Gensites

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Overview

To simplify this configuration process, versions of Swift starting with 0.92 include a utility called gensites. The gensites command is used to generate a sites.xml file for running a Swift script on a given site. It accomplishes this by using site templates. The templates used by gensites are the same templates used for internal testing, so they are likely up to date and known to work on a given site.

2 Viewing All Available Site Templates

To view a list of all available templates, run the following command:

```
$ gensites -T
```

You should see output similar to this:

```
beagle
beagle-ssh
intrepid
local
midway
pads
persistent-coasters
queenbee
sge-local
ssh-pbs-coasters
stampede
surveyor
uc3
```

You will notice that the templates can be specific to a particular set of machines like Intrepid and Queenbee, or they may be more general and aim to work across a variety of machines, as in the case of ssh-pbs-coasters. Gensites will look in three directories for available templates: your current directory, $SWIFT_HOME/etc/sites and $HOME/.swift/sites.

3 Listing the Template

To view the contents of a template, type:

```
$ gensites -l templatename
```

Running this command will print the contents of sites.xml file corresponding to the template and give you an idea of what settings you will need to specify.

The tokens are required to properly use the templates. These are placeholder values you will need to specify in the following steps.

4 Providing Site Specific Values

The gensites script needs to know how to replace the placeholder values in the template. This is done by configuring the swift.properties file. Gensites will first look for a swift.properties file in the current directory. If it does not exist, it will next look in $HOME/.swift.

To add site specific values to swift.properties, add a line in the follow format:

```
#site templatename setting=value
```

For example, here is what you could add to swift.properties to replace the values of project, queue and work for the surveyor template:

```
#site surveyor project=MyProject
#site surveyor queue=MyQueue
#site surveyor work=/path/to/workdir
```

Now, running the command *gensites surveyor* will produce the following valid configuration file:

```
<config>
  <pool handle="surveyor">
    <filesystem provider="local" />
    <execution provider="coaster" jobmanager="local:cobalt"/>
    <profile namespace="globus"  key="project">MyProject</profile>
    <profile namespace="globus"  key="queue">MyQueue</profile>
    <profile namespace="globus"  key="kernelprofile">zeptoos</profile>
    <profile namespace="globus"  key="alcfbgpnat">true</profile>
    <profile namespace="karajan" key="jobthrottle">21</profile>
    <profile namespace="karajan" key="initialScore">10000</profile>
    <profile namespace="globus"  key="workersPerNode">1</profile>
    <profile namespace="globus"  key="workerLoggingLevel">DEBUG</profile>
    <profile namespace="globus"  key="slots">1</profile>
    <profile namespace="globus"  key="maxTime">900</profile>
    <profile namespace="globus"  key="nodeGranularity">64</profile>
    <profile namespace="globus"  key="maxNodes">64</profile>
    <workdirectory>/path/to/workdir</workdirectory>
  </pool>
</config>
```

5 Generating Application Configurations with Gensites

Gensites can also be used to create a valid application catalog, commonly called tc.data. Here are some examples of how to specify applications within your swift.properties file:

```
#app intrepid echo=/usr/bin/echo
```

This first example shows a site specific application. The #app definition tells gensites this is related to an application rather than a #site definition. In the second part, echo=/usr/bin/echo, the left hand side is the name of the application that will be called from within Swift. The right hand site is the path name which points to the binary.

```
#app intrepid echo=$HOME/bin/echo
```

Environment variables will be interpreted and converted to full path names for Swift.

```
#app intrepid echo=bin/echo
```

Gensites can take relative paths (relative to your current directory) and translate them to full path names for Swift.

Warning
Running gensites with #app definitions will replace any file called tc.data in your current directory. If a file called tc.data exists, it will be renamed to tc.data.old. If you run gensites twice, the original contents of your tc.data will be lost. Either rename your tc file or copy to a different location.

6 Running Swift With the New Configuration

Now that the gensites is configured and producing a valid configuration file, Swift needs to know to use it. The first step is to create a unique config file based on the preferences you specified.

```
$ gensites surveyor > myconfig.xml
```

This will send the output of gensites to myconfig.xml. This example will use a swift.properties location in the default directories (your current directory, ~/.swift/swift.properties). To specify a different location to the swift.properties, use:

```
$ gensites surveyor -p myswift.properties > sites.surveyor.xml
```

Next, provide the configuration filename to Swift:

```
$ swift -sites.file sites.surveyor.xml myscript.swift
```

Alternatively, if you have specified applications, be sure to load that into Swift:

```
$ swift -sites.file sites.surveyor.xml -tc.file tc.data mycript.swift
```

7 Providing Default Values for All Templates

It is also possible to specify a default value for a setting, regardless of template you use. If you want to set your queue to default to "fast" across all templates, you can do this by omitting the template name. Consider the following swift.properties:

```
#site queue=fast
#site surveyor project=MyProject
#site surveyor work=/path/to/workdir
```

By omitting the template name, the default value for queue on surveyor (and any other template you use) will be set to "fast". One thing to keep in mind when setting default values is that order matters. Be sure to set your default values first before setting template specific values.

Just like the #site definitions, when a site name is not specified, an app will be created for every site that is defined in your template.

```
#app echo=/bin/echo
```

8 More Help

The gensites script provides additional options not discussed here, such as using templates and swift.properties in non-standard directories. For more information, run gensites -h. Here is a full list of all options available.

```
$ gensites -help

   usage: gensites template [-p properties.file] [-L template_directory] [-h] [-T] [-l]

   template              Name of template to use
   -p properties.file    Specify a swift.properties to use
   -L template_directory Specify a non-standard template directory
   -T                    List all templates available
   -h                    Help / usage information
   -l                    List the contents of a specific template

   Examples:
```

```
Create a site configuration file for sites.xml using default properties.file in current ↩
    directory
     $ gensites pads > sites.xml

Use a specific properties file for a site
     $ gensites -p sites.properties pads > sites.xml

Specify a non-standard directory where templates are located
     $ gensites -L template.dir pads > sites.xml
```