\${deployment-name}/configuration/security.properties

Security properties is the main configuration file for security related aspects of a Genesis II grid container. This Java properties file is largely parsed by **edu.virginia.vcgr.genii.client.configuration.Security**.

11 deployments/\${deployment-name}/configuration/server-config.xml

Similar to the client-config.xml configuration file, the server-config.xml is where a large portion of the configuration options for a Genesis II container happen. However, this file has had significant sections of its content deprecated (or more correctly, moved) into the code base (largely because patching it was becoming

problematic due to container “overriding” values inside it). Currently, it supports the following configuration items:

- Global Properties
- Container-wide Instance Configuration (things like thread pools and DB connection pools)
- AuthZ Providers

The server-config.xml is parsed by the following Java classes:

- **org.morgan.util.configuration.PropertiesConfigurationSectionHandler**
- **org.morgan.util.configuration.InstanceConfigurationSectionHandler**
- **org.morgan.util.configuration.PropertiesConfigurationSectionHandler**
- **org.morgan.util.configuration.***

12 deployments/{deployment-name}/configuration/ui.properties

The ui.properties file is a Java properties file that contains exactly one property, edu.virginia.vcgr.genii.ui.error.report-target. This property specifies the URL of a php script that receives error messages that pop up during the use of the UI and which the user of said UI chooses to send an “error report” to the Genesis II development team. Currently, this file is parsed by **edu.virginia.vcgr.genii.ui.UIConfiguration**.

13 deployments/{deployment-name}/configuration/uri-manager.properties

The uri-manager.properties file is a Java properties file that has exactly one Java property, edu.virginia.vcgr.genii.client.io.uri-manager.max-simultaneous-connections, which tells the BES data stage download/upload code (the URIManager) how many simultaneous stages can be happening at any given time. This file is parsed by **edu.virginia.vcgr.genii.client.io.URIManager**.

14 deployments/{deployment-name}/configuration/web-container.properties

The web-container.properties file describes a couple of properties about the “web” aspect of the Genesis II container. Specifically, what ports to use for web services, which one if any to use for dynamic web page support, whether or not to use SSL, etc. This file is parsed by **edu.virginia.vcgr.genii.container.configuration.ContainerConfiguration**.

15 example-fsproxy/*

The files in the example-fsproxy directory give examples of construction properties that can be used to create a new FSProxy (file system proxy). Largely these files are

superfluous as they are created automatically by the client-ui GUI. However, if you need to work with one, the construction properties are JAXB parsed by **edu.virginia.vcgr.genii.client.exportdir.FSProxyConstructionParameters**.

16 example-nativeq/*

The example-nativeq configuration examples show sample construction property files for configuring various Batch System backed BES containers. Currently we support two types, PBS and SGE and their construction parameters are parsed cooperatively by:

- **edu.virginia.vcgr.genii.client.bes.BESConstructionParameters**
- **edu.virginia.vcgr.genii.client.nativeq.pbs.PBSQueueConfiguration**
- **edu.virginia.vcgr.genii.client.nativeq.sge.SGEQueueConfiguration**

17 src/edu/virginia/vcgr/externalapp/default-config.xml

External applications are used when a user types in an “edit” command on the command line such as “edit grid:foo.txt” or if a user double-clicks on an item in the client-ui browser. This configuration file describes the default mechanism for how to match up various entry types with applications that handle them. It can also be overridden by place a similarly formatted file in the user’s local home directory called .grid-applications.xml as well as a similarly named file in the user’s grid home directory. It is JAXB parsed by **edu.virginia.vcgr.externalapp.ApplicationRegistry**.

18 src/edu/virginia/vcgr/genii/client/resource/known-porttypes.xml

The known-porttypes.xml configuration file lists all of the port types that are known by the Genesis II system. This file is used partially for sanity checking on port type QNames inside the code, but also to assign an order to the port types so that they can be encoded into a bit vector for representation in EPRs. This file is parsed by **edu.virginia.vcgr.genii.client.resource.PortType**.

19 src/edu/virginia/vcgr/genii/client/ser/blob-limits.cfg

The blob-limits configuration file is used internally in the code base to help give us warnings when BLOBs in the database start growing out of control. We used to have a problem where we would allocate a BLOB of some size in a database table, and then overflow it as the system became more complex. To deal with this, we instead made all blobs 2 Gigabytes in size, but started keeping a table of blob limits. If a blob gets checked into the database that exceeds its recorded limit, we allow the operation but log warnings indicating that something has grown unreasonably large. This file is parsed by **edu.virginia.vcgr.genii.client.ser.BlobLimits**.

20 `src/edu/virginia/vcgr/genii/client/sysinfo/property-map.properties`

The `property-map.properties` file is a Java properties file that describes how to map various values returned by the Java System Properties relating to operating system and architecture to JSDL types (basically, if Java says we are an X, how do we translate that to a Y in JSDL). This file is used by **`edu.virginia.vcgr.genii.client.sysinfo.SystemUtils`**.

21 `src/edu/virginia/vcgr/genii/ui/plugins/config.xml`

The `client-ui` uses a notion of “plugins” to configure how the GUI behaves. These plugins are configured using the `plugins config.xml` file. This file is parsed by **`edu.virginia.vcgr.genii.ui.plugins.UIPluginConfigParser`**.

22 `src/edu/virginia/vcgr/secrun/runnables/test/runnable-description.properties`

The `runnable-description.properties` file is the file included in a secure runnable’s bundle (inside `META-INF/secure-runnable`) that describes the secure-runnable’s hooks and configuration options. It is parsed by **`edu.virginia.vcgr.secrun.SecureRunnableDescriptor`**.