EUROPEAN MIDDLEWARE INITIATIVE

EUROPEAN MIDDLEWARE INITIATIVE

PROXYRENEWAL – ADMINISTRATOR GUIDE

Document version:	**1.0.2-1**
EMI Component Version:	**1.x**
Date:	**March 29, 2013**

This work is co-funded by the European Commission as part of the EMI project under Grant Agreement INFSO-RI-261611.

CONTENTS

1 INTRODUCTION

The proxy renewal daemon runs as an internal component of the WMS node, which is responsible for keeping proxy certificates valid throughout all the lifetime of corresponding jobs. The daemon maintains a repository of proxy certificates that have been registered by the WMS for renewal. After succesfull registration of a proxy, the WMS refers to the repository file whenever it needs access the proxy of a particular job. When the finishes the WMS unregisters the proxy from the repository.

The proxy renewal daemon uses a simple text-based protocol to communicate with the clients (i.e., the WMS components). The communication is done over a local unix socket entirely. The renewal daemon do not expose any network interface. In order for the clients to be able to communicate over the socket and access the proxies from the repositoru, they have to run under the same unix id as the renewal daemon. By default, the glite user account is used as the common service user.

In order to keep the proxy certificates, the renewal daemon periodally contacts a MyProxy server and retrieves a fresh proxy. Therefore, for the renewal mechanism to be working, the users have to store their credentials in a MyProxy server first. When contacting the MyProxy server, the renewal daemon authenticates itself using an X.509 certificate and key (usualy the WMS credentials). The configuration of the MyProxy server has to allow renewal requests done with these credentials.

When VOMS attributes are renewed the renewal daemon uses credentials of the user, thus the process resembles the common way how VOMS attributes are obtained and no special authorization is needed on the VOMS server side.

The renewals are attempted well before a proxy is about to expire. If an attempt fails, other ones are triggered until either one of them succeeds or the proxy expires. If a proxy cannot be renewed, its record is removed from the repository. Informatioo about renewal attempts are logged via syslog, additional detailed information can be enabled using the -d switch.

The repository used by the proxy renewal daemon is a directory containg all registered proxy certificates along with some additional information. Names of the files always start with a hash computed from the X.509 subject name of the proxy. Besides the actual credentials, the daemon stores for each registered proxy also a list of job identifiers identifying jobs that were submitted with the related identity. In order to decrease the management overheads, the renewal daemon aggregates multiple proxy certificates of the same identity into a single proxy in the repository. Using this precaution decreases the number of renewal attempts and overall management operations.

2 GLITE-PROXY-RENEWD COMMAND REFERENCE

2.1 NAME

glite-proxy-renewd - proxy renewal daemon

2.2 SYNOPSIS

glite-proxy-renewd [*options*]

2.3 DESCRIPTION

glite-proxy-renewd registers X.509 proxy certificates and periodicaly renews them using a MyProxy repository.

2.4 OPTIONS

-A *DIR*, --VOMSdir *DIR*
Renew also VOMS attributes if they are embeded in the renewed proxy. If the option is given, the renewal daemon will retrieve a fresh copy of the VOMS attributes and place it inside the new proxy.

-C *DIR*, --CAdir *DIR*
An alternative directory with trusted root anchors. This option overrides the **$X509_USER_DIR** environment variable.

-c *NUM*, --condor-limit *NUM*
Specifies how many *NUM* seconds before expiration of a proxy should the renewal process be started. It defaults to 1800 seconds.

-d*, --debug*
Don't daemonize and start logging to stdout. Increased level of debugging is enabled, too.

-G *FILE*, --voms-config *FILE*
An alternative location of the VOMS configuration.

-h*, --help*
Display a list of valid options.

-k *FILE*, --key *FILE*
Get certificate from *FILE.* This option overrides the **$X509_USER_CERT** environment variable.

-O*, --order-attributes*
Make sure that the order of renewed VOMS attributes is retained. Enabling this option may cause crashes of old VOMS servers (older than 1.8.12).

-r *DIR*, --repository *DIR*
All registered proxies and corresponding metadata will be stored in *repository.* The directory must exist and be writeable by the proxy renewal daemon.

-t *FILE*, --cert *FILE*
Get private key from *FILE.* This option overrides the **$X509_USER_KEY** environment variable.

-V *DIR*, --VOMSdir *DIR*
An alternative directory with trusted VOMS certificates

-v, --version
Display the version of the proxy renewal daemon.

2.5 FILES

/tmp/dgpr_renew_<uid>
A unix socket used to talk to the daemon. It is created the daemon upon its start

***proxy* repository**
A directory containing all the registered proxy certificates and additional meta-data.

There is no configuration file used the proxy renewal daemon.

2.6 ENVIRONMENT

GLITE_PR_TIMEOUT
Sets the maximum number of seconds that the daemon can spend on serving the client over the unix socket. The default value is 120 seconds.

Also, standard globus variables are honoured:

X509_USER_KEY
If **$X509_USER_KEY** is set, it is used to locate the private key file.

X509_USER_CERT
If **$X509_USER_CERT** is set, it is used to locate the certificate file.

X509_CERT_DIR
If **$X509_CERT_DIR** is set, it is used to locate trusted CA's certificates and ca-signing-policy files.

2.7 BUGS

Please report all bugs to the gLite bug tracking system available at *https://savannah.cern.ch*

2.8 SEE ALSO

glite-proxy-renew(1)

2.9 AUTHOR

EU EGEE, EU EMI

Table of Contents

- Files
- Environment
- Bugs
- See Also
- Author

3 GLITE-PROXY-RENEW COMMAND REFERENCE

3.1 NAME

glite-proxy-renew - simple client for the proxy renewal daemon.

3.2 SYNOPSIS

glite-proxy-renew -j *<jobid>* [*options*] *command*

3.3 DESCRIPTION

glite-proxy-renew communicates with the proxy renewal daemon and allows administrators to check it functions correctly. It is not meant to be used regularly or by regular users.

3.4 COMMANDS

start
: Register a proxy with the renewal daemon to keep it renewed. The name of the file from the repository is returned as the result. The **-f** *FILE*, **--file** *FILE* option must be given.

stop
: Unregisters a proxy from the renewal repository. The proxy file will be removed.

get
: Lookup the renewal repository to find whether a proxy for the jobid is registered. If found, the command returns the filename from the repository.

3.5 OPTIONS

-f *FILE*, --file *FILE*
: Specifies the filename containing the proxy certificate that should be renewed.

-h*, --help*
: Display a list of valid options.

-j *JOBID*, --jobid *JOBID*
: Specifies the job id that is uniquely tied with the proxy. This option is obligatory and is required for all commands.

-p *PORT*, --port *PORT*
: Specifies port of the MyProxy server, which should be used to renew the proxy.

-s *SERVER*, --server *SERVER*
: Specifies the name of the MyProxy server, which should be used to renew the proxy.

-v*, --version*
: Display the version of the proxy renewal daemon.

3.6 FILES

/tmp/dgpr_renew_<uid>
: A unix socket used to talk to the daemon. It is created the daemon upon its start

3.7 ENVIRONMENT

GLITE_PR_TIMEOUT
Sets the maximum number of seconds that the daemon is given to answer a request done over the unix socket. The default value is 120 seconds.

3.8 BUGS

Please report all bugs to the gLite bug tracking system available at *https://savannah.cern.ch*

3.9 SEE ALSO

glite-proxy-renewd(8)

3.10 AUTHOR

EU EGEE, EU EMI

Table of Contents

4 MYPROXY CONFIGURATION BY YAIM

4.1 NAME

yaim - YAIM (YAIM Aint an Installation Manager) is, as the name suggests, a way of configuring Grid Services. The aim of YAIM is to provide a simple installation and configuration method that can be used to set up a simple Grid Site but can be easily adapted and extended to meet the need of larger sites. The yaim-myproxy module is configuring the MyProxy server.

4.2 DESCRIPTION

The yaim-myproxy module allows you to configure the MyProxy server.

4.3 CONFIGURATION VARIABLES

This is the list of variables needed to set up in order to configure the MyProxy server.

For more details, please check:

https://twiki.cern.ch/twiki/bin/view/LCG/PX_configuration_variables

Mandatory Variables:

Site admins must ensure these variables are properly defined according to the features of the site

site-info.def variables: These variables are defined in /opt/glite/yaim/examples/site-info.def.

INSTALL_ROOT : Installation root - change if using the re-locatable distribution.

SITE_NAME : The GIIS of the site where the MyProxy server belongs to.

GLOBUS_TCP_PORT_RANGE: Port range for Globus IO. It should be specified as "num1,num2". YAIM automatically handles the syntax of this variable depending on the version of VDT. If it's VDT 1.6 it leaves "num1,num2". If it's a version < VDT 1.6 it changes to "num1 num2".

node specific variables: These variables are defined in /opt/glite/yaim/examples/services/glite-px.

GRID_TRUSTED_BROKERS : List of the DNs of the Resource Brokers host certificates which are trusted by the Proxy node. (ex: /O=Grid/O=CERN/OU=cern.ch/CN=host/testbed013.cern.ch). Now deprecated, use GRID_DEFAULT_RENEWERS instead.

GRID_AUTHORIZED_RENEWERS : List of authorized_renewrs.

GRID_DEFAULT_RENEWERS : List of default_renewers

GRID_AUTHORIZED_RETRIEVERS : List of authorized_retrievers.

GRID_DEFAULT_RETRIEVERS : List of default_retrievers.

GRID_AUTHORIZED_KEY_RETRIEVERS : List of authorized_key_retrievers.

GRID_DEFAULT_KEY_RETRIEVERS : List default_key_retrievers.

GRID_TRUSTED_RETRIEVERS : List of trusted_retrievers.

GRID_DEFAULT_TRUSTED_RETRIEVERS : List of default_trusted_retrievers.

GRID_ALLOW_SELF_AUTHORIZATION : Enable refreshing or renewing proxy by itself. Enabling this is not recommended for security reasons - the compromised proxy could be renewed indefinitely.

MYPROXY_DISABLE_USAGE_STATS : Disable Usage Metrics reporting. The myproxy server may be blocked when target server is inaccessible, thus you may want to disable it.

4.4 EXAMPLES

How to configure the Myproxy node.

```
cat << EOF > /root/site-info.def
SITE_NAME=emitb
PX_HOST=`hostname -f`
GRID_AUTHORIZED_RETRIEVERS="$\backslash$*"
GRID_AUTHORIZED_RENEWERS="
  '/DC=org/DC=terena/DC=tcs/C=CZ/O=Masaryk University/CN=emitb2.ics.muni.cz'
  '/DC=ch/DC=cern/OU=computers/CN=cvitbrcnagios.cern.ch'
  '/DC=ch/DC=cern/OU=computers/CN=lxbra2302.cern.ch'
"
GRID_TRUSTED_RETRIEVERS="
  '/DC=ch/DC=cern/OU=computers/CN=cvitbrcnagios.cern.ch'
"
EOF
/opt/glite/yaim/bin/yaim -c -s /root/site-info.def -n glite-PX
```

To debug the configuration process:

/opt/glite/yaim/bin/yaim -c -s /root/site-info.def -n glite-PX -d 6

4.5 DOCUMENTATION

You can find useful information on these web pages:

Entry point for YAIM documentation:

https://twiki.cern.ch/twiki/bin/view/EGEE/YAIM

The Generic Installation and Configuration guides as well as the YAIM guides:

https://twiki.cern.ch/twiki/bin/view/LCG/LcgDocs

Useful links:

http://lcg.web.cern.ch/LCG/Sites/the-LCG-directory.html

4.6 AUTHORS

YAIM is a collaborative project where different modules are developed and maintened by different groups. Here are some of the present contributors:

Maria Allandes Pradillo, Gergely Debreczeni, Laurence Field, Di Qing, Andreas Unterkircher, Oliver Keeble, Steve Traylen, Owen Synge, Gavin Mccance , Maarten Litmaath, and we are happy to receive patches from everybody !

4.7 CONTACT

To contact YAIM people use the **yaim-contact@cern.ch** email address.

Table of Contents

5 YAIM-LESS MYPROXY DEPLOYMENT PROCEDURE – SCIENTIFIC LINUX

```
#! /bin/sh
# The primary purpose of this script is to document the procedure that may
# be followed in case MyProxy server cannot be configured by YAIM

DN=`openssl x509 -in /etc/grid-security/hostcert.pem -noout -subject |sed 's
    /subject= //'`
echo "$DN"
#cp -p /etc/myproxy-server.config /tmp
cat >> /etc/myproxy-server.config <<EOF

# local configuration for `uname -n`
authorized_renewers "$DN"
authorized_retrievers "*"
EOF

cp -pv /etc/grid-security/host*.pem /etc/grid-security/myproxy
chown -v myproxy:myproxy /etc/grid-security/myproxy/host*.pem

/etc/init.d/myproxy-server restart

chkconfig myproxy-server on

#
# setup BDII resource (optional)
#
# required packages: bdii glite-info-provider-service sudo redhat-lsb
#

INFO_SERVICE_CONFIG='/etc/glite/info/service'
SITE_NAME='sitename'

cp ${INFO_SERVICE_CONFIG}/glite-info-service-myproxy.conf.template ${
    INFO_SERVICE_CONFIG}/glite-info-service-myproxy.conf
cp ${INFO_SERVICE_CONFIG}/glite-info-glue2-myproxy.conf.template ${
    INFO_SERVICE_CONFIG}/glite-info-glue2-myproxy.conf
cat <<EOF >/var/lib/bdii/gip/provider/glite-info-provider-service-myproxy-
    wrapper
/usr/bin/glite-info-service ${INFO_SERVICE_CONFIG}/glite-info-service-
    myproxy.conf $SITE_NAME
/usr/bin/glite-info-glue2-simple ${INFO_SERVICE_CONFIG}/glite-info-glue2-
    myproxy.conf $SITE_NAME
EOF
chmod +x /var/lib/bdii/gip/provider/glite-info-provider-service-myproxy-
    wrapper

# newer slapd with rwm backend required for SL5
```

```
[ -x /usr/sbin/slapd2.4 ] && echo "SLAPD=/usr/sbin/slapd2.4" >> /etc/
    sysconfig/bdii

chkconfig bdii on
/etc/init.d/bdii restart
```

6 YAIM-LESS MYPROXY DEPLOYMENT PROCEDURE – DEBIAN

```
#! /bin/sh
# The primary purpose of this script is to document the procedure that may
# be followed in case MyProxy server cannot be configured by YAIM

DN=`openssl x509 -in /etc/grid-security/hostcert.pem -noout -subject |sed 's
    /subject= //'`
echo "$DN"
#cp -p /etc/myproxy-server.config /tmp
cat >> /etc/myproxy-server.config <<EOF

# local configuration for `uname -n`
authorized_renewers "$DN"
authorized_retrievers "*"
EOF

cp -pv /etc/grid-security/host*.pem /etc/grid-security/myproxy
chown -v myproxy:myproxy /etc/grid-security/myproxy/host*.pem

/etc/init.d/myproxy-server restart

#cp -p /etc/init.d/myproxy-server /tmp/
sed -i /etc/init.d/myproxy-server -e 's/\(# Default-Start:\).*/\1      2 3 4
    5/'
sed -i /etc/init.d/myproxy-server -e 's/\(# Default-Stop:\).*/\1       0 1
    6/'
update-rc.d myproxy-server defaults


#
# setup BDII resource (optional)
#
# required packages: bdii glite-info-provider-service sudo lsb-release
#

INFO_SERVICE_CONFIG='/etc/glite/info/service'
SITE_NAME='sitename'

cp ${INFO_SERVICE_CONFIG}/glite-info-service-myproxy.conf.template ${
    INFO_SERVICE_CONFIG}/glite-info-service-myproxy.conf
cp ${INFO_SERVICE_CONFIG}/glite-info-glue2-myproxy.conf.template ${
    INFO_SERVICE_CONFIG}/glite-info-glue2-myproxy.conf
cat <<EOF >/var/lib/bdii/gip/provider/glite-info-provider-service-myproxy-
    wrapper
/usr/bin/glite-info-service ${INFO_SERVICE_CONFIG}/glite-info-service-
    myproxy.conf $SITE_NAME
/usr/bin/glite-info-glue2-simple ${INFO_SERVICE_CONFIG}/glite-info-glue2-
```

```
    myproxy.conf $SITE_NAME
EOF
chmod +x /var/lib/bdii/gip/provider/glite-info-provider-service-myproxy-
    wrapper

BDII_PASSWD=`dd if=/dev/random bs=1 count=10 2>/dev/null | base64`
cat << EOF > /etc/default/bdii
RUN=yes
SLAPD_CONF=
SLAPD=
BDII_RAM_DISK=
EOF
sed -i "s#.*rootpw.*#rootpw\t${BDII_PASSWD}#" /etc/bdii/bdii-slapd.conf

/etc/init.d/bdii restart
```