

# Setting up a YUM Repository

**SEE-GRID-SCI Training Event,  
Yerevan, Armenia, 24-25 July 2008**

**Hrachya Astsatryan**

Institute for Informatics and Automation Problems  
National Academy of Sciences of the Republic of Armenia  
hrach@sci.am



# Overview

- In some cases it can be very useful to have the use of your own software (OS, middleware, etc.) repository:
  - to speed up the installation of a large number of machines at your site;
  - to arrange a tutorial in a location with a narrow bandwidth.
- To reach this goal you can use a tool by *Dag Wieers* called *mrepo* (FKA YAM); its features:
  - *building of a local APT/YUM repository, downloading all packages required from mirror sites and setting up metadata; support of ISO files;*
  - *configuring HTTP access;*
  - *support of FISH, FTP, HTTP, RSYNC, SFTP, YUM and RHN;*
  - *and more ... (see References)*

# Installation of mrepo

- Binary RPM packages of *mrepo* are available for RedHat, CentOS and Fedora;
- you can also install it on a Scientific Linux 5.X using this package:
  - <http://dag.wieers.com/rpm/packages/mrepo/mrepo-0.8.4-1.el5.rf.noarch.rpm>

- a tool called **createrepo** is required by *mrepo*:

```
# yum install createrepo
```

- ```
# yum localinstall mrepo-0.8.4-1.el5.rf.noarch.rpm
```

# Configuration of mrepo

- The main configuration file is */etc/mrepo.conf* :

```
# cat /etc/mrepo.conf

[main]
srcdir = /var/mrepo
wwwdir = /var/www/mrepo
confdir = /etc/mrepo.conf.d
arch = i386
metadata = yum repomd
```

# Configuration of mrepo

- Configuration files needed to set mirror sources are in */etc/mrepo.conf.d/*
- you can set a separate file for each part of your repository:
  - OS (e.g. Scientific Linux 4.X, 32-bit):

```
# cat /etc/mrepo.conf.d/sl4x_i386.conf
[sl4x]
name      = Scientific Linux 4 (i386)
arch      = i386
os        = http://linuxsoft.cern.ch/scientific/4x/i386/apt/RPMS.os/
updates  = http://linuxsoft.cern.ch/scientific/4x/i386/apt/RPMS.updates/
contrib   = http://linuxsoft.cern.ch/scientific/4x/i386/apt/RPMS.contrib/
```

# Configuration of mrepo

- OS (Scientific Linux 4.X 64-bit):

```
# cat /etc/mrepo.conf.d/sl4x_x86_64.conf
[sl4x]
name = Scientific Linux SL (x86_64)
arch = x86_64
os = http://linuxsoft.cern.ch/scientific/4x/x86_64/apt/RPMS.os/
updates = http://linuxsoft.cern.ch/scientific/4x/x86_64/apt/RPMS.updates/
contrib = http://linuxsoft.cern.ch/scientific/4x/x86_64/apt/RPMS.contrib/
```

- Java packages:

```
# cat /etc/mrepo.conf.d/jpackage.conf
[jpackage17-generic]
name=JPackage 1.7, generic
free = http://linuxsoft.cern.ch/jpackage/1.7/generic/RPMS.free/
non-free = http://linuxsoft.cern.ch/jpackage/1.7/generic/RPMS.non-free/
[jpackage5-generic]
name=JPackage 5, generic
free = http://linuxsoft.cern.ch/jpackage/5.0/generic/RPMS.free/
non-free = http://linuxsoft.cern.ch/jpackage/5.0/generic/RPMS.non-free/
```

# Configuration of mrepo

- *gLite* middleware (a long list, see References):

For example in case of UI, we have to create

- `wget http://grid-deployment.web.cern.ch/grid-deployment/yaim/repos/glite-UI.repo` and rename to `conf`

# Configuration of mrepo

- LCG CAs:

```
# cat /etc/mrepo.conf.d/lcg-ca.conf
[glite_sl4]
name          = gLite middleware (i386)
arch          = i386
security      = http://linuxsoft.cern.ch/LCG-
CAs/current/RPMS.production/
```

- `wget http://grid-deployment.web.cern.ch/grid-deployment/yaim/repos/lcg-CA.repo` and rename to `confg`

# Using mrepo

## ■ *mrepo* options:

```
$ mrepo -h
```

```
usage: mrepo [options] dist1 [dist2-arch ..]
```

```
Set up a distribution server from ISO files
```

```
mrepo options:
```

|                        |                                          |
|------------------------|------------------------------------------|
| -c, --config=file      | specify alternative configfile           |
| -f, --force            | force repository generation              |
| -g, --generate         | generate mrepo repositories              |
| -n, --dry-run          | show what would have been done           |
| -q, --quiet            | minimal output                           |
| -r, --repo=repo1,repo2 | restrict action to specific repositories |
| --remount              | remount distribution ISOs                |
| -u, --update           | fetch OS updates                         |
| -v, --verbose          | increase verbosity                       |
| -vv, -vvv, -vvvv..     | increase verbosity more                  |
| --unmount              | unmount distribution ISOs                |

# Using mrepo

## ■ Usage examples:

- to update repository (the first time it downloads all the packages):

```
# mrepo -uvv
```

- to generate the repository (with all metadata and symlinks required):

```
# mrepo -gvv
```

- to update and generate your repository:

```
# mrepo -guvv
```

- to update only a section or a *repository*:

```
# mrepo -guvv gilda_app
```

```
# mrepo -guvv -r security
```

# Packages and symlinks

- Once the creation of the repository has been completed, you will find all downloaded packages in:

```
$ ls /var/mrepo/  
all                glite_sl3-i386    jpackage17-generic-i386  sl4x-  
x86_64  
gilda_app-i386    glite_sl4-i386    jpackage5-generic-i386  
gilda_sl4-i386    ig_sl4-i386       sl4x-i386
```

- and symlinks here:

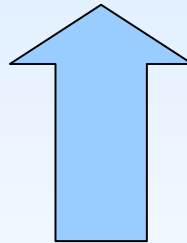
```
$ ls /var/www/mrepo/  
gilda_app-i386    glite_sl4-i386    jpackage17-generic-i386  repos  
gilda_sl4-i386    HEADER.shtml      jpackage5-generic-i386   sl4x-i386  
glite_sl3-i386    ig_sl4-i386       README.shtml             sl4x-  
x86_64
```

# Metadata

- All metadata is in the *wwwdir* :

```
$ ls /var/www/mrepo/sl4x-i386/RPMS.os/repo/
```

```
filelists.xml.gz  other.xml.gz  primary.xml.gz  repomd.xml
```



YUM Metadata

# Required space

- About disk usage :

```
# du -hs /var/mrepo/*
4.0K    /var/mrepo/all
973M    /var/mrepo/gilda_app-i386
656K    /var/mrepo/glite_sl3-i386
13G     /var/mrepo/glite_sl4-i386
102M    /var/mrepo/ig_sl4-i386
803M    /var/mrepo/jpackage17-generic-i386
1.1G    /var/mrepo/jpackage5-generic-i386
2.8G    /var/mrepo/sl4x-i386
3.1G    /var/mrepo/sl4x-x86_64

# du -hs /var/mrepo/
22G     /var/mrepo/
```

# Web server configuration

- Inside mrepo package there is also a configuration file for the Apache web server:

*[/etc/httpd/conf.d/mrepo.conf](#)*

- the default web directory is *[/var/www/mrepo](#)* , so you can access your repository at following URL:

*[http://<your\\_rep\\_FQDN>/mrepo](http://<your_rep_FQDN>/mrepo)*

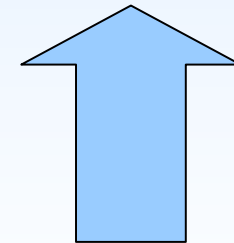
- Remember to start Apache and set it to start at boot:

```
# /etc/init.d/httpd start  
# chkconfig httpd on
```

# Client Configuration (YUM)

- In order to let a client machine install software from your repository, you have to properly set YUM configuration files on that machine:

```
# ls /etc/yum.repos.d/  
gilda.repo          ig.repo             sl.repo  
glite-bdii.repo     jpackage.repo  
ca.repo            glite-lcg_ce.repo  dag.repo  glite-torque_server.repo  
glite-torque_utils.repo
```



example of a *gLite* Torque CE

# Client Configuration (YUM)

- Example of a *.repo* file:

```
# cat /etc/yum.repos.d/glite-torque_server.repo

# gLite TORQUE server repositories
[gliteTORQUEserver_sl4_externals]
name      = gLite WN 3.1 (externals)
baseurl  = http://<your_rep_FQDN>/mrepo/glite_sl4-
i386/RPMS.torque_server-externals/
enabled  = 1
protect  = 0

[gliteTORQUEserver_sl4_release]
name      = gLite WN 3.1 (release)
baseurl  = http://<your_rep_FQDN>/mrepo/glite_sl4-i386/RPMS.torque_server-
release/
enabled  = 1
protect  = 0

[gliteTORQUE_sl4_updates]
name      = gLite WN 3.1 (updates)
baseurl  = http://<your_rep_FQDN>/mrepo/glite_sl4-i386/RPMS.torque_server-
updates/
enabled  = 1
protect  = 0
```

# Client Configuration (YUM)

- Another example of a *.repo* file:

```
# cat /etc/yum.repos.d/ig.repo
```

```
#  
# INFNGRID repositories  
#
```

```
[ig_sl4]
```

```
name      = ig 3.1
```

```
baseurl  = http://<your_rep_FQDN>/mrepo/ig_sl4-i386/RPMS.3_1_0/
```

```
enabled  = 1
```

```
protect  = 0
```

```
[ig_sl4_externals]
```

```
name      = ig 3.1 (externals)
```

```
baseurl  = http://<your_rep_FQDN>/mrepo/ig_sl4-i386/RPMS.3_1_0_externals/
```

```
enabled  = 1
```

```
protect  = 0
```

# How to *freeze* your repository

- If for some reasons you wish to *freeze* your repository, you only need to comment a line in *mrepo* cron file:

```
# cat /etc/cron.d/mrepo  
#30 2 * * * root /usr/bin/mrepo -q -ug
```

- A good idea is to update CAs package anyway:

```
30 2 * * * root /usr/bin/mrepo -q -ug -r security
```



Case of CAs mirror defined by:

```
security = http://linuxsoft.cern.ch/LCG-CAs/current/RPMS.production/
```

# References

- mrepo: Yum/Apt repository mirroring (fka yam)
  - <http://dag.wieers.com/home-made/mrepo/>
- CNAF wiki pages about *mrepo* (you can find here a full list of gLite repository settings)
  - <http://igrelease.forge.cnaf.infn.it/doku.php?id=doc:tips:mrepo>
- Generic Installation and Configuration Guide for gLite 3.1
  - <https://twiki.cern.ch/twiki/bin/view/LCG/GenericInstallGuide310>
- gLite 3.1 middleware repositories:
  - <http://grid-deployment.web.cern.ch/grid-deployment/yaim/repos/>