

Information Gatherer built, installation and configuration

T. Levshina

Information Gatherer (IG) is a part of the Resource Selection Service (ReSS) project. ReSS allows automatically select OSG resources based on the job requirements. The project provides OSG with a general resource selection service. IG subscribes to multiples CEMon (<http://grid.pd.infn.it/cemon/field.php>) to obtain the resource information of Compute Element and advertises this information to a specified condor pools.

Creating RESS rpm

The steps that are needed for creation of rpm are listed below:

1. List of products needed to be on the built node:
 - a. Java 1.4 or higher *
 - b. Ant 1.6.5 *(* both products could be installed from vdt)
- c. CVS client
2. `ress_thirdparty` (ReSS dependencies libraries – upd product)
The `ress_thirdparty` directory contains the following jar files:

```
/cemon
    glite-ce-monitor-api-java.jar
/classad
    v2.2/
        classad.jar
/common
    activation.jar
    axis-ant.jar
    axis.jar
    bcmail-jdk14-122.jar
    bcpj-jdk14-122.jar
    bcprov-jdk14-122.jar
    bctest-jdk14-122.jar
    commons-beanutils.jar
    commons-discovery-0.2.jar
    commons-logging-1.0.4.jar
    jaxrpc.jar
    log4j-1.2.8.jar
    saaj.jar
    wsdl4j-1.5.1.jar
/security
    glite-security-trustmanager.jar
    glite-security-util-java.jar
```

3. Download production ReSS from cvs:
export CVSROOT=cvsuser@cdcvns.fnal.gov:/cvs/cd

cvs checkout -r v1_0_7_prod ReSS

4. Build ress rpm:
 - a. Export JAVA_HOME
 - b. Export ANT_HOME
 - c. Add java and ant to PATH:
export PATH= JAVA_HOME/bin:\$ANT_HOME/bin:\$PATH
 - d. setup ress_thridparty via ups or
export ReSS_THIRDPARTY_DIR=
 - e. cd ReSS
 - f. ant rpm_dist

The rpm file will be located in

rpm_root/RPMS/noarch/ress-1.0-7.noarch.rpm

IG Installation

The following products should be installed on the node prior IG installation:

1. condor
2. tomcat

We assume that these products are installed from VDT cache and apache, tomcat are already properly configured.

IG can be installed from rpm:

rpm -i ress-1.0-7.noarch.rpm

The software will be located in /opt/ress-1.0 directory.

IG Configuration

The following steps need to be done in order to properly configure IG:

1. Customization of configuration file.
IG configuration file is called ig.properties. It is located in /opt/ress-1.0/var/config directory. The subsequent properties should be specified:
 - fnal.ress.condor.pools*** - list of condor's pools, separated by comma
 - fnal.ress.condor.path*** - fully qualified path to condor installation
 - fnal.ress.condor.config*** – fully qualified path to condor_config script (default is \$CONDOR_LOCATION/etc/condor_config)

All the other properties in the configuration file can stay unchanged, there are in use only if dynamic subscription to CEMonitor is used.

The log4j.properties file that is located in the same directory does not need any modification unless you want to change log output level, log file size, name or its location. The default debug level is “INFO” and the log file is placed under \$CATALINA_BASE/logs/ig.log

2. File “staticCondorClassadAttributes.data” contains the additional attributes that are added to all classads:

```
MyType = "Machine"
TargetType = "Job"
CurMatches = 0
Requirements = (CurMatches < 10)
WantAdRevaluate = true
ReSSVersion= "1.0.6"
isClassadValidAreCriticalAttributesPresent = ( GlueSiteName != UNDEFINED &&
GlueHostApplicationSoftwareRunTimeEnvironment != UNDEFINED && GlueHostNetworkAdapterInboundIP !=
UNDEFINED && GlueHostNetworkAdapterOutboundIP != UNDEFINED && GlueSubClusterTmpDir !=
UNDEFINED && GlueSubClusterWNTmpDir != UNDEFINED )
isClassadValidAreImportantAttributesPresent = ( GlueSubClusterPhysicalCPUs != UNDEFINED &&
GlueSubClusterLogicalCPUs != UNDEFINED && GlueCEStateStatus != UNDEFINED && GlueCEInfoContactString
!= UNDEFINED )
isClassadValidAreTotalSlotsAndCPUPositive = ( GlueCEInfoTotalCPUs != UNDEFINED && GlueCEInfoTotalCPUs > 0
&& GlueCEPolicyAssignedJobSlots != UNDEFINED && GlueCEPolicyAssignedJobSlots > 0 )
isClassadValidAreStateSlotsAndCPUNonNegative = ( GlueCEStateFreeCPUs != UNDEFINED &&
GlueCEStateFreeCPUs >= 0 && GlueCEStateFreeJobSlots != UNDEFINED && GlueCEStateFreeJobSlots >= 0 &&
GlueCEStateTotalJobs != UNDEFINED && GlueCEStateTotalJobs >= 0 && GlueCEStateWaitingJobs !=
UNDEFINED && GlueCEStateWaitingJobs >= 0 && GlueCEStateRunningJobs != UNDEFINED &&
GlueCEStateRunningJobs >= 0 )
isClassadValidIsCEHostNetAvailable = ( GlueCEInfoHostName != UNDEFINED && regexp( "\.lan$",
GlueCEInfoHostName ) != 1 && regexp( "\.localhost$", GlueCEInfoHostName ) != 1 && regexp( "\.localdomain$",
GlueCEInfoHostName ) != 1 && regexp( "\.local$", GlueCEInfoHostName ) != 1 && regexp( "\.internal$",
GlueCEInfoHostName ) != 1 )
isClassadValid = ( isClassadValidAreCriticalAttributesPresent && isClassadValidAreImportantAttributesPresent &&
isClassadValidAreTotalSlotsAndCPUPositive && isClassadValidAreStateSlotsAndCPUNonNegative &&
isClassadValidIsCEHostNetAvailable )
```

You can add more attributes but you will need to restart tomcat in order that changes become active.

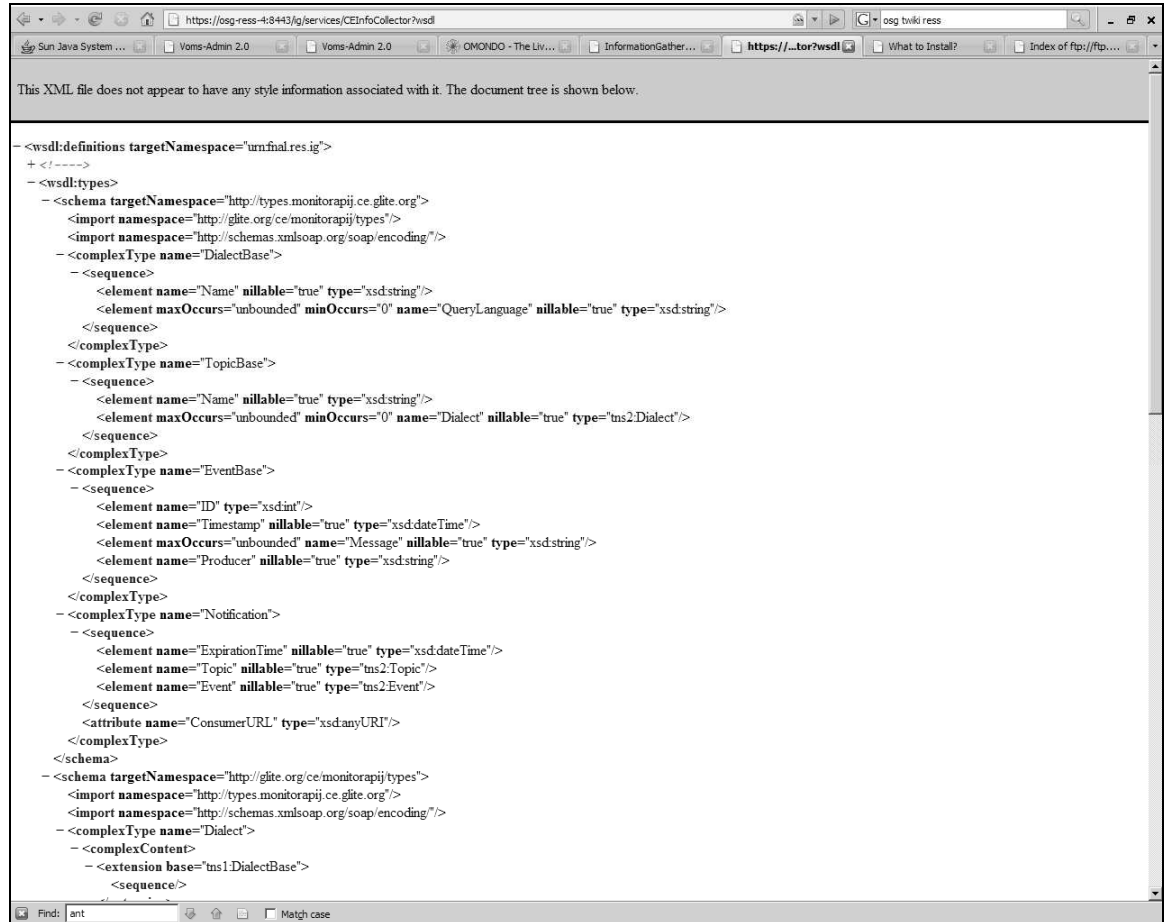
3. Customization of web application configuration file.

The file ig.xml.template located in /opt/ress-1.0/var/config should be modified and then copied under \$CATALINA_HOME/conf/Catalina/localhost directory as ig.xml. The <PRODUCT_DIR> placeholder should be replaced by actual path to IG location (/opt/ress-1.0)

4. Modification of apache configuration for forwarding IG requests to tomcat :
The following lines need to be inserted into \$APACHE_LOCATION/config/httpd.conf

```
JkMount /ig/* tomcat5
JkMount /ig tomcat5
<Location /ig>
SSLVerifyClient require
</Location>
```

5. Restart apache
vdt-control -on apache
6. Restart tomcat
vdt-control -on tomcat-5 or (55)
7. Verify that IG is available (you have to have your certificate uploaded into your browser):
<https://localhost:8443/ig/services/CEInfoCollector?wsdl>
You should see something like that:



```
<?xml version='1.0' encoding='UTF-8'>
<wsdl:definitions targetNamespace="urn:final.res.ig">
  + <!-- -->
  - <wsdl:types>
    - <schema targetNamespace="http://types.monitorapij.ce.glite.org">
      <import namespace="http://glite.org/ce/monitorapij/types"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      - <complexType name="DialectBase">
        - <sequence>
          <element name="Name" nillable="true" type="xsd:string"/>
          <element maxOccurs="unbounded" minOccurs="0" name="QueryLanguage" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
      - <complexType name="TopicBase">
        - <sequence>
          <element name="Name" nillable="true" type="xsd:string"/>
          <element maxOccurs="unbounded" minOccurs="0" name="Dialect" nillable="true" type="tns:Dialect"/>
        </sequence>
      </complexType>
      - <complexType name="EventBase">
        - <sequence>
          <element name="ID" type="xsd:int"/>
          <element name="Timestamp" nillable="true" type="xsd:dateTime"/>
          <element maxOccurs="unbounded" name="Message" nillable="true" type="xsd:string"/>
          <element name="Producer" nillable="true" type="xsd:string"/>
        </sequence>
      </complexType>
      - <complexType name="Notification">
        - <sequence>
          <element name="ExpirationTime" nillable="true" type="xsd:dateTime"/>
          <element name="Topic" nillable="true" type="tns2:Topic"/>
          <element name="Event" nillable="true" type="tns2:Event"/>
        </sequence>
        <attribute name="ConsumerURL" type="xsd:anyURI"/>
      </complexType>
    </schema>
    - <schema targetNamespace="http://glite.org/ce/monitorapij/types">
      <import namespace="http://types.monitorapij.ce.glite.org"/>
      <import namespace="http://schemas.xmlsoap.org/soap/encoding"/>
      - <complexType name="Dialect">
        - <complexContent>
          - <extension base="tns1:DialectBase">
            <sequence>
```