



EUROPEAN MIDDLEWARE INITIATIVE

STORM: SYSTEM ADMINISTRATION GUIDE INSTALLATION AND CONFIGURATION

Document version: **1.5.0**
EMI Component Version: **1.11.0**
Date: **February 12, 2013**



This work is co-funded by the EC EMI project under the FP7 Collaborative Projects Grant Agreement Nr. INFOSO-RI-261611.

CONTENTS

1	INTRODUCTION	5
1.1	SERVICE ARCHITECTURE	5
2	QUICKSTART GUIDE	6
3	PREREQUISITES AND RECOMMENDATIONS	8
3.1	OPERATING SYSTEM	8
3.2	HARDWARE	8
3.3	SYSTEM USERS	8
3.3.1	LDAP SERVER INSTALLATION AND CONFIGURATION	8
3.3.2	LDAP CLIENT INSTALLATION AND CONFIGURATION	12
3.3.3	TEST CLIENT-SERVER LDAP INSTALLATION	13
4	INSTALLATION PREREQUISITES	14
4.1	OPERATING SYSTEM	14
4.2	NTP SERVICE	14
4.3	FQDN HOSTNAME	14
4.4	HOST CERTIFICATES FOR STORM NODES	15
4.5	ACL SUPPORT	15
4.6	EXTENDED ATTRIBUTE SUPPORT	16
5	INSTALLATION GUIDE	18
5.1	REPOSITORY SETTINGS	18
5.1.1	COMMON REPOSITORIES SETTINGS	18
5.1.2	EMI REPOSITORY SETTING	19
5.2	INSTALL STORM NODES	19
6	CONFIGURATION	20
6.1	GENERAL YAIM VARIABLES	20
6.2	FRONT-END CONFIGURATION	20
6.2.1	YAIM VARIABLES	20
6.3	BACK-END CONFIGURATION	22
6.3.1	YAIM VARIABLES	22
6.4	GRIDHTTPS CONFIGURATION	28
6.4.1	YAIM VARIABLES	28
6.5	CONFIGURING MULTIPLE SERVICES ON A SINGLE HOST	29

7	ADVANCED CONFIGURATION	30
7.1	FRONTEND CONFIGURATION	30
7.1.1	FRONTEND SERVICE INFORMATION: STORM-FRONTEND-SERVER.CONF . .	30
7.1.2	LOGGING FILES AND LOGGING LEVEL	31
7.2	BACKEND CONFIGURATION	33
7.2.1	BACKEND SERVICE INFORMATION: STORM.PROPERTIES	33
7.2.2	BACKEND STORAGE INFORMATION: NAMESPACE.XML	37
7.2.3	BACKEND STORAGE USAGE INITIALIZATION: <i>USED-SPACE.INI</i> [STORM VER- SION >= 1.8.0]	44
7.2.4	LOGGING: LOGGING.XML	46
7.2.5	SPACE AUTHORIZATION: AUTHZ.DB	48
7.3	GRIDFTP CONFIGURATION	49
7.3.1	LOGGING FILES AND LOGGING LEVEL	49
7.4	GRIDHTTPS CONFIGURATION	49
7.4.1	GRIDHTTPS SERVICE INFORMATION: WEB.XML	49
7.4.2	GRIDHTTPS PLUGIN INFORMATION: STORM.GRIDHTTPS.PLUGIN.PROPERTIES	51
7.4.3	LOGGING FILES AND LOGGING LEVEL	51
7.5	STORM EMIR CONFIGURATION	52

1 INTRODUCTION

This document is addressed to Site Administrators responsible for the StoRM installation and configuration.

1.1 SERVICE ARCHITECTURE

Service is characterized by the following components:

mandatory components: FrontEnd (FE), BackEnd (BE), Dynamic Info Provider (DIP);

optional components: GridFTP, GridHTTPs, Client.

To more details look at Functional Description Guide.

2 QUICKSTART GUIDE

This guide gives the steps to be followed when installing StoRM services in a standalone deployment on a single host. The only Virtual Organizations that will be configured are testers.eu-emi.eu and dteam

1. Install one of the supported Operating System: SL5 64 bit, SL6 64 bit

2. Check for NTP service:

```
service ntpd status
```

3. Check for hostname (FQDN):

```
hostname -f
```

4. Check hosts file content

```
cat /etc/hosts
```

5. Check for host certificates:

```
ls -l /etc/grid-security/hostcert.pem /etc/grid-security/hostkey.pem
```

6. Check for Extended ACL support:

```
touch testfile; setfacl -m u:root:rw testfile
```

7. Check for EA support:

```
setfattr -n user.test -v test testfile
```

8. Satisfy common repository settings:

- on SL5

```
wget http://archives.fedoraproject.org/pub/epel/5/x86_64/  
epel-release-5-4.noarch.rpm &&  
yum localinstall --nogpgcheck epel-release-5-4.noarch.rpm
```

- on SL6

```
wget http://www.nic.funet.fi/pub/mirrors/fedora.redhat.com/pub/epel/6/  
x86_64/epel-release-6-8.noarch.rpm &&  
yum localinstall --nogpgcheck epel-release-6-8.noarch.rpm
```

9. Satisfy EMI repository settings:

- on SL5

```
wget http://emisoft.web.cern.ch/emisoft/dist/EMI/3/s15/x86_64/base/  
emi-release-3.0.0-1.s15.noarch.rpm &&  
yum localinstall --nogpgcheck emi-release-3.0.0-1.s15.noarch.rpm
```

- on SL6

```
wget http://emisoft.web.cern.ch/emisoft/dist/EMI/3/sl6/x86_64/base/  
emi-release-3.0.0-1.sl6.noarch.rpm &&  
yum localinstall --nogpgcheck emi-release-3.0.0-1.sl6.noarch.rpm
```

10. Satisfy EGI repository settings:

```
wget http://repository.egi.eu/sw/production/cas/1/current/repo-files/  
EGI-trustanchors.repo -O /etc/yum.repos.d/EGI-trustanchors.repo
```

11. Clean YUM cache:

```
yum clean all
```

12. Install ca-policy-egi-core:

```
yum install ca-policy-egi-core
```

13. Install StoRM node metapackages:

```
yum install emi-storm-backend-mp emi-storm-frontend-mp emi-storm-globus-gridftp-mp  
emi-storm-gridhttps-mp
```

14. Install StoRM pre-assembled configuration:

```
yum install storm-pre-assembled-configuration
```

15. Edit StoRM pre-assembled configuration:

```
vi /etc/storm/siteinfo/storm.def
```

- `STORM_BACKEND_HOST=<your full hostname>`

SL6 and SL5.X X >= 9 only:

- `JAVA_LOCATION="/usr/lib/jvm/java"`

16. Configure StoRM nodes:

```
/opt/glite/yaim/bin/yaim -c -d 6 -s /etc/storm/siteinfo/storm.def -n se_storm_backend  
-n se_storm_frontend -n se_storm_gridftp -n se_storm_gridhttps
```

17. Verify StoRM services:

- `service storm-backend-server status`
- `service storm-frontend-server status`
- `service storm-globus-gridftp status`
- `service storm-gridhttps-server status`

3 PREREQUISITES AND RECOMMENDATIONS

3.1 OPERATING SYSTEM

The following operating systems are supported:

- SL5 64 bit
- SL6 64 bit

It is assumed that the operating system is already properly installed.

3.2 HARDWARE

- Memory: at least 1GB on BackEnd host

3.3 SYSTEM USERS

If you are going to install StoRM on a multi-node scenario where StoRM and StoRM GridHTTPs are going to be deployed on different hosts then the user running BackEnd service and the GridHTTPs service **must be equivalent** on those hosts (the same GID and UID).

To satisfy this requirement you can configure a NIS service for the involved hosts and add the two users to the NIS maps. A tutorial on how to setup a NIS service can be found here: <http://www.tldp.org/HOWTO/NIS-HOWTO/index.html>

A valid solution to share GID and UID between different hosts and provide a user authentication can be found with a client-server LDAP installation. The following is a tutorial that will describe how to install and configure OpenLDAP on a Scientific Linux distributed architecture.

3.3.1 LDAP SERVER INSTALLATION AND CONFIGURATION

- LDAP installation (as root) on SL5:

```
yum install openldap-servers
```

or on SL6:

```
yum install openldap openldap-servers
```

- OpenLDAP installs several files in /etc and other places. The slapd daemon's configuration file is slapd.conf and can be found in /etc/openldap. First of all make sure service is not running. On SL5:

```
service ldap stop
```

or on SL6:

```
service slapd stop
```


- Edit `/etc/openldap/slapd.conf`. We're looking for 5 entries, that of database, suffix, rootdn, rootpw, and directory. It should look something like:

```
database      bdb
suffix        "dc=example,dc=com"
rootdn        "cn=Manager,dc=example,dc=com"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slapd.conf(5) for details.
# Use of strong authentication encouraged.
# rootpw      secret
directory     /var/lib/ldap
```

Define ourselves as a company called `storm.cnaf.infn.it`. For the database we can leave the default `bdb`, which is the Berkeley Database. The suffix will be our main domain or organization. So, we can change it to:

```
suffix        "dc=storm,dc=cnaf,dc=infn.it"
```

Next, the `rootdn`. This means root Distinguished Name. Every entry in an LDAP database has a distinguished name. The default is `Manager`. In our example we will use `root` instead of `Manager`, so according to the suffix we have to change it to:

```
rootdn        "cn=root,dc=storm,dc=cnaf,dc=infn.it"
```

The `dc` is for domainComponent. Next the password. We want an encrypted password, so we can launch `slappasswd` command, insert our password and copy the output string. Just replace `secret` with the encrypted password created, so it looks something like:

```
rootpw        {SSHA}ca6CWAHXogaQ2Cib9sxOYRwHRzyKoSXA
```

We could edit the `directory` line but we will leave the default value `/var/lib/ldap` in our example.

- Clean up previous LDAP content and configuration with:

```
rm -rf /var/lib/ldap/*
```

On SL6 run also:

```
rm -rf /etc/openldap/slapd.d/*
```

- Create the `DB_CONFIG` file into `/var/lib/ldap/` (or copy from OpenLDAP example files `/etc/openldap/DB_CONFIG.example`)

```
# $OpenLDAP: pkg/ldap/servers/slapd/DB_CONFIG,v 1.3.2.4 2007/12/18 11:53:27 ghenry Exp $
# Example DB_CONFIG file for use with slapd(8) BDB/HDB databases.
#
# See the Oracle Berkeley DB documentation
# <http://www.oracle.com/technology/documentation/berkeley-db/db/ref/env/db_config.html>
# for detail description of DB_CONFIG syntax and semantics.
#
# Hints can also be found in the OpenLDAP Software FAQ
# <http://www.openldap.org/faq/index.cgi?file=2>
```

```
# in particular:
# <http://www.openldap.org/faq/index.cgi?file=1075>

# Note: most DB_CONFIG settings will take effect only upon rebuilding
# the DB environment.

# one 0.25 GB cache
set_cachesize 0 268435456 1

# Data Directory
#set_data_dir db

# Transaction Log settings
set_lg_regionmax 262144
set_lg_bsize 2097152
#set_lg_dir logs

# Note: special DB_CONFIG flags are no longer needed for "quick"
# slapadd(8) or slapindex(8) access (see their -q option).
```

- Create a pair of files to initialize the LDAP database by adding the organization entry and the root DN. In this case I create storm.cnaf.infn.it.ldif that contains the organization entry:

```
dn: dc=storm,dc=cnaf,dc=infn.it
objectClass: dcObject
objectClass: organization
dc: storm
o: StoRM
```

and root.storm.cnaf.infn.it.ldif that contains the root DN:

```
dn: cn=root,dc=storm,dc=cnaf,dc=infn.it
objectClass: organizationalRole
cn: root
```

- Initialize DB files for content in /var/lib/ldap directory:

```
echo "" | slapadd -f /etc/openldap/slapd.conf
```

This is required, otherwise you will get this error:

```
bdb_db_open: database "dc=example,dc=com": db_open(/var/lib/ldap/id2entry.bdb) failed:
No such file or directory (2).
```

- On SL6, convert configuration file into dynamic configuration under /etc/openldap/slapd.d directory:

```
slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d
```

- Set permissions:

```
chown -R ldap:ldap /var/lib/ldap
```

and on SL6:

```
chown -R ldap:ldap /etc/openldap/slapd.d
```

- Initialize LDAP DB with initial content:

```
slapadd -l storm.cnaf.infn.it.ldif  
slapadd -l root.storm.cnaf.infn.it.ldif
```

- Now we are ready to add to our LDAP database the necessary users and groups. In particular we need storm and gridhttps users and also relative groups. But, how can we organize our directory tree? In a UNIX file system, the top level is the root. Underneath the root you have numerous files and directories. As mentioned above, LDAP directories are set up in much the same manner. Into the directory's base are conventionally created containers that logically separate data. For historical reasons, most LDAP directories set these logical separations up as OU entries. OU stands for "Organizational Unit", which in X.500 was used to indicate the functional organization within a company. Current LDAP implementations have kept the ou= naming convention. In our case, let's create a pair of Organizational Unit: People and Group.

- Define People organizational-unit into **People.storm.cnaf.infn.it.ldif**:

```
dn: ou=People,dc=storm,dc=cnaf,dc=infn.it  
objectClass: organizationalUnit  
objectClass: top  
ou: People
```

- Define Group organizational-unit into **Group.storm.cnaf.infn.it.ldif**:

```
dn: ou=Group,dc=storm,dc=cnaf,dc=infn.it  
objectClass: organizationalUnit  
objectClass: top  
ou: Group
```

- Define storm user into **storm.People.storm.cnaf.infn.it.ldif**:

```
dn: uid=storm,ou=People,dc=storm,dc=cnaf,dc=infn.it  
objectClass: account  
objectClass: posixAccount  
objectClass: top  
objectClass: shadowAccount  
cn: storm  
gidNumber: 494  
homeDirectory: /home/storm  
uid: storm  
uidNumber: 495
```

- Define storm group into **storm.Group.storm.cnaf.infn.it.ldif**:

```
dn: cn=storm,ou=Group,dc=storm,dc=cnaf,dc=infn.it  
objectClass: posixGroup  
objectClass: top  
cn: storm  
gidNumber: 494
```

- Define gridhttps user into **gridhttps.People.storm.cnaf.infn.it.ldif**:

```
dn: uid=gridhttps,ou=People,dc=storm,dc=cnaf,dc=infn.it
objectClass: account
objectClass: posixAccount
objectClass: top
objectClass: shadowAccount
cn: gridhttps
gidNumber: 504
homeDirectory: /home/gridhttps
uid: gridhttps
uidNumber: 503
```

- Define gridhttps group into **gridhttps.Group.storm.ldif**

```
dn: cn=gridhttps,ou=Group,dc=storm,dc=cnaf,dc=infn.it
objectClass: posixGroup
objectClass: top
cn: gridhttps
gidNumber: 504
```

Set the various GIDs and UIDs values as you want.

- Add the users and groups defined to the LDAP server database:

```
slapadd -l storm.Group.storm.cnaf.infn.it.ldif
slapadd -l gridhttps.Group.storm.cnaf.infn.it.ldif
slapadd -l storm.People.storm.cnaf.infn.it.ldif
slapadd -l gridhttps.People.storm.cnaf.infn.it.ldif
```

Using a free program like Apache Directory Studio we can easily connect to the LDAP server and add other entries, export ldif configurations, etc. Important: as you can see from the files above, a user must contain **account**, **posixAccount** and **shadowAccount** objectClasses, a group instead must define only a **posixGroup** objectClass.

- Start server, on SL6:

```
service slapd start
```

or on SL5:

```
service ldap start
```

3.3.2 LDAP CLIENT INSTALLATION AND CONFIGURATION

- On SL5 install openldap-clients and nss-ldap packages:

```
yum install openldap-clients nss_ldap
```

On SL6:

```
yum install openldap-clients nss-pam-ldapd
```

- Be sure that service nscd is stopped:

```
service nscd status
```

- Modify `/etc/nsswitch.conf` by adding "ldap" to following lines:

```
passwd: files ldap
shadow: files ldap
group: files ldap
```

- On SL5 modify both `/etc/ldap.conf` and `/etc/openldap/ldap.conf` by adding:

```
uri ldap://<ldap-server-hostname>
base dc=storm,dc=cnaf,dc=infn.it
```

- on SL6 modify both `/etc/openldap/ldap.conf` and `/etc/nslcd.conf` by adding:

```
uri ldap://<ldap-server-hostname>
base dc=storm,dc=cnaf,dc=infn.it
```

3.3.3 TEST CLIENT-SERVER LDAP INSTALLATION

From a configured client we need to know UIDs and/or GIDs of server's LDAP users. That users has not to be defined as UNIX-users on clients. To query the LDAP server from one of the clients type for example:

- List all the contents in db:

```
ldapsearch -x -b 'dc=storm,dc=cnaf,dc=infn.it'
```

or search a particular uid or group:

```
ldapsearch -x "uid=storm"
ldapsearch -x "group=storm"
```

- Get the UID or GID of a username:

```
id -u storm
id -g storm
```

Verify that the obtained values are equals to the previous defined.

4 INSTALLATION PREREQUISITES

4.1 OPERATING SYSTEM

All StoRM components are certified to work on Scientific Linux SL5/64 (x86_64) and on Scientific Linux SL6/64 (x86_64) both with EPEL as repository for external dependencies. Therefore **install a proper version of Scientific Linux on your machine(s)**.

All the information about the OS Scientific Linux can be found at <http://www.scientificlinux.org>. SL5 and SL6 are also available in the SL5.X and SL6.X repository respectively mirrored at CERN <http://linuxsoft.cern.ch/scientific/5x/> and <http://linuxsoft.cern.ch/scientific/6x/>.

4.2 NTP SERVICE

NTP service **must** be installed.

To check:

```
[~]# rpm -qa | grep ntp-  
ntp-4.2.2p1-9.el5_4.1  
[~]# chkconfig --list | grep ntpd  
ntpd          0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

To install:

```
[~]# yum install ntp  
[~]# chkconfig ntpd on  
[~]# service ntpd restart
```

4.3 FQDN HOSTNAME

Hostname **must** be set correctly, containing a Fully Qualified Domain Name (FQDN).

To check:

```
[~]# hostname -f  
omii005-vm02.cnaf.infn.it
```

The command must return the host FQDN.

To correct: Unless you are using bind or NIS for host lookups you can change the FQDN and the DNS domain name, which is part of the FQDN, in the `/etc/hosts` file.

```
[~]# cat /etc/hosts  
# Do not remove the following line, or various programs  
# that require network functionality will fail.  
127.0.0.1      omii005-vm02.cnaf.infn.it omii005-vm02 localhost.localdomain localhost  
::1           localhost6.localdomain6 localhost6  
[~]#
```

Add a row with the host IP address and its FQDN and hostname. If FQDN and hostname are specified in the row of the loopback address (127.0.0.1), remove them from the row.

```
[~]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1          localhost.localdomain localhost
::1              localhost6.localdomain6 localhost6
131.154.100.194  omii005-vm02.cnaf.infn.it omii005-vm02
```

Restart the network service.

```
[~]# service network restart
```

4.4 HOST CERTIFICATES FOR STORM NODES

Hosts participating to the StoRM-SE (FE, BE and GridFTP hosts) **must** be configured with X.509 certificates signed by a trusted Certification Authority (CA). Usually the `hostcert.pem` and `hostkey.pem` certificates are located in the `/etc/grid-security/` directory, and they must have permission `0644` and `0400` respectively:

To check:

```
[~]# ls -l /etc/grid-security/hostkey.pem
-r----- 1 root root 887 Mar  1 17:08 /etc/grid-security/hostkey.pem
[~]# ls -l /etc/grid-security/hostcert.pem
-rw-r--r-- 1 root root 1440 Mar  1 17:08 /etc/grid-security/hostcert.pem
[~]# openssl x509 -checkend 0 -in hostcert.pem
```

Certificate will not expire.

To change permission:

```
[~]# chmod 0400 hostkey.pem
[~]# chmod 0644 hostcert.pem
```

4.5 ACL SUPPORT

StoRM uses the ACLs on files and directories to implement the security model. Doing so, StoRM uses the native access to the file system. Therefore in order to ensure a proper running, ACLs need to be enabled on the underlying file system (sometime they are enabled by default) and work properly.

To check:

```
[~]# touch test
[~]# setfacl -m u:storm:rw test
```

Note: the `storm` user used to set the ACL entry **must** exist.

```
[~]# getfacl test
# file: test
# owner: root
# group: root
user::rw-
user:storm:rw-
group::r--
mask::rw-
other::r--
[~]# rm -f test
```

To install ACL (eventually): If the `getfacl` and `setfacl` commands are not available on your host:

```
[~]# yum install acl
```

To enable ACL (if needed): To enable ACL, you must add the `acl` property to the relevant file system in your `/etc/fstab` file. For example:

```
[~]# vi /etc/fstab
...
/dev/hda3    /storage    ext3        defaults, acl    1 2
...
```

Then you need to remount the affected partitions as follows:

```
[~]# mount -o remount /storage
```

This is valid for different file system types (i.e., `ext3`, `xf`s, `gpfs` and others).

4.6 EXTENDED ATTRIBUTE SUPPORT

StoRM uses the Extended Attributes (EA) on files to store some metadata related to the file (e.g. the checksum value); therefore in order to ensure a proper running, the EA support needs to be enabled on the underlying file system and work properly.

Note: Depending on OS kernel distribution, for Reiser3, `ext2` and `ext3` file systems, the default kernel configuration should not enable the EA.

To check:

```
[~]# touch testfile
[~]# setfattr -n user.testea -v test testfile
[~]# getfattr -d testfile
# file: testfile
user.testea="test"
[~]# rm -f testfile
```

To install attr (eventually): If the `getfattr` and `setfattr` commands are not available on your host:

```
[~]# yum install attr
```


To enable EA (if needed): To set extended attributes, you **must** add the `user_xattr` property to the relevant file systems in your `/etc/fstab` file. For example:

```
[~]# vi /etc/fstab
...
/dev/hda3    /storage    ext3        defaults,acl,user_xattr    1 2
...
```

Then you need to remount the affected partitions as follows:

```
[~]# mount -o remount /storage
```

5 INSTALLATION GUIDE

5.1 REPOSITORY SETTINGS

The default package management tool used by SL5/SL6 and YAIM is YUM. In order to install all the stuff requested by StoRM, the following repositories have to be configured in the `/etc/yum.repos.d` directory:

5.1.1 COMMON REPOSITORIES SETTINGS

EPEL Repository Download the EPEL release file

- SL5

```
[~]# wget http://archives.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm  
...
```

- SL6

```
[~]# wget http://www.nic.funet.fi/pub/mirrors/fedora.redhat.com/pub/epel/6/  
x86_64/epel-release-6-8.noarch.rpm  
...
```

and install it.

- SL5

```
[~]# yum localinstall --nogpgcheck epel-release-5-4.noarch.rpm  
...
```

- SL6

```
[~]# yum localinstall --nogpgcheck epel-release-6-8.noarch.rpm  
...
```

EGI Trust Anchors Repository To install the EGI trust anchors follow EGI instructions available at https://wiki.egi.eu/wiki/EGI_IGTF_Release#Using_YUM_package_management

Disable the DAG repository If the DAG repository is enabled, you **must** disable it.

To check:

```
[~]# grep enabled /etc/yum.repos.d/dag.repo  
enabled=0
```

To fix, if needed: To disable the DAG repository you must set to 0 the `enabled` property in your `/etc/yum.repos.d/dag.repo` file:

```
[~]# vi /etc/yum.repos.d/dag.repo  
...  
enabled=0  
...
```

5.1.2 EMI REPOSITORY SETTING

Download and install the EMI release file

- SL5

```
[~]# wget http://emisoft.web.cern.ch/emisoft/dist/EMI/3/sl5/x86_64/base/emi-release-3.0.0-1.sl5.noarch.rpm  
...
```

- SL6

```
[~]# wget http://emisoft.web.cern.ch/emisoft/dist/EMI/3/sl6/x86_64/base/emi-release-3.0.0-1.sl6.noarch.rpm  
...
```

and install it.

- SL5

```
[~]# yum localinstall --nogpgcheck emi-release-3.0.0-1.sl5.noarch.rpm  
...
```

- SL6

```
[~]# yum localinstall --nogpgcheck emi-release-3.0.0-1.sl6.noarch.rpm  
...
```

5.2 INSTALL STORM NODES

Refresh the yum cache:

```
[~]# yum clean all  
...
```

Install the StoRM metapackages you need in every node participating to the StoRM instance.

```
[~]# yum install emi-storm-backend-mp  
...  
[~]# yum install emi-storm-frontend-mp  
...  
[~]# yum install emi-storm-globus-gridftp-mp  
...  
[~]# yum install emi-storm-gridhttps-mp  
...
```

The storm-srm-client is distributed with UI EMI components, but if you need it on your node you can install it using the command:

```
[~]# yum install emi-storm-srm-client-mp  
...
```

6 CONFIGURATION

StoRM is configured by using the YAIM tool, that is a set of configuration scripts that read a set of configuration files.

6.1 GENERAL YAIM VARIABLES

Create a `site-info.def` file in your `<confdir>/` directory. Edit this file by providing a value to the general variables summarized in Table 1.

Table 1: General Variables

Var. Name	Description	Mandatory
SITE_NAME	t is the human-readable name of your site used to set the Glue-SiteName attribute. Example: <code>SITE_NAME="INFN EMI TESTBED"</code>	Yes
BDII_HOST	BDII hostname. <code>BDII_HOST="emitb-bdii-site.cern.ch"</code>	Yes
NTP_HOSTS_IP	Space separated list of the IP addresses of the NTP servers (preferably set a local ntp server and a public one, e.g. pool.ntp.org). Example: <code>NTP_HOSTS_IP="131.154.1.103"</code>	Yes
USERS_CONF	Path to the file containing the list of Linux users (pool accounts) to be created. This file must be created by the site administrator. It contains a plain list of the users and their IDs. An example of this configuration file is given in <code>/opt/glite/yaim/examples/users.conf</code> file. More details can be found in the User configuration section in the YAIM guide.	Yes
GROUPS_CONF	Path to the file containing information on the mapping between VOMS groups and roles to local groups. An example of this configuration file is given in <code>/opt/glite/yaim/examples/groups.conf</code> file. More details can be found in the Group configuration section in the YAIM guide.	Yes
MYSQL_PASSWORD	mysql root password. Example: <code>MYSQL_PASSWORD="carpediem"</code>	Yes
VOS	List of supported VOs. Example: <code>VOS="testers.eu-emi.eu dteam"</code>	Yes

6.2 FRONT-END CONFIGURATION

After having built the `site-info.def` services file, you can configure the profile by using YAIM as follows:

```
/opt/glite/yaim/bin/yaim -c -d 6 -s <site-info.def> -n se_storm_frontend
```

6.2.1 YAIM VARIABLES

Specific variables are in the following file:

```
/opt/glite/yaim/examples/siteinfo/services/se_storm_frontend
```

Please copy and edit that file in your <confdir>/services directory.

You have to set at least the `STORM_DB_PWD` variable and check the other variables to evaluate if you like the default set or if you want to change those settings. Table 2 summaries YAIM variables for storm frontend component.

Table 2: Specific StoRM Frontend Variables

Var. Name	Description
ARGUS_PEPD_ENDPOINTS	The complete service endpoint of Argus PEP server. Mandatory if <code>STORM_FE_USER_BLACKLISTING</code> is true. Example: <code>https://host.domain:8154/authz</code>
STORM_BACKEND_REST_SERVICES_PORT	StoRM backend server rest port. Optional variable. Default value: 9998
STORM_CERT_DIR	Host certificate directory for StoRM Frontend service. Optional variable. Default value: <code>/etc/grid-security/\${STORM_USER}</code>
STORM_DB_HOST	Host for database connection. Optional variable. Default value: localhost
STORM_DB_PWD	Password for database connection. Mandatory.
STORM_DB_USER	User for database connection. Default value: storm
STORM_FE_BE_XMLRPC_HOST	StoRM Backend hostname. Optional variable. Default value: localhost
STORM_FE_BE_XMLRPC_PATH	StoRM Backend XMLRPC server path. Optional variable. Default value: /RPC2
STORM_FE_BE_XMLRPC_PORT	StoRM Backend XMLRPC server port. Optional variable. Default value: 8080
STORM_FE_ENABLE_MAPPING	Enable the check in gridmapfile for client DN. Optional variable. Available values: [true false] Default value: false
STORM_FE_ENABLE_VOMSCHECK	Enable the check in gridmapfile for client VOMS attributes. Optional variable. Available values: [true false] Default value: false
STORM_FE_GSOAP_MAXPENDING	Max number of request pending in the GSOAP queue. Optional variable. Default value: 2000
STORM_FE_LOG_FILE	StoRM frontend log file. Optional variable. Default value: <code>/var/log/storm/storm-frontend.log</code>
STORM_FE_LOG_LEVEL	StoRM Frontend log level. Optional variable. Available values: [KNOWN ERROR WARNING INFO DEBUG DEBUG2] Default value: INFO
STORM_FE_MONITORING_DETAILED	Flag to enable/disable detailed SRM requests Monitoring. Optional variable. Available values: [true false]. Default value: false
STORM_FE_MONITORING_ENABLED	Flag to enable/disable SRM requests Monitoring. Optional variable. Available values: [true false]. Default value: true
STORM_FE_MONITORING_TIME_INTERVAL	Time intervall in seconds between each Monitoring round. Optional variable. Default value: 60
STORM_FE_THREADS_MAXPENDING	Max number of request pending in the Threads queue. Optional variable. Default value: 200

STORM_FE_THREADS_NUMBER	Max number of threads to manage user's requests. Optional variable. Default value: 50
STORM_FE_USER_BLACKLISTING	Flag to enable/disable user blacklisting Optional variable. Available values: [true false]. Default value: false
STORM_FE_WSDL	WSDL to be returned to a GET request. Optional variable. Default value: /usr/share/wsd1/srm.v2.2.wsd1
STORM_FRONTEND_OVERWRITE	This parameter tells YAIM to overwrite storm-frondend.conf configuration file. Optional variable. Available values: [true false]. Default value: true
STORM_FRONTEND_PORT	StoRM Frontend service port. Optional variable. Default value: 8444
STORM_PEPC_RESOURCEID	Argus StoRM resource identifier. Optional variable. Default value: storm
STORM_PROXY_HOME	Directory used to exchange proxies. Optional variable. Default value: /etc/storm/tmp
STORM_USER	Service user. Optional variable. Default value: storm

Please note that:

STORM_CERT_DIR was STORM_HOSTCERT and STORM_HOSTKEY;
STORM_FE_ENABLE_MAPPING was STORM_FE_DISABLE_MAPPING;
STORM_FE_ENABLE_VOMSCHECK was STORM_FE_DISABLE_VOMSCHECK.

6.3 BACK-END CONFIGURATION

After having built the `site-info.def` services file, you can configure the profile by using YAIM as follows:

```
/opt/glite/yaim/bin/yaim -c -d 6 -s <site-info.def> -n se_storm_backend
```

6.3.1 YAIM VARIABLES

Specific variables are in the following file:

```
/opt/glite/yaim/exaples/siteinfo/services/se_storm_backend
```

Please copy and edit that file in your `<confdir>/services` directory.

You have to set at least these variables:

```
STORM_BACKEND_HOST
STORM_DEFAULT_ROOT
STORM_DB_PWD
```

and check the other variables to evaluate if you like the default set or if you want to change those settings. Table 3 summaries YAIM variables for storm backend component.

Table 3: Specific StoRM Backend Variables

Var. Name	Description
STORM_ACLMODE	ACL enforcing mechanism (default value for all Storage Areas). Note: you may change the settings for each SA acting on <code>STORM_<SA>_ACLMODE</code> variable. Available values: [aot jit] (use aot for WLCG experiments). Optional variable. Default value: aot
STORM_ANONYMOUS_HTTP_READ	Storage Area anonymous read access via HTTP. Note: you may change the settings for each SA acting on <code>STORM_<SA>_ANONYMOUS_HTTP_READ</code> variable. Available values: [aot jit] (use aot for WLCG experiments). Optional variable. Available values: [true false] Default value: false
STORM_AUTH	Authorization mechanism (default value for all Storage Areas). Note: you may change the settings for each SA acting on <code>STORM_<SA>_AUTH</code> variable. Available values: [permit-all deny-all <filename>] Optional variable. Default value: permit-all
STORM_BACKEND_HOST	Host name of the StoRM Backend server. Mandatory.
STORM_BACKEND_REST_SERVICES_PORT	StoRM backend server rest port. Optional variable. Default value: 9998
STORM_CERT_DIR	Host certificate directory for StoRM Backend service. Optional variable. Default value: /etc/grid-security/\${STORM_USER}
STORM_DEFAULT_ROOT	Default directory for Storage Areas. Mandatory.
STORM_DB_HOST	Host for database connection. Optional variable. Default value: localhost
STORM_DB_PWD	Password for database connection. Mandatory.
STORM_DB_USER	User for database connection. Optional variable. Default value: storm
STORM_FRONTEND_HOST_LIST	StoRM Frontend service host list: SRM endpoints can be more than one virtual host different from <code>STORM_BACKEND_HOST</code> (i.e. dynamic DNS for multiple StoRM Frontends). Optional variable. Default value: <code>STORM_BACKEND_HOST</code>
STORM_FRONTEND_PATH	StoRM Frontend service path. Optional variable. Default value: /srm/managerv2
STORM_FRONTEND_PORT	StoRM Frontend service port. Optional variable. Default value: 8444
STORM_FRONTEND_PUBLIC_HOST	StoRM Frontend service public host. Optional variable. Default value: <code>STORM_BACKEND_HOST</code>
STORM_FSTYPE	File System Type (default value for all Storage Areas). Note: you may change the settings for each SA acting on <code>STORM_<SA>_FSTYPE</code> variable. Optional variable. Available values: [posixfs gpfs] Default value: posixfs

STORM_GRIDFTP_POOL_LIST	<p>GRIDFTP servers pool list (default value for all Storage Areas). Note: you may change the settings for each SA acting on \$STORM_<SA>_GRIDFTP_POOL_LIST variable. ATTENTION: this variable define a list of pair values space-separated: host weight, e.g.: STORM_GRIDFTP_POOL_LIST="host1 weight1, host2 weight2, host3 weight3" Weight has 0-100 range; if not specified, weight will be 100. Optional variable. Default value: \$STORM_BACKEND_HOST</p>
STORM_GRIDFTP_POOL_STRATEGY	<p>Load balancing strategy for GRIDFTP server pool (default value for all Storage Areas). Note: you may change the settings for each SA acting on \$STORM_<SA>_GRIDFTP_POOL_STRATEGY variable. Optional variable. Available values: [round-robin smart-rr random weight] Default value: round-robin</p>
STORM_GRIDHTTPS_ENABLED	<p>If set to true enables the support of http(s) protocols. Optional variable. Available values: [true false] Default value: false</p>
STORM_GRIDHTTPS_PLUGIN_CLASSNAME	<p>GridHTTPS plugin implementation class. Optional variable. Mandatory to value it.grid.storm.https.GhttpsHTTPSPluginInterface if StoRM GridHTTPS is installed. Default value: it.grid.storm.https.HTTPSPluginInterfaceStub</p>
STORM_GRIDHTTPS_SERVER_USER_UID	<p>StoRM GridHTTPS server service user UID. Mandatory if STORM_GRIDHTTPS_ENABLED is true</p>
STORM_GRIDHTTPS_SERVER_GROUP_UID	<p>StoRM GridHTTPS server service user GID. Mandatory if STORM_GRIDHTTPS_ENABLED is true</p>
STORM_GRIDHTTPS_SERVER_HOST	<p>StoRM GridHTTPS server service host Optional variable. Default value: localhost</p>
STORM_GRIDHTTPS_HTTP_PORT	<p>StoRM GridHTTPS server mapping service port. Optional variable. Default value: 8085</p>
STORM_INFO_FILE_SUPPORT	<p>If set to false, the following variables prevent the corresponding protocol to be published by the StoRM gip. Optional variable. Available values: [true false] Default value: true</p>
STORM_INFO_GRIDFTP_SUPPORT	<p>If set to false, the following variables prevent the corresponding protocol to be published by the StoRM gip. Optional variable. Available values: [true false] Default value: true</p>
STORM_INFO_RFIO_SUPPORT	<p>If set to false, the following variables prevent the corresponding protocol to be published by the StoRM gip. Optional variable. Available values: [true false] Default value: false</p>
STORM_INFO_ROOT_SUPPORT	<p>If set to false, the following variables prevent the corresponding protocol to be published by the StoRM gip. Optional variable. Available values: [true false] Default value: false</p>
STORM_INFO_HTTP_SUPPORT	<p>If set to false, the following variables prevent the corresponding protocol to be published by the StoRM gip. Optional variable. Available values: [true false] Default value: false</p>
STORM_INFO_HTTPS_SUPPORT	<p>If set to false, the following variables prevent the corresponding protocol to be published by the StoRM gip. Optional variable. Available values: [true false] Default value: false</p>

STORM_INFO_OVERWRITE	This parameter tells YAIM to overwrite static-file-StoRM.Idif configuration file. Optional variable. Available values: [true false] Default value: true
STORM_NAMESPACE_OVERWRITE	This parameter tells YAIM to overwrite namespace.xml configuration file. Optional variable. Available values: [true false] Default value: true
STORM_PROXY_HOME	Directory used to exchange proxies. Optional variable. Default value: /etc/storm/tmp
STORM_RFIO_HOST	Rfio server (default value for all Storage Areas). Note: you may change the settings for each SA acting on \$STORM_<SA>_RFIO_HOST variable. Optional variable. Default value: \$STORM_BACKEND_HOST
STORM_ROOT_HOST	Root server (default value for all Storage Areas). Note: you may change the settings for each SA acting on \$STORM_<SA>_ROOT_HOST variable. Optional variable. Default value: \$STORM_BACKEND_HOST
STORM_SERVICE_SURL_DEF_PORTS	comma-separated list of managed SURL's default ports used to check SURL validity. Optional variable. Default value: 8444
STORM_SIZE_LIMIT	Limit Maximum available space on the Storage Area (default value for all Storage Areas). Note: you may change the settings for each SA acting on \$STORM_<SA>_SIZE_LIMIT variable. Optional variable. Available values: [true false] Default value: true
STORM_STORAGEAREA_LIST	List of supported Storage Areas. Usually at least one Storage Area for each VO specified in \$VOS should be created. Optional variable. Default value: \$VOS
STORM_STORAGECLASS	Storage Class type (default value for all Storage Areas). Note: you may change the settings for each SA acting on \$STORM_<SA>_STORAGECLASS variable. Optional variable. Available values: [T0D1 T1D0 T1D1] - No default value.
STORM_SURL_ENDPOINT_LIST	StoRM SURL endpoint list. Optional variable. Default values: srm://\${STORM_FRONTEND_PUBLIC_HOST}:\${STORM_FRONTEND_PORT}\${STORM_FRONTEND_PATH}
STORM_USER	Service user. Optional variable. Default value: storm
STORM_ENDPOINT_QUALITY_LEVEL	Endpoint maturity level to be published by the StoRM gip. Optional variable. Default value: 2
STORM_ENDPOINT_SERVING_STATE	Endpoint serving state to be published by the StoRM gip. Optional variable. Default value: 4
STORM_ENDPOINT_CAPABILITY	Capability according to OGSA to be published by the StoRM gip. Optional variable. Default value: data.management.storage

Please note that:

STORM_BACKEND_HOST was STORM_HOST;
STORM_CERT_DIR was STORM_HOSTCERT and STORM_HOSTKEY;
STORM_FRONTEND_PORT was STORM_PORT;
STORM_FRONTEND_PUBLIC_HOST was STORM_ENDPOINT.

value xfs for variable STORM_FSTYPE is not valid since StoRM 1.9.0.

Table 4 summarized Backend YAIM variables not used in this StoRM version.

Table 4: Backend Variables Not Used

Var. Name	Description
STORM_CKSUM_ALGORITHM	Algorithm used by checksum agents. Available values: [Adler32 CRC32 MD2 MD5 SHA-1 SHA-256 SHA-384 SHA-512] Default value: Adler32
STORM_CKSUM_SERVICE_LIST	Checksum agents list. ATTENTION: this variable define a list of three comma-separated values: host name, status_port, service_port e.g.: STORM_CKSUM_SERVICE_LIST="host1,status_port1,service_port1 host2,status_port2,service_port2" Default value: \${STORM_BACKEND_HOST},9995,9996
STORM_CKSUM_SUPPORT	Enable the support to checksum agents. Available values: [true false] Default value: false

Then, for each <SA> (i.e., Storage Area) listed in the STORM_STORAGEAREA_LIST variable, which is not the name of a valid VO, you have to edit the STORM_<SA>_VONAME compulsory variable (detailed in Table 5). <SA> has to be written in capital letters as in the other variables included in the <site-info.def> file, otherwise default values will be used.

WARNING: for the DNS-like names (using special characters as . (dot), - (minus)) you have to remove the . and -. For example <SA> for STORM_STORAGEAREA_LIST="testers.eu-emi.eu" should be TESTERSEUEMIEU like: STORM_TESTERSEUEMIEU_VONAME=testers.eu-emi.eu

For each <SA> listed in STORM_STORAGEAREA_LIST you have to set at least these variables:

STORM_<SA>_ONLINE_SIZE

You can edit the optional variables summarized in Table 5.

Table 5: Storage Area Variables

Var. Name	Description
STORM_<SA>_VONAME	Name of the VO that will use the Storage Area (use the complete name, e.g., "lights.infn.it"). This variable becomes Mandatory if the value of <SA> is not the name of a VO
STORM_<SA>_DN_C_REGEX	Regular expression specifying the format of C (Country) field of DNs that will use the Storage Area. Optional variable.
STORM_<SA>_DN_O_REGEX	Regular expression specifying the format of O (Organization name) field of DNs that will use the Storage Area. Optional variable.
STORM_<SA>_DN_OU_REGEX	Regular expression specifying the format of OU (Organizational Unit) field of DNs that will use the Storage Area. Optional variable.
STORM_<SA>_DN_L_REGEX	Regular expression specifying the format of L (Locality) field of DNs that will use the Storage Area. Optional variable.
STORM_<SA>_DN_CN_REGEX	Regular expression specifying the format of CN (Common Name) field of DNs that will use the Storage Area. Optional variable.
STORM_<SA>_ACCESSPOINT	Path exposed by the SRM into the SURL. Optional variable. Default value: <sa>

STORM_<SA>_ACLMODE	See STORM_ACLMODE definition. Optional variable. Default value: \$STORM_ACLMODE
STORM_<SA>_AUTH	See STORM_AUTH definition. Optional variable. Default value: \$STORM_AUTH
STORM_<SA>_DEFAULT_ACL_LIST	A list of ACL entries that specifies a set of local groups with corresponding permissions (R, W, RW) using the following syntax: groupname1:permission1 [groupname2:permission2] [...]
STORM_<SA>_FILE_SUPPORT	Enable the corresponding protocol. Optional variable. Default value: \$STORM_INFO_<PROTOCOL>_SUPPORT
STORM_<SA>_GRIDFTP_SUPPORT	Enable the corresponding protocol. Optional variable. Default value: \$STORM_INFO_<PROTOCOL>_SUPPORT
STORM_<SA>_RFIO_SUPPORT	Enable the corresponding protocol. Optional variable. Default value: \$STORM_INFO_<PROTOCOL>_SUPPORT
STORM_<SA>_ROOT_SUPPORT	Enable the corresponding protocol. Optional variable. Default value: \$STORM_INFO_<PROTOCOL>_SUPPORT
STORM_<SA>_HTTP_SUPPORT	Enable the corresponding protocol. Optional variable. Default value: \$STORM_INFO_<PROTOCOL>_SUPPORT
STORM_<SA>_HTTPS_SUPPORT	Enable the corresponding protocol. Optional variable. Default value: \$STORM_INFO_<PROTOCOL>_SUPPORT
STORM_<SA>_FSTYPE	See STORM_<SA>_FSTYPE definition. Optional variable. Available values: [posixfs gpfs] Default value: \$STORM_FSTYPE
STORM_<SA>_GRIDFTP_POOL_LIST	See STORM_GRIDFTP_POOL_LIST definition. Optional variable. Default value: \$STORM_GRIDFTP_POOL_LIST
STORM_<SA>_GRIDFTP_POOL_STRATEGY	See STORM_GRIDFTP_POOL_STRATEGY definition. Optional variable. Default value: \$STORM_GRIDFTP_POOL_STRATEGY
STORM_<SA>_ONLINE_SIZE	Total size assigned to the Storage Area Expressed in GB. Must be an integer value. Mandatory.
STORM_<SA>_USED_ONLINE_SIZE	Storage space currently used in the Storage Area expressed in Bytes. Must be an integer value. Used by YAİM to populate used-space.ini file.
STORM_<SA>_QUOTA	Enables the quota management for the Storage Area and it works only on GPFS filesystem. Optional variable. Available values: [true false] Default value: false
STORM_<SA>_QUOTA_DEVICE	GPFS device on which the quota is enabled. It is mandatory if STORM_<SA>_QUOTA variable is set. No default value.
STORM_<SA>_QUOTA_USER	GPFS quota scope. Only one of the following three will be used (the first one with the highest priority in this order: USER, then GROUP, then FILESET). Optional variable. No default value.
STORM_<SA>_QUOTA_GROUP	GPFS quota scope. Only one of the following three will be used (the first one with the highest priority in this order: USER, then GROUP, then FILESET). Optional variable. No default value.
STORM_<SA>_QUOTA_FILESET	GPFS quota scope. Only one of the following three will be used (the first one with the highest priority in this order: USER, then GROUP, then FILESET). Optional variable. No default value.
STORM_<SA>_RFIO_HOST	See STORM_RFIO_HOST definition. Optional variable. Default value: \$STORM_RFIO_HOST
STORM_<SA>_ROOT	Physical storage path for the VO. Optional variable. Default value: \$STORM_DEFAULT_ROOT/<sa>

STORM_<SA>_ROOT_HOST	See STORM_ROOT_HOST definition. Optional variable. Default value: \$STORM_ROOT_HOST
STORM_<SA>_SIZE_LIMIT	See STORM_SIZE_LIMIT definition. Default value: \$STORM_SIZE_LIMIT
STORM_<SA>_STORAGECLASS	See STORM_STORAGECLASS definition. Available values: [T0D1 T1D0 T1D1 <null>] No default value.
STORM_<SA>_TOKEN	Storage Area token, e.g: LHCb_RAW, INFNGRID_DISK. No default value.

Please note that:

value `xf`s for variable `STORM_<SA>_FSTYPE` is not valid since StoRM 1.9.0.

6.4 GRIDHTTPS CONFIGURATION

After having built the `site-info.def` services file, you can configure the profile by using YAIM as follows:

```
/opt/glite/yaim/bin/yaim -c -d 6 -s <site-info.def> -n se_storm_gridhttps
```

6.4.1 YAIM VARIABLES

Specific variables are in the following file:

```
/opt/glite/yaim/examples/siteinfo/services/se_storm_gridhttps
```

Please copy and edit that file in your `<confdir>/services` directory.

You have to set at least these variables:

```
STORM_BACKEND_HOST
```

and check the other variables to evaluate if you like the default set or if you want to change those settings. Table 6 summaries YAIM variables for storm backend component.

Table 6: Specific StoRM Backend Variables

Var. Name	Description
STORM_BACKEND_HOST	Host name of the StoRM BackEnd server. Mandatory.
STORM_BACKEND_REST_SERVICES_PORT	StoRM BackEnd server REST port. Optional variable. Default value: 9998
STORM_BE_XMLRPC_PORT	StoRM BackEnd server XMLRPC port. Optional variable. Default value: 8080
STORM_FRONTEND_PORT	StoRM FrontEnd server SRM port. Optional variable. Default value: 8444
STORM_GRIDHTTPS_CERT_DIR	Host certificate folder for SSL connector Optional variable. Default value: <code>/etc/grid-security/gridhttps</code>
STORM_GRIDHTTPS_HTTP_ENABLED	Flag that enables/disables http connections Optional variable. Available values: <code>[true false]</code> Default value: <code>false</code>

STORM_GRIDHTTPS_HTTP_PORT	StoRM GridHTTPSs http port Optional variable. Default value: 8085
STORM_GRIDHTTPS_HTTPS_PORT	StoRM GridHTTPSs https port Optional variable. Default value: 8443
STORM_GRIDHTTPS_USER	StoRM GridHTTPSs service user. Optional variable. Default value: gridhttps
STORM_GRIDHTTPS_WEBAPP_DEPLOY_FOLDER	GridHTTPSs web application deploy folder Optional variable. Default value: /var/lib/storm
STORM_SRM_ENDPOINT	StoRM SRM EndPoint Optional variable. Default value: \${STORM_BACKEND_HOST}:\${STORM_FRONTEND_PORT}
STORM_USER	StoRM BackEnd service user. Optional variable. Default value: storm
X509_CERT_DIR	The location of certificates truststore. Optional variable. Default value: /etc/grid-security/certificates
X509_HOST_CERT	Host certificate location. Default value: /etc/grid-security/hostcert.pem
X509_HOST_KEY	Host certificate key location. Optional variable. Default value: /etc/grid-security/hostkey.pem

6.5 CONFIGURING MULTIPLE SERVICES ON A SINGLE HOST

If in your StoRM deployment scenario more than a StoRM service has been installed on a single host you have to provide a single `site-info.def` services file containing all the required YAIM variables. Then you can configure all service profiles at once with a single YAIM call:

```
/opt/glite/yaim/bin/yaim -c -d 6 -s <site-info.def> -n <node_type_1> -n <node_type_2> -n <node_type_3>
```

where for example `node_type_1` is `se_storm_backend`, `node_type_2` is `se_storm_frontend` and `node_type_3` is `se_storm_gridftp`

NOTE: if you are configuring on the same host profiles `se_storm_backend` and `se_storm_frontend` you have to specify those profiles in this order to YAIM. This is also the case of profiles `se_storm_backend` and `se_storm_gridhttps`

7 ADVANCED CONFIGURATION

Please note that most of the configuration parameters of StoRM can be automatically managed directly by YAIM. This means that for standard installation in WLCG site without special requirement is not needed a manual editing of StoRM configuration file, but only a proper tuning of StoRM YAIM variables. On the other hand, with this guide we would like to give to site administrators the opportunity to learn about StoRM details and internal behaviours, in order to allow advanced con

figuration and ad-hoc set up, to optimize performance and results.

7.1 FRONTEND CONFIGURATION

The Frontend component relies on a single configuration file that contains all the configurable parameters. This file is:

- `/etc/storm/frontend-server/storm-frontend-server.conf`

containing a list of:

- `key = value`

pairs that can be used to configure the Frontend server.

In case a parameter is modified, **the Frontend service have to be restarted in order to read the new value.**

7.1.1 FRONTEND SERVICE INFORMATION: STORM-FRONTEND-SERVER.CONF

Property name	Description
Database settings	
<code>db.host</code>	Host for database connection. Default is <code>localhost</code>
<code>db.user</code>	User for database connection. Default is <code>storm</code>
<code>db.passwd</code>	Password for database connection. Default is <code>password</code>
Frontend service settings	
<code>fe.port</code>	Frontend port. Default is <code>8444</code>
<code>fe.threadpool.threads.number</code>	Size of the worker thread pool. Default is <code>50</code>
<code>fe.threadpool.maxpending</code>	Size of the internal queue used to maintain SRM tasks in case there are no free worker threads. Default is <code>200</code>
<code>fe.gsoap.maxpending</code>	Size of the GSOAP queue used to maintain pending SRM requests. Default is <code>2000</code>
Log settings	
<code>log.filename</code>	Log file name, complete whit path. Default is <code>/var/log/storm/storm-frontend.log</code>
<code>log.debuglevel</code>	Loggin level. Possible value are: <code>ERROR WARN INFO DEBUG DEBUG2</code> . Default is <code>INFO</code>
Monitoring settings	
<code>monitoring.enabled</code>	Flag to enable/disable SRM requests Monitoring. Default is <code>true</code>
<code>monitoring.timeInterval</code>	Time intervall in seconds between each Monitoring round. Default is <code>60</code>
<code>monitoring.detailed</code>	Flag to enable/disable detailed SRM requests Monitoring. Default is <code>false</code>
XML-RPC communication settings	
<code>be.xmlrpc.host</code>	BackEnd hostname. Default is <code>localhost</code>

be.xmlrpc.port	XML-RPC server port running on the BackEnd machine. Default is 8080
be.xmlrpc.path	XML-RPC server path. Default is /RPC2
be.xmlrpc.check.ascii	Flag to enable/disable ASCII checking on strings to be sent via XML-RPC. Default is true
REST communication settings	
be.recalltable.port	REST server port running on the BackEnd machine. Default is 9998
Blacklisting settings	
check.user.blacklisting	Flag to enable/disable user blacklisting. Default is false
argus-pepd-endpoint	The complete service endpoint of Argus PEP server. Mandatory if check.user.blacklisting is true. Example https://host.domain:8154/authz
Proxy settings	
proxy.dir	Directory used by the Frontend to save proxies files in case of requests with delegation. Default is /var/tmp/storm/proxy
proxy.user	Local user owner of proxies files. This have to be the same local user running the backend service Mandatory .
security.enable.vomscheck	Flag to enable/disable checking proxy VOMS credentials. Default is true
security.enable.mapping	Flag to enable/disable DN->userid mapping via gridmap-file. Default is false
General settings	
wSDL.file	WSDL file, complete with path, to be returned in case of GET request

7.1.2 LOGGING FILES AND LOGGING LEVEL

The Frontend logs information on the service status and the SRM request received and managed by the process. The Frontend logs support different level of logging, `ERROR|WARNING|INFO|DEBUG|DEBUG2`, that can be setted from the dedicated parameter in the `storm-frontend.conf` configuration file.

The Frontend log file named `storm-frontend.log` is placed in the:

`/var/log/storm` directory. At start-up time, the FE prints here the whole set of configuration parameters, this can be useful to check desired values.

When a new SRM request is managed, the FE logs information about the user (DN and FQANs) and the requests parameters. At each SRM request, the FE logs also this important information:

```
03/19 11:51:42 0x88d4ab8 main: AUDIT - Active tasks: 3
03/19 11:51:42 0x88d4ab8 main: AUDIT - Pending tasks: 0
```

these information are about the status of the worker pool threads and the pending process queue.

- `Active tasks:1` is the number of worker threads actually running.
- `Pending tasks: 0` is the number of SRM requests queued in the worker pool queue in case there are no more free worker threads.

These data gives you important information on the Frontend load.

Monitoring

If service monitoring is enabled information about the operations executed in a certain amount of time are recorded in the monitoring file:

```
/var/log/storm/monitoring.log.
```

This amount of time (called `Monitoring Round`) is configurable via the configuration property `monitoring.timeInterval`; its default value is 1 minute.

At each `Monitoring Round` in a single row the following information are reported for both `Synchronous` and `Asynchronous` requests:

- how many requests have been performed in the last `Monitoring Round`,
- how many of them were successful,
- how many failed,
- how many produced an error,
- the average execution time,
- the minimum execution time,
- the maximum execution time.

Then on the same line those information are reported considering the whole frontend execution time (the `Aggregate Monitoring`). This row reports the `Monitoring Summary`.

This is the default behaviour of the monitoring service.

Example

```
03/20 14:19:11 : [# 22927 lifetime=95:33:18] S [OK:47,F:15,E:0,m:0.085,M:3.623,Avg:0.201]
                                     A [OK:16,F:0,E:0,m:0.082,M:0.415,Avg:0.136]
                                     Last: (S [OK:12,F:5,E:0,m:0.091,M:0.255]
                                     A [OK:6,F:0,E:0,m:0.121,M:0.415])
```

Furthermore it can be requested a more detailed `Frontend Monitoring` activity by setting the configuration property `monitoring.detailed` to `true`.

Doing so at each `Monitoring Round` for each kind of `srm` operation performed in the `Monitoring Round` (`srmIs`, `srmPtp`, `srmRm`, ...) the following information are printed in a section with header "Last round details:"

- how many request succeeded,
- how many failed,
- how many produced an error,
- the average execution time,
- the minimum execution time,
- the maximum execution time,
- the execution time standard deviation.

This is the `Detailed Monitoring Round`.

Then, after the `Monitoring Summary` information the same information are reported in a section with header "Details:" but considering the whole frontend execution time.

This is the `Aggregate Detailed Monitoring`.

Example


```
03/20 14:19:11 : Last round details:
03/20 14:19:11 : [PTP] [OK:3,F:0,E:0,Avg:0.203,Std Dev:0.026,m:0.183,M:0.240]
03/20 14:19:11 : [Put done] [OK:2,F:0,E:0,Avg:0.155,Std Dev:0.018,m:0.136,M:0.173]
03/20 14:19:11 : [# 22927 lifetime=95:33:18] S [OK:47,F:15,E:0,m:0.085,M:3.623,Avg:0.201]
      A [OK:16,F:0,E:0,m:0.082,M:0.415,Avg:0.136]
      Last: (S [OK:12,F:5,E:0,m:0.091,M:0.255]
      A [OK:6,F:0,E:0,m:0.121,M:0.415])

03/20 14:19:11 : Details:
03/20 14:19:11 : [PTP] [OK:7,F:0,E:0,Avg:0.141,Std Dev:0.057,m:0.085,M:0.240]
03/20 14:19:11 : [Put done] [OK:5,F:0,E:0,Avg:0.152,Std Dev:0.027,m:0.110,M:0.185]
03/20 14:19:11 : [Release files] [OK:4,F:0,E:0,Avg:0.154,Std Dev:0.044,m:0.111,M:0.216]
03/20 14:19:11 : [Rm] [OK:3,F:0,E:0,Avg:0.116,Std Dev:0.004,m:0.111,M:0.122]
```

Note

Operations not performed in current Monitoring Round are not printed in Detailed Monitoring Round.

Operations never performed are not printed in Aggregate Detailed Monitoring.

Operation performed in current Monitoring Round are aggregated in Aggregate Detailed Monitoring

gSOAP tracefile

If you have problem at gSOAP level, and you have already looked at the **troubleshooting** section of the StoRM site without finding a solution, and you are brave enough, you could try to find some useful information on the gSOAP log file.

Set the following environment variables :

```
$CGSI_TRACE=1
$CGSI_TRACEFILE=/tmp/tracefile
```

restart the FrontEnd daemon by calling directly the init script `/etc/init.d/storm-frontend-server` and see if the error messages contained in `/tmp/tracefile` could help. Please be very careful, it prints really a huge amount of information.

7.2 BACKEND CONFIGURATION

The BackEnd is the core of StoRM. It executes all SRM requests, interacting with other Grid service, with database to retrieve SRM requests, with file systems to set up space and file, etc. It has a modular architecture made by several internal components. The BackEnd need to be configured for two main aspects:

- Service information: this section contains all the parameter regarding the StoRM service details. It relies on the `storm.properties` configuration file.
- Storage information: this section contains all the information regarding Storage Area and other storage details. It relies on the `namespace.xml` file.

7.2.1 BACKEND SERVICE INFORMATION: STORM.PROPERTIES

The file:

- `/etc/storm/backend-server/storm.properties`

contains a list of:

- key = value

pairs that represent all the information needed to configure the StoRM BackEnd service. The most important (and mandatory) parameters are configured by default through YAIM with a standard installation of StoRM. All the other parameters are optional and can be used to make advanced tuning of the BackEnd.

To change set a new value, or add a new parameter, just edit the `storm.properties` file and **restart the BackEnd daemon**.

When the BackEnd start, it writes into the log file the whole set of parameters read from the configuration file.

Property name	Description
Service information	
<code>storm.service.SURL.endpoint</code>	List of comma separated strings identifying the StoRM FrontEnd endpoint(s). This is used by StoRM to understand if a SURL is local. E.g. <code>srm://storm.cnaf.infn.it:8444/srm /managerv2</code>
<code>storm.service.port</code>	SRM service port. Default:8444
<code>storm.service.SURL.default-ports</code>	List of comma separated valid SURL port numbers. Default:8444
<code>storm.service.FE-public.hostname</code>	StoRM FrontEnd hostname in case of a single FrontEnd StoRM deployment, StoRM FrontEnds DNS alias in case of a multiple FrontEnds StoRM deployment.
<code>storm.service.FE-list.hostnames</code>	Comma separated list of FrontEnd(s) hostname(s). Default:localhost
<code>storm.service.FE-list.IPs</code>	Comma separated list of FrontEnd(s) IP(s). E.g. <code>131.154.5.127, 131.154.5.128</code> . Default:127.0.0.1
<code>proxy.home</code>	Directory used to contains delegated proxies used in case of <code>srmCopy</code> request. Please note that in case of clustered installation this directory have to be shared between the BackEnd and the FrontEnd(s) machines. Default: <code>/etc/storm/tmp</code>
<code>pinLifetime.default</code>	Default PinLifetime in seconds used for pinning files in case of <code>srmPrepareToPut</code> or <code>srmPrepareToGet</code> operation without any pinLifetime specified. Default:259200
<code>pinLifetime.maximum</code>	Maximum PinLifetime allowed in seconds. Default:1814400
<code>SRM22Client.PinLifeTime</code>	Default PinLifeTime in seconds used by StoRM in case of <code>SrmCOpy</code> operation. This value is the one specified in the remote <code>SrmPrepareToGet</code> request. Default:259200
<code>fileLifetime.default</code>	Default FileLifetime in seconds used for VOLATILE file in case of SRM request without FileLifetime parameter specified. Default: 3600
<code>extraslashes.gsiftp</code>	Add extra slashes after the "authority" part of a TURL for gsiftp protocol. e.g. <code>gsiftp://<hostname>:port/<PhysicalFN></code> . E.g. /
<code>extraslashes.rfio</code>	Add extra slashes after the "authority" part of a TURL for rfio protocol. e.g. <code>rfio://<hostname>:port/<PhysicalFN></code> . E.g. /
<code>extraslashes.root</code>	Add extra slashes after the "authority" part of a TURL for root protocol. e.g. <code>root://</code> . E.g. /
<code>extraslashes.file</code>	Add extra slashes after the "authority" part of a TURL for file protocol. e.g. <code>file:////</code> . E.g. /
<code>checksum.enabled</code>	Flag to enable or not the support of Adler32 checksum computation. Default: false

synchcall.directoryManager.maxLsEntry	Maximum number of entries returned by an srmLs call. Since in case of recursive srmLs results can be in order of million, this prevent a server overload. Default: 500
directory.automatic-creation	Flag to enable authomatic missing dyrectory creation upon srmPrepareToPut requests. Default:false
directory.writeperm	Flag to enable dyrectory write permission setting upon srmMkDir requests on created dyrectories. Default:false
default.overwrite	Default file overwrite mode to use upon srmPrepareToPut and srmCopy requests. Default:A. Possible values are: N A D. Please note that N stands for Never, A stands for Always and D stands for When files differs.
default.storagetype	Default File Storage Type to be used for srmPrepareToPut and srmCopy requests in case is not provided in the request. Default:V. Possible values are: V P D. Please note that V stands for Volatile, P stands for Permanent and D stands for Durable.
Requests garbage collector	
purging	Flag to enable the purging of expired requests. This garbage collector process cleans all database tables and proxies from the expired SRM requests. An appropriate tuning is needed in case of high throughput of SRM requests required for long time. Default:true. Possible values are: [true false]
purge.interval	Time interval in seconds between successive purging run. Default:600.
purge.size	Number of requests picked up for cleaning from the requests garbage collector at each run. This value is use also by Tape Recal Garbage Collector. Default:800
purge.delay	Initial delay before starting the requests garbage collection process, in seconds. Default: 10
expired.request.time	Time in seconds to consider a request expired after its submission. Default: 604800
Garbage collector	
gc.pinnedfiles.cleaning.delay	Initial delay before starting the reserved space, JIT ACLs and pinned files garbage collection process, in seconds. Default:10
gc.pinnedfiles.cleaning.interval	Time interval in seconds between successive purging run. Default:300
Synchronous call	
synchcall.xmlrpc.unsecureServerPort	Port to listen on for Incoming XML-RPC connections from FrontEnds(s). Default:8080
synchcall.xmlrpc.maxthread	Number of threads managing XML-RPC connection from FrontEnds(s). A well sized value for this parameter have to be at least equal to the sum of the number of working threads in all FrontEend(s). Default: 100
REST interface parameters	
storm.rest.services.port	REST services port. Default: 9998
Database connection parameters	
com.mysql.jdbc.Driver	JDBC driver to be used to connect with StoRM database. Default: com.mysql.jdbc.Driver
storm.service.request-db.protocol	Protocol to be used to connect with StoRM database. Default: jdbc:mysql://
storm.service.request-db.host	Host for StoRM database. Default: localhost

storm.service.request-db.db-name	Database name for SRM requests. Default: storm_db
storm.service.request-db.username	Username for database connection. Default: storm
storm.service.request-db.passwd	Password for database connection
asynch.db.ReconnectPeriod	Database connection refresh time intervall in seconds. Default: 18000
asynch.db.DelayPeriod	Database connection refresh initial delay in seconds. Default: 30
persistence.internal-db.connection-pool	Enable the database connection pool. Default:false
persistence.internal-db.connection-pool.maxActive	Database connection pool max active connections. Default:10
persistence.internal-db.connection-pool.maxWait	Database connection pool max wait time to provide a connection. Default:50
SRM Requests Picker	
asynch.PickingInitialDelay	Initial delay before starting to pick requests from the DB, in seconds. Default: 1
asynch.PickingTimeInterval	Polling interval in seconds to pick up new SRM requests. Default:2
asynch.PickingMaxBatchSize	Maximum number of requests picked up at each polling time. Default:100
scheduler.serial	DEPRECATED Flag to enable the execution of all the request on a single thread. Default:false
Worker threads	
scheduler.crusher.workerCorePoolSize	Crusher Scheduler worker pool base size. Default:10
scheduler.crusher.workerMaxPoolSize	Crusher Schedule worker pool max size. Default:50
scheduler.crusher.queueSize	Request queue maximum size. Default:2000
scheduler.chunksched.ptg.workerCorePoolSize	Prepare to Get worker pool base size. Default:50
scheduler.chunksched.ptg.workerMaxPoolSize	Prepare to Get Worker pool max size. Default:200
scheduler.chunksched.ptg.queueSize	Prepare to Get request queue maximum size. Default:2000
scheduler.chunksched.ptp.workerCorePoolSize	Prepare to Put worker pool base size. Default:50
scheduler.chunksched.ptp.workerMaxPoolSize	Prepare to Put Worker pool max size. Default:200
scheduler.chunksched.ptp.queueSize	Prepare to Put request queue maximum size.Default:1000
scheduler.chunksched.bol.workerCorePoolSize	Bring Online worker pool base size. Default:50
scheduler.chunksched.bol.workerMaxPoolSize	Bring Online Worker pool max size. Default:200
scheduler.chunksched.bol.queueSize	Bring Online request queue maximum size.Default:2000
scheduler.chunksched.copy.workerCorePoolSize	Copy worker pool base size. Default:10
scheduler.chunksched.copy.workerMaxPoolSize	Copy worker pool max size. Default:50

scheduler.chunksched.copy.queueSize	Copy request queue maximum size. Default:500
HTTP(S) protocol	
gridhttps.enabled	Flag to enable the support to HTTP and HTTPS protocols. Default:false
gridhttps.server.host	The complete hostname of the host running StoRM Grid-HTTPs. Default:localhost
gridhttps.server.port	The port on StoRM GridHTTPs host where Tomcat accepts HTTP connections. Default:8088
gridhttps.plugin.classname	The complete class-name of the HTTP-SPluginInterface implementation to be used. Default:it.grid.storm.https.HTTPSPluginInterfaceStub
Protocol balancing	
gridftp-pool.status-check.timeout	Time time in milliseconds after which the status of a GridFTP has to be verified. Default:20000 (20 Seconds)
Tape recall	
tape.support.enabled	Flag to enable tape support. Default:false
tape.buffer.group.read	System group to be assigned to files migrated from tape storage. Default:storm-SA-read
tape.buffer.group.write	System group to be assigned to files migrated to tape storage. Default:storm-SA-write
srmCopy parameters	
asynch.srmclient.retrytime	Timeout for a single srmPrepareToPut request execution performed to fulfill srmCopy requests in seconds. Default:60
asynch.srmclient.sleepTime	Interval between successive srmPrepareToPut request status polling performed to fulfill srmCopy requests in seconds. Default:5
asynch.srmclient.timeout	Timeout for srmPrepareToPut request execution performed to fulfill srmCopy requests in seconds. Default:180
asynch.srmclient.putdone.sleepTime	Interval between consecutive srmPutDone attempts performed to fulfill srmCopy requests in seconds. Default:1
asynch.srmclient.putdone.timeout	Timeout for srmPutDone request execution performed to fulfill srmCopy requests in seconds. Default:60
asynch.srmclient	The complete class-name of the SRMClient implementation providing SRM client features to be used to perform srm operations to fulfill srmCopy requests. Default:it.grid.storm.asynch.SRM22Client
asynch.srmcopy.gridftp.timeout	Timeout for GridFTP connection establishment during file transfer execution performed to fulfill srmCopy requests in seconds. Default:15000
asynch.gridftpclient	The complete class-name of the GridFTPTransferClient implementation providing GridFTP client features to be used to perform file transfer to fulfill srmCopy requests. Default:it.grid.storm.asynch.NaiveGridFTPTransferClient

7.2.2 BACKEND STORAGE INFORMATION: NAMESPACE.XML

Information about storage managed by StoRM is stored in a configuration file named namespace.xml located at /etc/storm/backend-server/ on StoRM BackEnd host. namespace.xml is the place where StoRM stores, together

with other information, the information needed to perform its mapping functionality.

The mapping functionality is the process of retrieving or building the transport URL (TURL) of a file addressed by a Site URL (SURL) together with grid user credential. The Fig 1 shows the different schema of SURL and TURL. A couple of quick concepts from SRM:

- The SURL is the logical identifier for a local data entity
- Data access and data transfer are made through the TURLs
- The TURL identify a physical location of a replica
- SRM services retrieve the TURL from a namespace database (like DPNS component in DPM) or build it through other mechanisms (like StoRM)

In StoRM, the mapping functionality is provided by the namespace component (NS).

- The Namespace component works without a database.
- The Namespace component is based on an XML configuration.
- It relies on the physical storage structure.

The basic features of the namespace component are:

- The configuration is modular and structured (representation is based on XML)
- The loading and the parsing of the configuration file occurs:
 - at start-up of the back-end service
 - when configuration file is modified
- An efficient structure of namespace configuration lives in memory.
- No access to disk or database is performed

StoRM is different from the other solution, where typically, for every SRM request a query to the data base have to be done in order to establish the physical location of file and build the correct transfer URL.

The namespace functions relies on two kind of parameters for mapping operations derived from the SRM requests, that are:

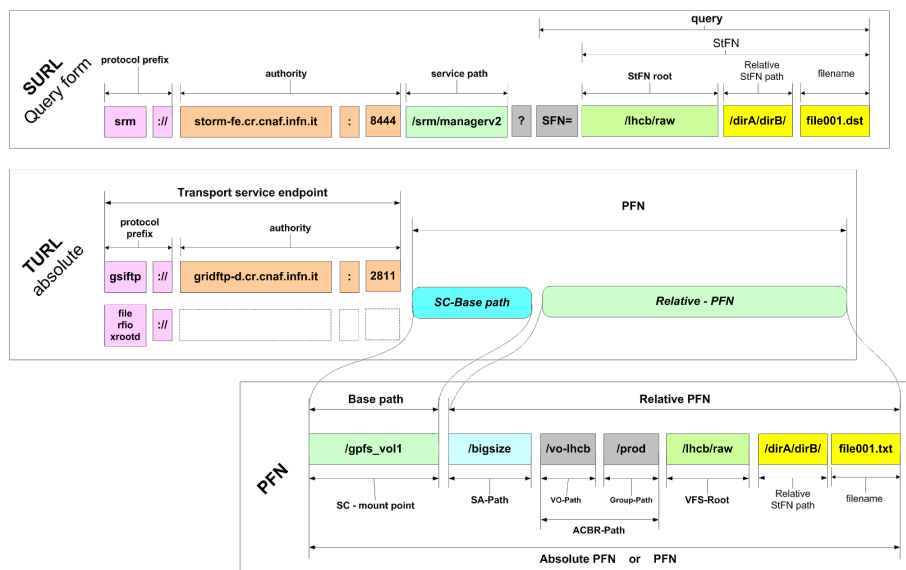


Figure 1: Site URL and Transfer URL schema.

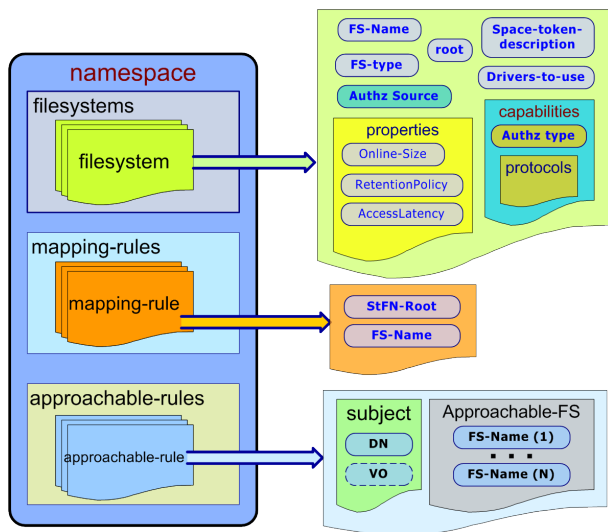


Figure 2: Namespace structure

- the grid user credential (a subject or a service acting on behalf of the subject)
- the SURL

The Fig.2 shows the main concepts of Namespace Component:

- NS-Filesystem: is the representation of a Storage Area
- Mapping rule: represents the basic rule for the mapping functionalities
- Approachable rule: represents the coarse grain access control to the Storage Area.

This is an example of the FS element:

```
<filesystem name="dteam-FS" fs_type="ext3">
  <space-token-description>DTEAM_TOKEN</space-token-description>
  <root>/storage/dteam</root>
  <filesystem-driver>it.grid.storm.filesystem.swig.posixfs</filesystem-driver>
  <spacesystem-driver>it.grid.storm.filesystem.MockSpaceSystem</spacesystem-driver>
  <storage-area-authz>
    <authz-db>DTEAM_AUTH</authz-db>
  </storage-area-authz>
  <properties>
    <RetentionPolicy>replica</RetentionPolicy>
    <AccessLatency>online</AccessLatency>
    <ExpirationMode>neverExpire</ExpirationMode>
    <TotalOnlineSize unit="GB" limited-size="true">291</TotalOnlineSize>
    <TotalNearlineSize unit="GB">0</TotalNearlineSize>
  </properties>
  <capabilities>
    <aclMode>AoT</aclMode>
    <default-acl>
      <acl-entry><groupName>lhcb</groupName><permissions>RW</permissions></acl-entry>
    </default-acl>
  </capabilities>
  <trans-prot>
```

```

<prot name="file">
  <schema>file</schema>
</prot>
<prot name="gsiftp">
  <id>0</id>
  <schema>gsiftp</schema>
  <host>gsiftp-dteam-01.cnaf.infn.it</host>
  <port>2811</port>
</prot>
<prot name="gsiftp">
  <id>1</id>
  <schema>gsiftp</schema>
  <host>gsiftp-dteam-02.cnaf.infn.it</host>
  <port>2811</port>
</prot>
<prot name="rfio">
  <schema>rfio</schema>
  <host>rfio-dteam.cnaf.infn.it</host>
  <port>5001</port>
</prot>
<prot name="root">
  <schema>root</schema>
  <host>root-dteam.cnaf.infn.it</host>
  <port>1094</port>
</prot>
</trans-prot>
<pool>
  <balance-strategy>round-robin</balance-strategy>
  <members>
    <member member-id="0"></member>
    <member member-id="1"></member>
  </members>
</pool>
</capabilities>
<defaults-values>
  <space lifetime="86400" type="volatile" guarsize="291" totalsize="291"/>
  <file lifetime="3600" type="volatile"/>
</defaults-values>
</filesystem>

```

Attributes meaning:

- `<filesystem name="dteam-FS" fs_type="ext3">`
The name is the element identifier. It identifies this Storage Area in the namespace domains. The `fs_type` is the type of the filesystem the Storage Area is built on. Possible values are: `ext3|gpfs`. Please note that `ext3` stands for all generic POSIX filesystem (`ext3`, `Lustre`, etc.)
- `<space-token-description>DTEAM_TOKEN</space-token-description>`
Storage Area space token description.
- `<root>/storage/dteam</root>`
Physical root directory of the Storage Area on the file system.
- `<filesystem-driver>it.grid.storm.filesystem.swig.posixfs </filesystem-driver>`
Driver loaded by the BackEnd for filesystem interaction. This driver is used mainly to set up ACLs on space and files.

- `<spacesystem-driver>it.grid.storm.filesystem.MockSpaceSystem</spacesystem-driver>`
Driver loaded by the BackEnd for filesystem interaction. This driver is used to manage space allocation. (E.g. on GPFS it uses the `gpfs_prealloc()` call)

```
<storage-area-authz>
  <authz-db>DTEAM_AUTH</authz-db>
</storage-area-authz>
```

- **Storage Area properties**

```
<properties>
  <RetentionPolicy>replica</RetentionPolicy>
  <AccessLatency>online</AccessLatency>
  <ExpirationMode>neverExpire</ExpirationMode>
  <TotalOnlineSize unit="GB" limited-size="true">291</TotalOnlineSize>
  <TotalNearlineSize unit="GB">0</TotalNearlineSize>
</properties>
```

in details:

- `<RetentionPolicy>replica</RetentionPolicy>`
Retention Policy of the Storage Area. Possible values are: `replica|custodial`.
- `<AccessLatency>online</AccessLatency>`
Access Latency of the Storage Area. Possible values: `online|nearline`.
- `<ExpirationMode>neverExpire</ExpirationMode>`
Expiration Mode of the Storage Area. Deprecated.
- `<TotalOnlineSize unit="GB" limited-size="true">291</TotalOnlineSize>`
Total on line size of the Storage Area in GigaBytes. In case the attribute `limited-size="true"`, StoRM enforce this limit at SRM level. When the space used for the Storage Area is at least equal to the size specified, every further SRM request to write files will fail with `SRM_NO_FREE_SPACE` error code.
- `<TotalNearlineSize unit="GB">0</TotalNearlineSize>` Total near line size of the Storage Area. This only means in case the Storage Area is in some way attached to a MSS storage system (such as TSM with GPFS)

- **Storage area capabilities:**

```
<aclMode>AoT</aclMode>
```

This is the ACL enforcing approach. Possible values are: `AoT|JiT`. In case of AheadOfTime(AoT) approach StoRM sets up a physical ACL on file and directories for the local group (`gid`) in which the user is mapped. (The mapping is done querying the LCMAPS service on the BE machine passing both user DN and FQANs). The group ACL remains for the whole lifetime of the file. In case of JustInTime(JiT) approach StoRM sets up and ACL for the local user (`uid`) the user is mapped. The ACL remains in place only for the lifetime of the SRM request, then StoRM removes it. (This is to avoid to grant access to pool account uid in case of reallocation on different users.)

```
<default-acl>
  <acl-entry><groupName>lhcb</groupName><permissions>RW</permissions></acl-entry>
</default-acl>
```

This is the Default ACL list. A list of ACL entry (that specify a local user (uid) or group id (gid) and a permission(R,W,RW). This ACL are automatically by StoRM at each read or write request. Useful for use cases where experiment want to allow local access to file on group different than the one that made the SRM request operation.

- **Access and Transfer protocol supported:**

```
<prot name="file">
  <schema>file</schema>
</prot>
```

The `file` protocol means the capability to perform local access on file and directory. If user performs an SRM request (`srmPtG` or `srmPtP`) specifying the file protocol, and it is supported by the selected Storage Area, StoRM return a TURL structured as: `file:///atlas/atlasdisk/filename`. This TURL can be used through GFAL or other SRM clients to perform a direct access on the file.

```
<prot name="gsiftp">
  <id>0</id>
  <schema>gsiftp</schema>
  <host>gridftp-dteam.cnaf.infn.it</host>
  <port>2811</port>
</prot>
```

The `gsiftp` protocol is GridFTP transfer system from Globus widely adopted in many Grid environments. This capability element contains all the information about the GridFTP server to use with this Storage Area. Site administrator can decide to have different server (or pools of server) for different Storage Areas. The `id` is the server identifier to be used when defining a pool. The `schema` have to be `gsiftp`. `host` is the hostname of the server (or the DNS alias used to aggregate more than one server). The `port` is the GridFTP server port, typically 2811. If user performs an SRM request (`srmPtG` or `srmPtP`) specifying the `gsiftp` protocol, and it is supported by the selected Storage Area, StoRM return a TURL structured as: `gsiftp://gridftp-dteam.cnaf.infn.it:2811/atlas/atlasdisk/filename`.

```
<prot name="rfio">
  <schema>rfio</schema>
  <host>rfio-dteam.cnaf.infn.it</host>
  <port>5001</port>
</prot>
```

The `rfio` protocol. This capability element contains all the information about the `rfio` server to use with this Storage Area. Like for GridFTP, site administrator can decide to have different server (or pools of server) for different Storage Areas. The `id` is the server identifier. The `schema` have to be `rfio`. `host` is the hostname of the server (or the DNS alias used to aggregate more than one server). The `port` is the `rfio` server port, typically 2811. If user performs an SRM request (`srmPtG` or `srmPtP`) specifying the `rfio` protocol, and it is supported by the selected Storage Area, StoRM return a TURL structured as: `rfio://rfio-dteam.cnaf.infn.it:5001/atlas/atlasdisk/filename`.

```
<schema>root</schema>
  <host>root-dteam.cnaf.infn.it</host>
  <port>1094</port>
</prot>
</trans-prot>
```

The `root` protocol. This capability element contains all the information about the root server to use with this Storage Area. Like for other protocols, site administrator can decide to have different server (or pools of server) for different Storage Areas. The `id` is the server identifier. The `schema` have to be `root`. `host` is the hostname of the server (or the DNS alias used to aggregate more than one server). The `port` is the root server port, typically 1094. If user performs an SRM request (`srnPtG` or `srnPtP`) specifying the `root` protocol, and it is supported by the selected Storage Area, StoRM return a TURL structured as: `root://root-dteam.cnaf.infn.it:1094/atlas/atlasncdisk/filename`.

```
<pool>
  <balance-strategy>round-robin</balance-strategy>
  <members>
    <member member-id="0"></member>
    <member member-id="1"></member>
  </members>
</pool>
```

Here is defined a pool of protocol servers. Within the pool element pool members are declared identified by their `id`, the list of members have to be homogenous with respect to their schema. This `id` is the server identifier specified in the `prot` element. The `balance-strategy` represent the load balancing strategy with which the pool has to be managed. Possible values are: `round-robin|smart-rr|random|weight`.

NOTE: Protocol server pooling is currently available only for `gsiftp` servers.

Load balancing strategies details:

- `round-robin`: At each TURL construction request the strategy returns the next server following the round-robin approach: a circular list with an index starting from the head and incremented at each request
- `smart-rr`: An enhanced version of round-robin. The status of pool members is monitored and maintained in a cache. Cache entries has a validity life time that is refreshed when expired. If the member chosen by round-robin is marked as not responsive another iteration of round-robin is performed.
- `random`: At each TURL construction request the strategy returns a random member of the pool.
- `weight`: An enhanced version of round-robin. When a server is chosen the list index will not be moved forward (and the server will be chosen again in next request) for as many times as specified in its weight.

NOTE: The weight has to be specified in a `weight` element inside the member element:

```
<pool>
  <balance-strategy>WEIGHT</balance-strategy>
  <members>
    <member member-id="0">
      <weight>5</weight>
    </member>
    <member member-id="1">
      <weight>1</weight>
    </member>
  </members>
</pool>
```

• **Default values:**

```
<defaults-values>
  <space lifetime="86400" type="volatile" guarsize="291" totalsize="291"/>
  <file lifetime="3600" type="volatile"/>
```

```
</defaults-values>
```

The mapping rule define how a certain NS-Filesystem, that correspond to a Storage Area in SRM meaning of terms, is exposed in Grid.

```
<mapping-rules>
  <map-rule name="dteam-maprule">
    <stfn-root>/dteam</stfn-root>
    <mapped-fs>dteam-FS</mapped-fs>
  </map-rule>
</mapping-rules>
```

The `<stfn-root>` is the path used to build SURL referring to that Storage Area. The mapping rule above define that the NS-Filesystem named `dteam-FS` have to be mapped in the `/dteam` SURL path. Following the NS-Filesystem element defined in the previpus section, the SURL:

`srm://storm-fe.cr.cnafl.infn.it:8444/dteam/testfile`, following the root expressed in the `dteam-FS` NF-Filesystem element, is mapped in the physical root path on the file system: `/storage/dteam`. This approach works similar to an alias, from the SURL `stfn-root` path to the NS-Filesystem root.

Approachable rules defines which users (or which class of users) can approach a certain Storage Area, always expressed as NS-Filesystem element.

If a user can approach a Storage Area, he can use it for all SRM operations. If a user is not allowed to approach a Storage Area, and he try to specify it in any SRM request, he will receive an `SRM_INVALID_PATH`. In practices, if a user cannot approach a Storage Area, for him that specific path does not exists at all.

Here is an example of approachable rule for the `dteam-FS` element:

```
<approachable-rules>
  <app-rule name="dteam-rule">
    <subjects>
      <dn>*</dn>
      <vo-name>dteam</vo-name>
    </subjects>
    <approachable-fs>dteam-FS</approachable-fs>
    <space-rel-path>/</space-rel-path>
  </app-rule>
</approachable-rules>
```

- `<dn>*</dn>` means that everybody can access the storage Area. Here you can define regular expression on DN fields to define more complex approachable rules.
- `<vo-name>*</vo-name>` means that everybody belonging to a VO access the storage Area. Please note that user without FQANs extension will not be recognized as belonging to VOs then they will not be allowed to approach the SA. Remove or comment this line if the Storage Area have to be open to users without VOMS extensions.
- `<vo-name>dteam</vo-name>` means that only users belonging to the VO `dteam` will be allowed to access the Storage Area. This entry can be a list of comma separated VO-name.

7.2.3 BACKEND STORAGE USAGE INITIALIZATION: *used-space.ini* [STORM VERSION >= 1.8.0]

StoRM maintains the information about of the status of managed storage areas (such as free, used, busy, available, guaranteed and reserved space [ref]), and store them into the DB. Whenever it is consumed or released some

storage space by creating or deleting files, the status is updated and stored in the DB. The storage space status stored into the DB is authoritative. The information about the Storage Space stored into the DB are used also as information source for the Information Provider through the DIP (Dynamic Info Provider).

There are cases in which the status of a storage area must be initialized, for example in the case of a fresh StoRM installation configured to manage a storage space already populated with files, where the space used is not zero.

There are different methods for initialize the Storage Area status, some executed within StoRM (GPFS quota [ref] and/or background-DU [ref]). In this section it is described how an administrator can initialize the status of a Storage Area by editing a configuration file, the *used-space.ini* configuration file, that it will be parsed at bootstrap time and only one time.

The structure of the content of *used-space.ini* is quite simple: a list of sections corresponding to the Storage Area in which are defined the used size, and eventually, the checktime.

For each Storage Area to be initialized there is a section named with the same alias *space-token-description* defined in the namespace.xml, that are defined with YAIM variables `STORM_<SA>_ACCESSPOINT`. Within the section there are two properties: *usedsize* (mandatory) and *checktime* (optional).

Inspecting the properties:

- *usedsize*: The used space in the Storage Area expressed in Bytes. Must be an value without digits after the decimal mark. **MANDATORY**
- *checktime*: The timestamp of the time to wich the *usedsize* computation refers. Must be a date in RFC-2822 format

Here is a sample of *used-space.ini*:

```
[sa-alias-1]
checktime = Fri, 23 Sep 2011 11:56:53 +0200
usedsize = 1848392893847
```

```
[sa-alias-2]
checktime = Fri, 16 Sep 2011 10:22:17 +0200
usedsize = 2839937589367
```

```
[sa-alias-3]
usedsize = 1099511627776
```

This file can be produced in two ways:

- by hand after StoRM BackEnd service configuration
 - write your own *used-space.ini* file adding a section for each Storage Area you want to initialize
 - as section name use the *space-token-description* value as in namespace.xml
 - set the value of *usedsize* property as in the example.
 - set the value of *checktime* property as in the example. To obtain an RFC-2822 timestamp of the current time you can execute the following command

```
date --rfc-2822
```
- by YAIM at StoRM BackEnd service configuration time
 - add a variable `STORM_<SA>_USED_ONLINE_SIZE` to your YAIM configuration file for each Storage Area you want to initialize where `<SA>` is the name of the Storage Area as in `STORM_STORAGEAREA_LIST` YAIM variable (for the value of this variable see paragraph 5.3.1)
 - run YAIM on StoRM profiles installed on this host

StoRM BackEnd will load `used-space.ini` file at bootstrap and initialize the used space of newly created Storage Areas to its values.

NOTE: running YAIM on StoRM BackEnd profile will produce a new `used-space.ini` file and backup any existent version with the extension `.bkp_`. Take this into account if you want to produce the `used-space.ini` file by hand.

7.2.4 LOGGING: LOGGING.XML

The BackEnd logs provide information on the execution process of all SRM requests. The BackEnd log files are all placed in the `/var/log/storm` directory.

BackEnd logging operations are based on the `logback` framework. `logback` provides a way to set the level of verbosity depending on the use case. The level supported are `FATAL`, `ERROR`, `INFO`, `WARN`, `DEBUG`. The `/etc/storm/backend-server/logging.xml` contains this information:

```
<logger name="it.grid.storm" additivity="false">
  <level value="DEBUG" />
  <appender-ref ref="PROCESS" />
</logger>
```

the `value` can be setted to the desired log level.

Please be careful that logging operation can impact on system performance (even 30% slower with `DEBUG` in the worst case). The suggest logging level for production endpoint is `INFO`. In case the log level is modified, the BackEnd have to be restarted to read the new value.

StoRM BackEnd log files:

- **storm-backend.log**

This is the main log file of StoRM Backend. All the information about the SRM execution process, error or warning are logged here depending on the log level. At startup time, the BE logs here all the `storm.properties` value, this can be useful to check value effectively used by the system. After that, the BE logs the result of the `namespace` initialization, reporting errors or misconfiguration.

At the `INFO` level, the BE logs for each SRM operation at least **who** have request the operation (DN and FQANs), on **which** files (SURLs) and the operation **result**. At `DEBUG` level, much more information are printed regarding the status of many StoRM internal component, depending on the SRM request type. `DEBUG` level have to be used carefully only for troubleshooting operation. If `ERROR` or `FATAL` level are used, the only event logged in the file are due to error condition.

- **storm-backend.stdout**

This file contains the standard out of the Backend process. Usually it does not contains any useful information.

- **storm-backend.stderr**

This file contains the event logged as `ERROR` or `FATAL` conditions. This event logs are presents both in the `storm-backend.log` file and here.

StoRM provides a bookkeeping framework that elaborates informations on SRM requests processed by the system to provide user-friendly aggregated data that can be used to get a quick view on system health.

- **heartbeat.log**

This useful file contains information on the SRM requests process by the system from its startup, adding new information at each `beat`. The `beat` time interval can be configured, by default is 60 seconds. At each beat, the hearthbeat component logs an entry like that:

```
[#.....71 lifetime=1:10.01]
  Heap Free:59123488 SYNCH [500] ASynch [PTG:2450 PTP:3422]
  Last:( [#PTG=10 OK=10 M.Dur.=150] [#PTP=5 OK=5 M.Dur.=300] )
```

Log	Meaning
[#.....71	Log entry number
lifetime=1:10.01	Lifetime from last startup, hh:mm:ss
Heap Free:59123488	BE Process free heap size in Bytes
SYNCH [500]	Number of Synchronous SRM requests executed in the last beat
ASynch [PTG:2450 PTP:3422]	Number of srmPrepareToGet and srmPrepareToPut requests executed from start-up.
Last:([#PTG=10 OK=10 M.Dur.=150]	Number of srmPrepareToGet executed in the last beat , with the number of request terminated with success (OK=10) and average time in millisecond (M.Dur.=150)
[#PTP=5 OK=5 M.Dur.=300]	Number of srmPrepareToGet executed in the last beat , with number of request terminated with success and average time in milliseconds.

This log information can be really useful to gain a global view on the overall system status. **A tail on this file is the first thing to do if you want to check the health of your StoRM installation.** From here you can understand if the system is receiving SRM requests or if the system is overloaded by SRM request or if PtG and PtP are running without problem or if the interaction with the filesystem is exceptionally low (in case the M.Dur. is much more than usual).

7.2.5 SPACE AUTHORIZATION: AUTHZ.DB

Space authorization component define access control policy on the Storage Area managed by StoRM. It allows to define rules as: *users* (expressed in terms of regular expression on FQANs or DN), *operation* (READ/WRITE/others) and *target* Storage Area. This rules are stored in a file named `authz.db` located at `/etc/storm/backend-server/`

- **List of operations**

Operation name	Code	Description
RELEASE_SPACE	D	Release Space
UPDATE_SPACE	U	Update Space
READ_FROM_SPACE	R	Read from space
WRITE_TO_SPACE	W	Write to space
STAGE_TO_SPACE	S	Stage in space
REPLICATE_FROM_SPACE	C	replicate from space
PURGE_FROM_SPACE	P	Purge from space
QUERY_SPACE	Q	Query space

- **Subjects**

DN, FQANs, regular expression on them.

- **Space auth DB file**

This file contains all the rule defining access policies for a Storage Area, and it is expressed in the `auth_db` element of the NS-Filesystem element referring to that Storage Area.

```
ace.1=dn:/DC=ch/DC=cern/OU=Organic Units/OU=Users/CN=lagnoni/CN=576235/CN=Luca Magnoni:DURWSCP:AL
ace.2=dn:/O=GermanGrid/OU=DESY/CN=Tigran Mkrtchyan:S:ALLOW
ace.3=fqan:EVERYONE:RQ:ALLOW
ace.4=fqan:EVERYONE:S:DENY
ace.5=fqan:dteam/Role=production:RSWQP:ALLOW
ace.6=fqan:dteam/Role=lcgamin:DURWSPQM:ALLOW
```



```
ace.7=fqan:dteam/Role=NULL:RSQ:ALLOW  
ace.8=fqan:EVERYONE:DURWSPQMC:DENY
```

- **Evaluation algorithm**

The evaluation algorithm is taken from the NFS4 approach.

7.3 GRIDFTP CONFIGURATION

At each transfer request, the GridFTP uses LCMAPS to get user mapping and start a new processes on behalf of the user to proceed with data transfer. GridFTP relies on a different db file to get the plugin to use. Obviously LCMAPS have to answer to GridFTP requests and StoRM requests in coherent way.

- **Configuration files**

The GridFTP uses the LCMAPS configuration file located at `/etc/lcmaps/lcmaps.db`

7.3.1 LOGGING FILES AND LOGGING LEVEL

GridFTP produce two separated log files:

- `/var/log/storm/gridftp-session.log` For the command session information
- `/var/log/storm/globus-gridftp.log` For the transfer logs

The logging level can be specified editing the configuration file `/etc/globus-gridftp-server/gridftp.gfork`
The supported logging levels are: ERROR | WARN | INFO | DUMP | ALL

7.4 GRIDHTTPTS CONFIGURATION

StoRM GridHTTPTS server is the component responsible to provide HTTP(s) file transfer capabilities to a StoRM endpoint. GridHTTPTS server is implemented as a web component residing in Tomcat Servlet container. It interacts with StoRM BackEnd to configure itself. Another GridHTTPTS component, the GridHTTPTS Plugin, is deployed on StoRM BackEnd host and represents a bridge between BackEnd and GridHttps server. GridHTTPTS server interacts with BackEnd to validate file transfer requests. Its configuration is entirely managed by YAIM.

7.4.1 GRIDHTTPTS SERVICE INFORMATION: WEB.XML

The configuration of the service requires the configuration of Tomcat. Tomcat configuration relies on a set files located in `/etc/tomcat5/`, among those there is the `web.xml` file defining how default servlet must behave. Grid-HTTPs leverage on Tomcat default servlet to perform file transfers so a customized `web.xml` file is provided by the service and used to configure Tomcat properly. GridHTTPTS `web.xml` file is located in `/etc/storm/gridhttps-server` directory, there is a link pointing to this file from `/etc/tomcat5/`.

`web.xml` structure looks like this:

```
<web-app xmlns="http://java.sun.com/xml/ns/j2ee" ...>
```

```
[...]
```

```
<servlet>  
  <servlet-name>default</servlet-name>  
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>  
  <init-param>
```

```
<param-name>readonly</param-name>
  <param-value>0</param-value>
</init-param>
[...]
```

```
</servlet>

[...]
```

```
<filter>
  <filter-name>gridhttps-filter</filter-name>
  <filter-class>it.grid.storm.gridhttps.filter.RequestFilter</filter-class>
  <init-param>
    <param-name>server.hostname</param-name>
    <param-value>omii004-vm01.cnaf.infn.it</param-value>
  </init-param>
  <init-param>
    <param-name>server.port</param-name>
    <param-value>9998</param-value>
  </init-param>
  <init-param>
    <param-name>contextDeployFolder</param-name>
    <param-value>/usr/share/tomcat5/conf/Catalina/localhost</param-value>
  </init-param>
</filter>

<filter-mapping>
  <filter-name>gridhttps-filter</filter-name>
  <url-pattern> /* </url-pattern>
</filter-mapping>

[...]
```

```
</web-app>
```

- **<servlet> tag**

The servlet tag contains a behavioural parameter :

- *readonly*

This is a boolean parameter that permits (if value is 0) or forbids (if value is 1) the writing capability of the service. If value is not 0 StoRM will not perform srmPrepareToPut operation using neither http or https protocols.

- **<filter> tag**

The filter tag contains GridHTTPs initialization parameters :

- *server.hostname*

This is a string parameter that contains the full hostname of StoRM BackEnd host.

- *server.port*

This is a numeric parameter that contains the port of StoRM BackEnd REST service. It have to match with StoRM BackEnd configuration parameter `storm.rest.services.port`

- *contextDeployFolder*

This is a string parameter specifying the foldere where the context files are checked by the servlet container. It is not intended to be changed by users. It is related to GridHTTPs compatibility with different Servlet containers

- **<filter-mapping> tag**

The filter-mapping tag specifies which paths should be managed by GridHTTps. His parameter has not to be changed, all paths have to be enabled; user authorization is performed in GridHTTps logic.

In case a parameter is modified, **the Tomcat service have to be restarted in order to read the new value.**

7.4.2 GRIDHTTps PLUGIN INFORMATION: STORM.GRIDHTTps.PLUGIN.PROPERTIES

StoRM GridHTTps Plugin is shipped with StoRM BackEnd metapackage and is installed on BackEnd host. Its configuration information are stored in a file named `storm.gridhttps.plugin.properties` located in `/etc/storm/gridhttps-plugin` directory. This file contains a list of:

- `key = value`

pairs that can be used to configure the GridHTTps Plugin.

The GridHTTps Plugin lives within BackEnd Java process; in case a parameter is modified, **the BackEnd service have to be restarted in order to read the new value.**

Property name	Description
GridHTTps Server settings	
<code>gridhttps.server.user.uid</code>	The User ID associated to the local user running the Gridhttps server service
<code>gridhttps.server.user.gid</code>	The primary Group ID associated to the local user running the Gridhttps server service

7.4.3 LOGGING FILES AND LOGGING LEVEL

GridHTTps produce two log files located in `/var/log/storm/` directory:

- `storm-gridhttps-server.log` For managed requests
- `velocity-gridhttps.log` For the creation of Storage Area context files

The logging level of these files can be specified editing the configuration file `log4j-gridhttps-server.properties` located in `/etc/storm/gridhttps-server/` directory modifying the value of `log4j.logger.it.grid.storm.gridhttps` parameter. The supported logging levels are: FATAL | ERROR | WARN | INFO | DEBUG | TRACE. The logging level must always be followed by `, gridhttpsLogFile`.

The suggest logging level for production endpoint is `INFO`. In case the log level is modified, Tomcat service has to be restarted to read the new value.

Furtermore, residing in Tomcat its execution produces also Tomcat log files located in `texttt/var/log/tomcat5/` directory:

- `catalina.out` Containing Tomcat standard output and standard error
- `tomcat.log` Containing Tomcat execution information

The logging level of these files can be specified editing the configuration file `log4j-tomcat.properties` located in `/etc/storm/gridhttps-server/` directory modifying the value of `log4j.rootLogger` parameter. The supported logging levels are: FATAL | ERROR | WARN | INFO | DEBUG | TRACE. The logging level must always be followed by `, R`.

The suggest logging level for production endpoint is `INFO`. In case the log level is modified, Tomcat service has to be restarted to read the new value.

7.5 STORM EMIR CONFIGURATION

You can use EMIR-SERP to publish StoRM information to EMIR. EMIR_SERP uses the information already available in the resource resource bdii (aka ERIS) and publish it to an EMIR DSR endpoint.

First check that the resource bdii is up and running. Executing

```
ldapsearch -x -h localhost -p 2170 -b 'GLUE2GroupID=resource,o=glue' objectClass=GLUE2Service
```

should return two services for each vo installed on the machine. If this is not the case, there is some problem with your installation.

Then install serp

```
sudo yum install emir-serp
```

and edit the configuration file `/etc/emi/emir-serp/emir-serp.ini`, providing the url for the EMIR DSR and the url for the resource bdii

```
...  
url = http://emitbdsr1.cern.ch:9126  
...  
[servicesFromResourceBDII]  
resource_bdii_url = ldap://localhost:2170/GLUE2GroupID=resource,o=glue  
...
```

You can change the update interval

```
# Period of registration/update messages  
# Mandatory configuration parameter  
# Value is given in hours  
period = 1
```

and the time EMIR will be told to consider the information valid

```
# Time of registration entry validity  
# Mandatory configuration parameter  
# Value is given in hours  
validity = 2
```

You might want to set the logging level to debug the first time you start the service

```
verbosity = debug
```

Start the service

```
sudo service emir-serp start
```

Verify the publication by inspecting the page: <http://emitbdsr1.cern.ch:9126/services> searching for an entity with "Name" attribute equal to StoRM YAIM variable "SITE_NAME"

It is recommended to set back the logging level to error and restart the service.

Stopping emier-serp will cause the entry to be deleted.