OSG ReSS Information Gatherer Deployment Instructions

Table of Contents

Document Change Log

Version	Date	Change Description	Updated By
V 1.0	05/26/2009	First Version of the Document based on VDT 1.10.1	Parag Mhashilkar
V1.1	06/08/2009	Added ReSS monitoring deployment instructions	Parag Mhashilkar
V1.2	06/17/2009	Corrected the filenames to copy using crontab in HA made	Parag Mhashilkar
V1.3	06/17/2009	Corrected the URLs to verify the Monitoring	Parag Mhashilkar
V1.4	10/29/2009	Documented features supported in v1.0.10	Parag Mhashilkar
V1.5	11/08/2010	Documented features supported in v1.0.11	Parag Mhashilkar

1. Introduction

This document describes steps required to install, configure and operate ReSS Information Gatherer (IG) central service. Initial document was written by Tanya Levshina, and included instructions for building, installation and configuring IG.

2. Downloading ReSS rpm

ReSS rpm can be downloaded from the ReSS homepage –
https://twiki.grid.iu.edu/bin/view/ResourceSelection/WebHome

3. Variables and Explanation

Table below explains the meaning of the variables used in this document.

Variable	Files/Directories to backup
IG_HOME	Place where IG is installed by rpm. Usually, /opt/ress-<ress-major-version>/
VDT_LOCATION	Directory where VDT is installed. If you source the setup file of VDT, this variable is available in the environment.
APACHE_LOCATION	Apache installation directory. If used from VDT, it is – $VDT_LOCATION/apache
CATALINA_HOME	Tomcat installation directory. If used from VDT it is – $VDT_LOCATION/tomcat/<tomcat-version>
CONDOR_LOCATION	Condor Installation directory

Unless specified all references to the path variables are fully qualified path.

4. Installing ReSS

4.1. Pre-Requisites

We assume that following software is already installed and well configured on your machine.

Software	Version	Comments
VDT	Latest version	Use the latest production version for OSG. We will be using Java, Condor and Tomcat from the VDT installation
Java	From VDT above	
Condor	From VDT above	
Tomcat	From VDT above	

4.2. Backup existing configuration

If you are upgrading the ReSS installation, backup the existing configuration files. If this is a new install, skip to next section. Since you may have changed the Apache and Tomcat configurations to tune the IG installation, backup following files/directories before the upgrade -

Software	Files/Directories to backup
IG	$IG_HOME/var/config* $CATALINA_HOME/conf/Catalina/localhost/ig.xml
Apache	$APACHE_LOCATION/conf/httpd.conf

	$APACHE_LOCATION/conf/extra/httpd-ssl.conf
Tomcat	$VDT_LOCATION/post-install/tomcat-55

4.3. Install/Upgrade ress rpm

```
rpm –Uvh ress-<ress-version>.noarch.rpm
```

5. Configure ReSS

5.1. Configure IG

If this is a new install, make changes to the required configuration files as explained below. This is also applicable to changes between major versions, for example, upgrade from ress-1.x-y to ress-2.x-y. Unless there are version specific changes, these general configuration changes hold. For upgrading between minor versions, make sure that the configuration changes are valid after the upgrade process.

5.1.1. Customize $IG_HOME/var/config/ig.properties

Specify following properties –

Properties	Comments
fnal.ress.condor.pools	List of comma-separated condor pools. Use of FQDN is strongly recommended. It is also recommended to disable condor_startd on the condor pool machines to which IG will be advertising.
fnal.ress.condor.path	$CONDOR_LOCATION
fnal.ress.condor.config	Path to condor_config file (default is $CONDOR_LOCATION/etc/condor_config)
fnal.ress.condor.classadfilelifetime	Lifetime in minutes for classad files created in `fnal.ress.condor.dirname`. Defaults to 30 with minimum value allowed as 10. At the beginning of publication cycle, any files older than their lifetime are deleted. Only available in **v1.0.10+**

Explanation of some other properties that could also be useful is given below. All other properties in the configuration file are self explanatory and can stay unchanged. These are only used if dynamic subscription to CEMonitor is used.

Properties	Comments
fnal.ress.ig.allow	White list of accepted VOs
fnal.ress.ig.deny	Black list of accepted VOs
fnal.ress.ig.maxthreads	Maximum number of threads IG will spawn. Useful to tune the performance on a system

A typical ig.properties file looks like -

```
fnal.ress.ig.allow=/opt/ress-1.0/var/config/vo.allow
fnal.ress.ig.deny=/opt/ress-1.0/var/config/vo.deny
#data required for condor advertising

fnal.ress.condor.pools=osg-ress-1
fnal.ress.condor.path=/opt/condor
```

```
fnal.ress.condor.config=/opt/condor/etc/condor_config
fnal.ress.condor.file=/opt/ress-
1.0/var/config/staticCondorClassadAttributes.data
fnal.ress.condor.submitter=/opt/ress-1.0/bin/condor_advertise.sh
fnal.ress.condor.logdir=/usr/local/vdt/tomcat/v55/logs
fnal.ress.condor.dirname=/tmp/condor
fnal.ress.condor.classadfilelifetime=30
fnal.ress.ig.maxthreads=80


#####################################################################
#data required for only in case of dynamic subscription to CEMON
#security
sslCAFiles=/etc/grid-security/certificates/*.0
axis.socketSecureFactory=org.glite.security.trustmanager.axis.AXISSocketFactory
sslKey=/etc/grid-security/httpkey.pem
sslCertfile=/etc/grid-security/httpcert.pem

#cemon specific values
#topic
fnal.ress.ig.topic=OSG_CE
#dialect
fnal.ress.ig.dialect=OLD_CLASSAD
#update period in minutes
fnal.ress.ig.update=5
#subscription expiration date yyyy-mm-dd
fnal.ress.ig.expdate=2010-01-01
```

5.1.1.1. Configuring IG for secure registration

fnal.ress.ig.allow and *fnal.ress.ig.deny* represents the white and black list for the resources you want the IG to allow/deny. Configure these two parameters in ig.properties as described above. Format for these files is as shown below.

```
[parag@ress1x3 ReSS]$ cat /opt/ress-1.0/var/config/vo.deny
# Comments start with hash (#)
# Hostname and comma-separated VO list is separated by tab
# VO list supports patterns
gk04.swt2.uta.edu       .*
fester.utdallas.edu     .*
osg-edu.cs.wisc.edu     .*
```

For OSG installation, IG provides a tool *get-registered-resources.sh* for automatically generating the white list. Use following crontab to do so.

```
*/5 * * * * /opt/ress-1.0/bin/get-registered-resources.sh \
--capath=/usr/local/vdt-2.0.0/globus/TRUSTED_CA \
--url=https://oim.grid.iu.edu/pub/resource/show.php?format=plain-text \
--whitelist-file=/opt/ress-1.0/var/config/vo.allow \
2>&1>>/opt/ress-1.0/var/config/vo.allow.generate.log
```

5.1.2. Customize $IG_HOME/var/config/log4j.properties

The log4j.properties file does not need any modification unless you want to change the log output level, log file size, name or its location. The default log4j.properties file is shown below.

```
log4j.rootLogger=info, R
log4j.rootCategory=info

#-------------------------
log4j.appender.R=org.apache.log4j.RollingFileAppender
log4j.appender.R.MaxFileSize=100000KB
log4j.appender.R.MaxBackupIndex=40
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{MM/dd/yy HH:mm:ss,SSS} :%-
5p:%t:%c.%M: %m%n

#-------------------------
log4j.category.org.globus.gsi=error
log4j.category.org.apache.axis=error
#--------------------------------------
#log4j.appender.R.File=full path to a log file
log4j.appender.R.File=${catalina.base}/logs/ig.log
```

5.1.3. Customize $IG_HOME/var/config/staticCondorClassadAttributes.data

This file contains additional attributes that are added to all the condor classads. Sample attribute list is shown below. You can add more attributes to the list as per your requirement. Change the ReSSVersion to match your current installation version. For the changes to be effective you need to restart tomcat.

```
MyType = "Machine"
TargetType = "Job"
CurMatches = 0
Requirements = (CurMatches < 20)
WantAdRevaluate = true
ReSSVersion= "1.0.8"

isClassadValidAreCrtiticalAttributesPresent = ( GlueSiteName =!=
UNDEFINED && GlueHostApplicationSoftwareRunTimeEnvironment =!=
UNDEFINED && GlueHostNetworkAdapterInboundIP =!= UNDEFINED &&
GlueHostNetworkAdapterOutboundIP =!= UNDEFINED && GlueSubClusterTmpDir
=!= UNDEFINED && GlueSubClusterWNTmpDir =!= UNDEFINED )

isClassadValidAreImportantAttributesPresent = (
GlueSubClusterPhysicalCPUs =!= UNDEFINED && GlueSubClusterLogicalCPUs
=!= UNDEFINED && GlueCEStateStatus =!= UNDEFINED &&
GlueCEInfoContactString =!= UNDEFINED )
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 = ( isClassadValidAreCrtiticalAttributesPresent &&
```

```
isClassadValidAreImportantAttributesPresent &&
isClassadValidAreStateSlotsAndCPUNonNegative &&
isClassadValidIsCEHostNetAvailable )
```

5.1.4. Customize $IG_HOME/var/config/ig.xml.template

Default `ig.xml.template` file is shown below. Change the `<PRODUCT_DIR>` to actual path of `$IG_HOME` and copy the file as `ig.xml` to `$CATALINA_HOME/conf/Catalina/localhost/ig.xml`

```
<!-- This file must be put into $CATALINA_HOME/conf/Catalina/localhost
-->
<!-- It contains the definition for the configuration file path -->

<Context path="/ig" docBase="<PRODUCT_DIR>/webapps/ig.war">
        <Environment name="ConfFileDir"
                     value="<PRODUCT_DIR>/var/config"
                     type="java.lang.String" override="false"/>

</Context>
```

The `$CATALINA_HOME/conf/Catalina/localhost/ig.xml` for above installation looks like following –

```
<!-- This file must be put into $CATALINA_HOME/conf/Catalina/localhost
-->
<!-- It contains the definition for the configuration file path -->

<Context path="/ig" docBase="/opt/ress-1.0/webapps/ig.war">
        <Environment name="ConfFileDir"
                     value="/opt/ress-1.0/var/config"
                     type="java.lang.String" override="false"/>

</Context>
```

5.2. Configure Apache

5.2.1. Customize $APACHE_LOCATION/conf/httpd.conf

Configure apache to forward IG requests to tomcat by adding following lines to the `httpd.conf`. Service name for tomcat "tomcat55" is based on what VDT uses. Change it accordingly for different versions of tomcat.

```
JkMount /ig/* tomcat55
JkMount /ig tomcat55
<Location /ig>
    SSLCACertificatePath /etc/grid-security/certificates
    SSLVerifyClient require
    SSLVerifyDepth  10
    SSLOptions +StdEnvVars +ExportCertData
</Location>
```

5.2.2. Customize $APACHE_LOCATION/conf/extra/httpd-ssl.conf

Comment the following line in `httpd-ssl.conf`

```
# RewriteRule (.*) https://%{SERVER_NAME}:8443$1
```

5.3. Enable apache and tomcat services in VDT

```
vdt-control -enable apache
vdt-control -enable tomcat-55
```

5.4. Increase the memory and file handle limit for tomcat

Edit `$VDT_LOCATION/post-install/tomcat-55` and make the changes (shown in bold) to increase memory and file handle limit of tomcat.

```
# Determine if we're superuser
case `id` in
    "uid=0("* ) vdt_is_superuser=y ;;
    * )         vdt_is_superuser=n ;;
esac
ulimit -n 10000
export CATALINA_HOME=/usr/local/vdt-1.10.1/tomcat/v55
export JAVA_OPTS='-server -Xmx768M -XX:MaxPermSize=256m'
export CATALINA_USER=daemon
```

6. Restart Services

Restart tomcat and apache for the configuration changes to be effective.

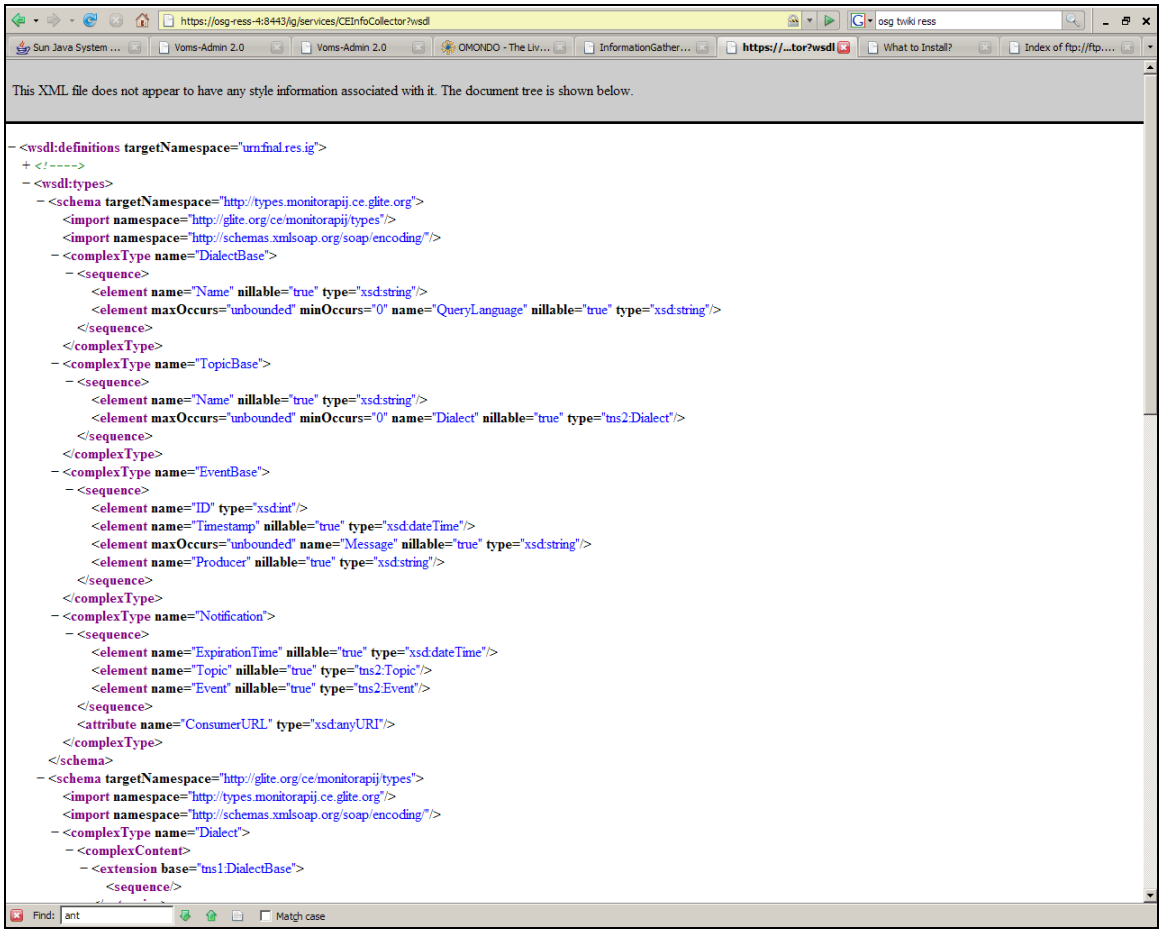
```
# Stop the services
vdt-control -off tomcat-55
vdt-control –off apache

# Start the services
vdt-control -on apache
vdt-control –on tomcat-55
```

7. Verify IG Deployment

Load your x509 certificate in the browser and point your browser to the following url –
`https://<full-hostname>:8443/ig/services/CEInfoCollector?wsdl`

If the services are configured correctly, your browser should display the WSDL for IG



8. *Debugging the Deployment*

Log file location, logging level for IG and duration for which logs will be stored is configured in the properties file `$IG_HOME/var/config/log4j.properties` as explained in previous section. By default it is `$CATALINA_HOME/logs/ig.log`.

For debugging purposes, you can get some useful information from tomcat log `$CATALINA_HOME/logs/catalina.out` as well.

If the services are configured running but could not be contacted, check the firewall and iptables. Port on which IG is running should allow incoming connections.

In case of any issue, `ig.log*` and `catalina.out*` should be preserved for the developers for further troubleshooting. If you find that logs are rotated fast enough increase the log file size in `log4j.properties`.

9. Monitoring Day-to-day Operations

9.1. Number of classads reported

Monitoring the number of classads reported to condor collector(s) is a good way to monitor steady state operations. As of OSG 1.1 there are around 7900+ classads reported by OSG sites. The number of classads could vary based on the downtime of the individual OSG sites. However, this can still be used to identify potential problem with the IG service.

To get the number of classads reporting to the collector (osg-ress-1.fnal.gov in this case) do -

```
condor_status -pool osg-ress-1.fnal.gov -total
```

9.2. Check tomcat is running

```
[root@osg-ress-1 config]# ps -efwww| grep tomcat | grep -v grep
daemon    12578       1   8 May22 ?           08:48:40 /usr/local/vdt-
1.10.1/jdk1.5/bin/java   -server   -Xmx1024M   -XX:MaxPermSize=256m   -
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager       -
Djava.util.logging.config.file=/usr/local/vdt-
1.10.1/tomcat/v55/conf/logging.properties                              -
Djava.endorsed.dirs=/usr/local/vdt-1.10.1/tomcat/v55/common/endorsed   -
classpath                                                 :/usr/local/vdt-
1.10.1/tomcat/v55/bin/bootstrap.jar:/usr/local/vdt-
1.10.1/tomcat/v55/bin/commons-logging-api.jar                          -
Dcatalina.base=/usr/local/vdt-1.10.1/tomcat/v55                        -
Dcatalina.home=/usr/local/vdt-1.10.1/tomcat/v55                        -
Djava.io.tmpdir=/usr/local/vdt-1.10.1/tomcat/v55/temp
org.apache.catalina.startup.Bootstrap start
```

Appendix A – Deploying ReSS Monitoring Tools

ReSS has two monitoring tools -

1. Tool to display history of resources that were advertised in past 30 days.
2. Tool to display the validity of resource classads

This monitoring information is available for production and integration deployments.

Downloading ReSS Monitoring Tools

You can download the ReSS Monitoring Tools by checking out the scripts from CVS

```
export CVSROOT=cvsuser@cdcvs.fnal.gov:/cvs/cd
mkdir -p /usr/local/cron-scripts/log
cd /usr/local/cron-scripts
cvs co ReSS
```

Configure ReSS Monitoring Tools to Collect Information Periodically

In order to collect monitoring information from the ReSS services for monitoring purposes, install following crontabs to be run as root. These crontabs can run as any user provided the user can write to director `/usr/local/cron-scripts/log` and the apache's htdocs directory `$VDT_LOCATION/apache/htdocs`. To override the default VDT_LOCATION of `/usr/local/vdt`, pass VDT_LOCATION environment variable to each of the following crontabs.

```
# Crontab for getting classads from Integration ReSS
22 */6 * * * RESS_MON_CONFIG=/usr/local/cron-
scripts/ReSS/ReSS_mon/etc/config.int.sh /usr/local/cron-
scripts/ReSS/ReSS_mon/bin/run_cron.sh >> /usr/local/cron-
scripts/log/cron_log_int 2>&1

# Crontab to validate classads in Integration ReSS
26 * * * * RESS_CLASSAD_VALIDATION_CONFIG=/usr/local/cron-
scripts/ReSS/ReSS_client_utils/isClassadValid/config.int.sh
/usr/local/cron-
scripts/ReSS/ReSS_client_utils/isClassadValid/run_cron.sh >>
/usr/local/cron-scripts/log/cron_isClassadValid_log_int 2>&1


# Crontab for getting classads from Production ReSS
24 */6 * * * RESS_MON_CONFIG=/usr/local/cron-
scripts/ReSS/ReSS_mon/etc/config.prd.sh /usr/local/cron-
scripts/ReSS/ReSS_mon/bin/run_cron.sh >> /usr/local/cron-
scripts/log/cron_log_prd 2>&1

# Crontab to validate classads in Production ReSS
28 * * * * RESS_CLASSAD_VALIDATION_CONFIG=/usr/local/cron-
scripts/ReSS/ReSS_client_utils/isClassadValid/config.prd.sh
/usr/local/cron-
scripts/ReSS/ReSS_client_utils/isClassadValid/run_cron.sh >>
/usr/local/cron-scripts/log/cron_isClassadValid_log_prd 2>&1
```

Cleaning the Diskspace used by Monitoring Information

By default the monitoring information will only store information for past one month. Any information older than one month will be automatically deleted and diskspace recycled. So the administrators do not have to worry about putting additional scripts to recover disk space used by ReSS monitoring tools.

Monitoring in HA mode

Currently, the monitoring scripts do not have a smart interface to deal with the HA deployment mode. The monitoring information provided by these scripts is not mission critical and is only useful to understand the trend of the resources sending classads to ReSS. To configure ReSS monitoring to run in HA deployment mode, you only need to install the crontabs on one machine and copy the output files to other nodes. The monitoring/output files to be copied are created in `/usr/local/vdt/apache/htdocs/ReSS`. An example for one of the crontabs shown above is as follows –

```
# Crontab for getting classads from Integration ReSS
22 */6 * * * export KRB5CCNAME=/tmp/krb5cc_0_for_cron.$$;
RESS_MON_CONFIG=/usr/local/cron-scripts/ReSS/ReSS_mon/etc/config.int.sh
/usr/local/cron-scripts/ReSS/ReSS_mon/bin/run_cron.sh >>
/usr/local/cron-scripts/log/cron_log_int 2>&1; /usr/kerberos/bin/kinit
-k; scp /usr/local/vdt/apache/htdocs/ReSS/ReSS-int-History.html
root@ress1x2.fnal.gov:/usr/local/vdt/apache/htdocs/ReSS/;
/usr/kerberos/bin/kdestroy; unset KRB5CCNAME

[…]
```

Make sure that directory `ress1x2.fnal.gov:/usr/local/cron-scripts/log` exists. This applies to all the crontabs listed in section 10.2

Verify the Monitoring Tools Deployment

To confirm that the monitoring tools have been installed correctly –

1. Run the crontab manually once after the install.
2. For the four crontabs that were installed, you should get the information about the resources by pointing your browser at following four URLs.

   ```
   https://<hostname>:8443/ReSS/ReSS-int-ClassadValidity.html
   https://<hostname>:8443/ReSS/ReSS-prd-ClassadValidity.html
   https://<hostname>:8443/ReSS/ReSS-int-History.html
   https://<hostname>:8443/ReSS/ReSS-prd-History.html
   ```

Monitoring Information over http

To enable the above monitoring information to be displayed over http (non-secure port)

1. Create required symlinks from nonssl-htdocs directory of apache

```
mkdir $VDT_LOCATION/apache/nonssl-htdocs
cd $VDT_LOCATION/apache/nonssl-htdocs
ln –s ../htdocs/ReSS
```

2. For the four crontabs that were installed, you should get the information about the resources by pointing your browser at following four URLs.

```
http://<hostname>:8080/ReSS/ReSS-int-ClassadValidity.html
http://<hostname>:8080/ReSS/ReSS-prd-ClassadValidity.html
http://<hostname>:8080/ReSS/ReSS-int-History.html
http://<hostname>:8080/ReSS/ReSS-prd-History.html
```