GeniiJSDL Library

Mark Morgan (mmm2a@virginia.edu)
Genesis II Project, University of Virginia

Introduction

The GeniiJSDL Library is a pure Java library developed at the University of Virginia by Mark Morgan. This library implements JAXB XML marshalling and unmarshalling of core JSDL [1] types as well as a number of JSDL extension types (such as JSDL POSIX [1], HPC Basic Profile [2] and HPC File Staging Extensions [3], JSDL SPMD Application Extension [4], and the JSDL Parameter Sweep Job Extensions [5]). In addition to providing Java types for XML marshalling and unmarshalling, the GeniiJSDL library also provides a number of JSDL related services to applications that use the library.

The primary function of the GeniiJSDL library is to evaluate or expand[1] JSDL Parameter Sweep documents[2]. This evaluation function takes a single JSDL Parameter Sweep document and, based on the parameter sweep described therein, produces a series of "realized" JSDL documents, each one representing a single parameter sweep *instance* of the original[3]. Applications that use this function can then treat the resultant product JSDL documents as a set of simpler (non-Parameter Sweep) JSDL job definitions.

The ParameterSweep Application

While the primary purpose of the GeniiJSDL library is to provide a Java library for applications that wish to support JSDL Parameter Sweeps, the library is bundled in a

---

[1] I use the terms *evaluate*, *expand*, and *realize* to describe the process of iterating through the parameter sweep parameters and functions described in a JSDL Parameter Sweep document for the purpose of producing a set of non-parameter sweep job definition instances representing the appropriate substitution of parameters with values for each instance.

[2] Even though the GeniiJSDL library generally expands JSDL Parameter Sweep documents, it can also handle the degenerate case of expanding a JSDL document with no parameter sweep (producing the original document in the process).

[3] At the moment, the GeniiJSDL library only supports the DocumentNode Parameter substituent [5]. All Function substituents described in the JSDL Parameter Sweep Job Extension specification [5] are supported (Values, LoopInteger, and LoopDouble).

runnable jar file with a Java command line application that one can use to generate a set of independent JSDL files from a single JSDL Parameter Sweep description. This tool is primarily intended for use as a debugging aid, but could be used to do more rudimentary JSDL Parameter Sweep handling if desired.

The syntax for running this tool is

```
ParameterSweeper <input-jsdl> <output-directory>
```

When run, the tool reads in the parameter sweep described by the *input-jsdl* file and from that description it produces a set of resultant output JSDL files (one each for each parameter sweep instance realized) numbered from 0 to n – 1 (where n is the cardinality of the parameter sweep described) and stored in the specified output directory. The following figure shows an example session using this tool.

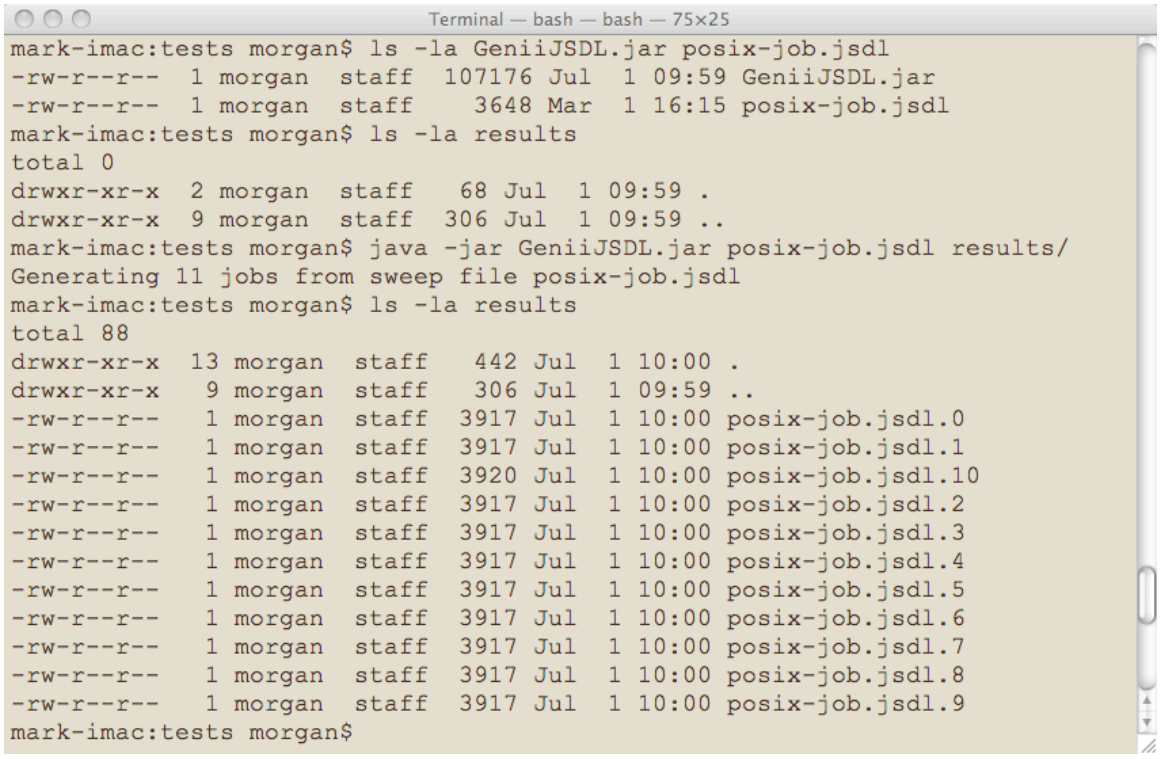

Figure 1: Example Usage of the GeniiJSDL ParameterSweep Application

Realizing Parameter Sweeps with the GeniiJSDL Library

The more common use for the GeniiJSDL is as an application library that Java applications call to expand individual parameter sweep instances from a JSDL Parameter Sweep document. To this end, clients wishing to perform parameter sweeps using GeniiJSDL must first unmarshall (or represent) the desired JSDL

Parameter Sweep document as the GeniiJSDL defined **JobDefinition** type[4]. Once in that form, the GeniiJSDL library consumes the job document along with a callback object provided by the caller and in turn produces a single parameter sweep instance document (in the same JAXB representation) for each parameter sweep expansion by repeatedly invoking the callback object.

Three Java classes provided by the GeniiJSDL library play key roles in this process. They are the **SweepListener**, **SweepToken**, and **SweepUtility** classes located in the *edu.virginia.vcgr.jsdl.sweep* Java package.

SweepListener

The **SweepListener** interface represents the callback interface that a client provides to the **SweepUtility** class when asking it to perform a parameter sweep expansion. This interface contains exactly one method, *emitSweepInstance*, which is called once for each "realized" JSDL document instance produced by the parameter sweep process. Note that because the parameter sweep process happens on a separate thread from the calling environment, invocations made to this method will be in a different Java thread from the one that initiated the parameter sweep expansion.

SweepToken

The **SweepToken** interface is returned to clients when a parameter sweep expansion starts. Parameter sweep expansion happens in a separate thread from the calling environment and the **SweepToken** object provides a convenient way for clients to *join* that thread and block until the parameter sweep expansion has finished.

SweepUtiltiy

The **SweepUtility** class acts as the primary access point for clients that wish to expand a parameter sweep document into it's represented instance documents. This class provides a single public static operation that can return the cardinality of an input JSDL Parameter Sweep document, and a second operation (consisting of two overloaded functions) that takes a JSDL Parameter Sweep **JobDefinition**

---

[4] I realize that not all JSDL aware Java applications use the Java JAXB API for XML serialization and deserialization. Doing so in GeniiJSDL was the result of Genesis II specific application decisions. However, it is not substantially difficult to translate between the JAXB representation and other representations provided sufficient XML serialization and deserialization mechanisms for the second representation type.

instance and a **SweepListener** callback object and with those parameters starts a new parameter sweep expansion process.

Example

The following code snippit shows an example Java application that reads a JSDL Parameter Sweep document from a file and expands that parameter sweep into its constituent **JobDefinition** instances.

```
File source = new File("sweep.jsdl");
Unmarshaller unmarshaller =
   JSDLUtility.JSDLContext.createUnmarshaller();
JobDefinition jobDef =
   (JobDefinition)unmarshaller.unmarshal(source);
System.out.format(
   "Generating %d jobs from sweep file %s\n",
   SweepUtility.sweepSize(jobDef), source);
SweepUtility.performSweep(jobDef,
   new SweepListenerImpl(source, targetDir));
```

Figure 2: Code Snippit showing Parameter Sweep Example

Licensing

Bibliography

[1] Anjomshoaa, A., Brisard, F., Drescher, M., Fellows, D., Ly, A., McGough, S., Pulsipher, D., and Savva, A., 28 July 2008, "Job Submission Description Language (JSDL) Specification," http://www.ogf.org/documents/GFD.136.pdf, GFD.136.

[2] Dillaway, B., Humphrey, M., Smith, C. ,Theimer, M., and Wasson, G., 28 August 2007, "HPC Basic Profile, Version 1.0," http://www.ogf.org/documents/GFD.114.pdf, GFD.114.

[3] Wasson, G. and Humphrey, M., 28 June 2008, "HPC File Staging Profile, Version 1.0," http://www.ogf.org/documents/GFD.135.pdf, GFD.135.

[4] Savva, A., 28 August 2007, "SPMD Application Extension, Version 1.0,", http://www.ogf.org/documents/GFD.115.pdf, GFD.115.

[5] Drescher, M., Anjomshoaa, A., Williams, G., and Meredith, D., 12 May 2009, "JSDL Parameter Sweep Job Extension," http://www.ogf.org/documents/GFD.149.pdf, GFD.149.

[6] Open Grid Forum (OGF), http://www.ogf.org.

[7] Genesis II, http://vcgr.cs.virginia.edu/genesisII.