

# How to install the StoRM back-end server

## StoRM backend server installation guide

**Author:** Riccardo Murri

**Date:** 2006-03-16

**Revision:** 1.13

Contents

- Release Notes
  - ◆ Caveat 1
  - ◆ Caveat 2
  - ◆ Caveat 3
  - ◆ Caveat 4
  - ◆ Caveat 5
- Install the RPM
  - ◆ Installation from RPM
- StoRM BE operations set-up
  - ◆ Prerequisites
  - ◆ Running the configuration script
- Starting/stopping the StoRM BE server
- Installation directory layout
- Configuration files reference

### Release Notes

The StoRM back-end server (BE, for short), together with the StoRM front-end server (FE, for short) makes up the the StoRM SRMv2 server.

#### Caveat 1

Although the StoRM system was designed to allow for split FE/BE operation and FE replication, currently the only supported configuration puts the FE and BE on one and the same machine.

Therefore, all *the instructions here assume that you have already installed and setup the FE on the same box where you are installing the BE.*

#### Caveat 2

*(added 2005-07-22)* StoRM BE interfaces with the LCMAPS library through the `lcmaps_return_poolindex_without_gsi()` function. This function will map the supplied user credentials *to a pool account only*. Therefore, it is not possible, with the current code, to use StoRM with a certificate that is not mapped to a pool account (such as experiment software manager . . . `sgm` accounts).

*(update 2006-03-15)* The LCMAPS maintainer has removed this limitation, but the new code will be released only with gLite 3.0.0:

[https://savannah.cern.ch/bugs/index.php?func=detailitem&item\\_id=8923](https://savannah.cern.ch/bugs/index.php?func=detailitem&item_id=8923)

### **Caveat 3**

*(added 2005-12-07)* The current StoRM BE code will use a user named "storm" if any error occurs when mapping a Grid user to a local user. This will be removed in future releases, but presently the configuration script will create a local user named "storm".

### **Caveat 4**

*(added 2005-12-08)* The current StoRM BE code uses the programs `/usr/lpp/mmfs/bin/mmputacl` and `/usr/lpp/mmfs/bin/mmgetacl` to get/set ACLs on the GPFS filesystem. The path to these programs is hard-coded, and cannot be changed.

### **Caveat 5**

*(added 2006-03-31)* GPFS can be configured to accept either NFSv4 ACLs or POSIX-like ACLs, via the `-k` option to `mmchfs`; StoRM assumes the filesystem is configured to use POSIX-like ACLs, and does not perform any check that this is the actual setup.

So, if the filesystem is instead setup to accept NFSv4 ACLs, the Just-in-Time ACL mechanism in StoRM will not work.

This issue is tracked at:

[http://www.egrid.it/cgi-bin/bugzilla/show\\_bug.cgi?id=79](http://www.egrid.it/cgi-bin/bugzilla/show_bug.cgi?id=79)

## **Install the RPM**

The StoRM BE comes packaged in a RPM file, that should install cleanly on Scientific Linux CERN 3.0.x and compatible boxes.

The RPM has dependencies on some gLite software (VOMS and LCMAPS); directions are given below on how to install the required software using SLC's standard `apt-get` tool.

Also, the StoRM BE needs a working MySQL RDBMS installation for its operation. You should have already setup and configured the RDBMS during the FE setup, so no directions will be given here on how to install MySQL.

### **Installation from RPM**

1. Add the gLite 1.4 software repository to the system package sources:

```
# add gLite 1.4 repository to APT sources
echo 'rpm http://glitesoft.cern.ch/EGEE/gLite/APT/R1.4/ rhel30 externals Release1.4 update' >> /etc/apt/sources.list

# update package list
apt-get update

# add gLite/Globus libraries to the system library search path
echo /opt/globus/lib >> /etc/ld.so.conf
echo /opt/glite/lib >> /etc/ld.so.conf
echo /opt/glite/lib/modules >> /etc/ld.so.conf
ldconfig
```

2. Add the StoRM software repository to the system package sources:

```
# add StoRM repository to APT sources
echo 'rpm http://egrid-doc.egrid.it/download/storm/rpm . storm' \
  > /etc/apt/sources.list.d/storm.list

# update package list
apt-get update
```

### 3. Install StoRM BE using the apt-get tool:

```
# pull BE rpm and its dependencies
apt-get install storm-backend-server
```

If these steps are successful, the StoRM BE software should have been unpacked in the `/opt/storm` default location. (See section [Installation directory layout](#) for a list of directory contents.)

## **StoRM BE operations set-up**

A script `/opt/storm/sbin/storm-backend-configure` is provided to do the most tedious part of the installation.

### **Prerequisites**

Since the BE server and the FE server need a shared database setup, we assume here that you have already successfully installed the FE server, and that during that installation phase you created the shared databases and the `storm` MySQL user.

In detail, the configuration script assumes that:

- ◆ *a MySQL database server is up and running (it should already have been setup during installation of the FE server);*
- ◆ the FE/BE shared databases have been created;
- ◆ a *MySQL* user named `storm` already exists, and it has access to the FE/BE shared databases;

### **Running the configuration script**

As user `root` run the following command:

```
/opt/storm/sbin/storm-backend-configure
```

This will perform the following actions on your system:

- ◆ create system user and group named `storm` (these names are presently hard-coded in the StoRM BE and cannot be changed.).

By default, the startup script will run when system enters runlevels 2,3,4,5.

- ◆ link and activate the startup script to the default system location:

```
/opt/storm/etc/init.d/storm-backend -> /etc/init.d/storm-backend
/opt/storm/etc/sysconfig/storm-backend -> /etc/sysconfig/storm-backend
```

- ◆ create the BE private MySQL tables, and grant MySQL user `storm` appropriate permissions on it; during this phase you will be asked the `root` MySQL user password.
- ◆ setup some configuration files from the distributed templates; you will be asked for the 'storm' MySQL user password, which you should have set during the FE setup.

The StoRM BE is *not* automatically started after the configuration script is run.

You should edit the namespace .cfg configuration file prior to starting the server; it will not run without a correctly configured namespace .cfg file.

See section [Configuration files reference](#) for a detailed explanation of what goes in each and every StoRM BE config file.

## **Starting/stopping the StoRM BE server**

As user root run the following command:

```
/opt/storm/etc/init.d/storm-backend start
```

to start the StoRM BE, and the command:

```
/opt/storm/etc/init.d/storm-backend stop
```

## **Installation directory layout**

By default, the BE RPM will install all files in the /opt/storm directory tree. The tree is organized as follows:

```
/opt/storm                                     (= $STORM_HOME)
|
+-- sbin/
|
+-- doc/
|
+-- etc/                                       (= $STORM_BE_CONFIGDIR)
|
|   +-- sysconfig/
|   |   |
|   |   |-- storm-backend
|   |
|   |-- init.d/
|   |   |
|   |   |-- storm-backend
|   |
|   |-- lib/
|   |   |
|   |   |-- storm-backend/                 (= $STORM_BE_LIBDIR)
|   |   |   |
|   |   |   |-- storm-backend.jar         (= $STORM_BE_JAR)
|   |   |   |
|   |   |   |-- jars/                     (= $STORM_BE_JARDIR)
```

You may move portions of the tree to another location, provided that you set the appropriate environment variables to point to the new location before calling the StoRM BE startup script etc/init.d/storm-backend.

## **Configuration files reference**

Edit the files in /opt/storm/etc to finish the setup of your StoRM BE installation.

*sysconfig/storm-backend*

Values used in the startup/shutdown script; see the explanatory comments in the file.

*log4j.properties*

Standard Log4J configuration file; by default logs all messages (excluding DEBUG level ones) to file `$(STORM_HOME)/var/log/storm-backend.log`, which is rotated on a daily basis.

See <http://www.vipan.com/htdocs/log4jhelp.html> for a quick guide on how to tweak it to your needs.

*storm.properties*

A file in the common Java .properties syntax; available keys are (listed in almost alphabetic order):

*asynch.picker.db.driver*

Which JDBC driver to use when connecting to the FE/BE communication DB. Only supported value is `com.mysql.jdbc.Driver`.

Default is

`com.mysql.jdbc.Driver`.

*asynch.picker.db.host, asynch.picker.db.port*

Host and port for making connection to the FE/BE communication DB.

Default for both is null, which connects StoRM BE to MySQL on the local host through the default filesystem socket.

*asynch.picker.db.pollingrateinsec [integer]*

Time (in seconds) that BE will wait between two successive polls of the communication DB.

*asynch.picker.db.user,*

*asynch.picker.db.passwd*

User and password for making the connection to the FE/BE communication DB.

These are *mandatory*, there is no default value.

*authorization.sources*

A comma-separated list of `AuthorizationSource` classes to use; when an authorization request is to be performed, they will be invoked in the order given, and results combined using the class

specified by  
`authorization.combining.algorithm`  
(see below) configuration key.

The value is case-sensitive.

Currently available authorization  
sources are:

*DenyAll*

Deny any  
requested  
operation.

*PermitAll*

Permit any  
requested  
operation.

Default value is `DenyAll`.

*authorization.combining.algorithm*

Algorithm to use for combining  
results from different authorization  
sources (see above).

The value is case-sensitive.

Currently available combining  
algorithms are:

*DenyOverrides*

Applies the  
XACML  
"deny-overrides"  
policy-combining  
algorithm  
(Appendix  
C.3 of the  
XACML  
1.0 spec,

<http://www.oasis-open.org/committees/download.php/2406/c>

). In short,  
if *any* of the  
configured  
authorization  
sources  
return a  
*deny*  
decision,  
then overall  
result is  
*deny*.

*FirstProper*

A variation  
on  
FirstApplicable:  
try all the  
authorization  
sources in  
the order  
given, until  
one returns  
a *deny*  
result or  
*permit*  
result.

*FirstApplicable*

Applies the  
XACML  
"First-applicable"  
policy-combining  
algorithm  
(Appendix  
C.3 of the  
XACML  
1.0 spec,

<http://www.oasis-open.org/committees/download.php/2406/c>

)

*PermitOverrides*

Applies the  
XACML  
"permit-overrides"  
policy-combining  
algorithm  
(Appendix  
C.3 of the  
XACML  
1.0 spec,

<http://www.oasis-open.org/committees/download.php/2406/c>

).

In short, if  
*any* of the  
configured  
authorization  
sources  
return a  
*permit*  
decision,  
then overall  
result is  
*permit*.

Default is FirstProper.  
*CleaningInitialDelay* [long integer; default

60]

Initial delay in seconds before starting the cleaning thread.

*CleaningTimeInterval* [*long integer*; *default 3600*]

Cleaning time interval, in seconds.

*DefaultLifeTime* [*long integer*; *default 30*]

Default life time (in seconds), used if client did not specify one.

*GridFTPTransferClient*

Used by Factory invoked in CopyChunk subclasses, to instantiate a GridFTPTransferClient. The String returned specifies the name of the class to instantiate; for now, there are two classes: NaiveGridFTPTransferClient and StubGridFTPTransferClient.

Default is

`it.grid.storm.async.NaiveGridFTPTransferClient.`

*local.surl.hostnames*

Any SURL whose hostname part matches (case-insensitively) one of the names in this list, is considered *local* in srmCopy operation.

*Picker2InitialDelay* [*long integer*; *default 15*]

Initial delay before starting to pick data from the DB.

*persistence.db.vendor*

Only supported value is `mysql`.

*persistence.db.driver*

Hibernate driver package for connecting to the BE-private DB; only supported value is `com.mysql.jdbc.Driver`.

*persistence.db.prefix*, *persistence.db.host*,

*persistence.db.name*

The JDBC URL to pass to Hibernate modules to connect to the BE-private data DB is constructed as the concatenation of these string values.

Only supported value for `.prefix` is `jdbc:mysql:.`

*persistence.db.username*,

*persistence.db.passwd*

Credentials for accessing the BE-private DB.

These are *mandatory*, there is no default value.

*persistence.db.pool*

Whether to use DB connection pooling.

*persistence.db.pool.maxActive*

**Boh?**

*persistence.db.pool.maxWait*

**Boh?**

*SRMClientPutTotalRetryTime* [*long integer*; default 60]

Used by CopyChunk when making a remote *srmPrepareToPut* (Push Mode). It needs it to establish the *totalRetryTime* in seconds to supply to the internal SRMClient. The parameter is passed to the *prepareToPut* functionality.

*SRMClientPutTimeOut* [*long integer*; default 180]

Used by CopyChunk when making a remote *srmPrepareToPut* (Push Mode). The CopyChunk will periodically invoke the *statusOfPutRequest* functionality of the internal SRMClient, for at most the time out interval in seconds returned by this method.

*SRMClientPutSleepTime* [*long integer*, default 5]

Used by CopyChunk when making a remote *srmPrepareToPut* (Push Mode). The CopyChunk will wait the amount of time in seconds returned by this method, before invoking again the *statusOfPutRequest* functionality of the internal SRMClient. That is, it tells the polling interval.

*SRMClient*

Used by Factory invoked in CopyChunk subclasses, to instantiate an SRMClient. The String returned specifies the name of the class to instantiate; for now, there are two classes: *NaiveSRMClient* and *StubSRMClient*.

Default is

*it.grid.storm.asynch.NaiveSRMClient*.

*storm.commandserver.port* [default 4444]

The port the command server socket should be bound to.

***This should not be changed from the default value 4444.***

*synchcall.directoryManager.maxLsEntry*

The maximum number of entries returned in an srmLs() request.

*synchcall.xmlrpc.secureServerPort*

The port the XML-RPC server (used in communication with the FE server) should bind to, for serving *crypted* https:// streams.

***This cannot be changed from 8089,***  
as there is currently some hard-coded dependency on this value in the FE server code.

*synchcall.xmlrpc.unsecureServerPort*

The port the XML-RPC server (used in communication with the FE server) should bind to, for serving unencrypted http:// streams.

***This cannot be changed from 8080,***  
as there is currently some hard-coded dependency on this value in the FE server code.

*wrapper.filesystem.acl.tmpdir*

For interaction with the GPFS ACL management commands, StoRM BE will create temporary files. This is the directory where to create those files in.

It must be writable by the user that runs the StoRM BE.

*namespace.cfg*

This file controls the mapping of SURLs into files on the locally mounted disks, and the mapping of SURLs into TURLs.

Each line of the file consists of 8 space-separated fields. All

System Message: WARNING/2 (<string>, line 518)

Block quote ends without a blank line; unexpected unindent.  
fields are mandatory, and they have no default values.

Example excerpt from *namespace.cfg*:

```
+-----+-----+-----+-----+-----+-----+-----+-----+
|StFNRoot | PFNRoot | VO |JustInTimeACL|MaxLifeTime|SpaceType|
```

/cnaif	/gpfs/cnaif	CNAIF	Y	3600	Permanent
--------	-------------	-------	---	------	-----------

The fields are as follows:

#### StFNRoot, PFNRoot

When mapping URLs to local files, replace the StFNRoot at the beginning of the StFN with the given PFNRoot.

In the example above, StoRM will map:

```
srm://hostname:port/cnaif/*  
--> /gpfs/cnaif/*
```

#### VO

Only users belonging to this VO are allowed SRM operations on the set of files with this StFNRoot.

If a VOMS certificate is used, exhibiting FQANs for more than one VO, then only the first one (main VO) is considered for matching against this field.

#### JustInTimeAcl [Y/N]

If Y, then an appropriate filesystem-level ACL will be added by the StoRM BE server when a file is requested (srmPrepareToGet, srmPrepareToPut, srmCopy), and removed when the operation is finished (srmPutDone, srmReleaseFile, or file lifetime expiration).

If N, then StoRM will not alter any ACLs present on the file, and just assume they are properly set to allow for Grid users access.

#### MaxLifeTime

Maximum time (in seconds) a client is allowed to pin a file under this StFNRoot.

For "volatile"-type files, when the lifetime expired, the file may be automatically removed by the StoRM BE server to free disk space.

#### SpaceType [volatile/permanent]

Files created under this StFNRoot have this space type by default.

#### GridFtp Server

When mapping SURLs to TURLs, use the value of this field as the endpoint hostname in the TURL for the GridFTP file transfer protocol.

#### Port

When mapping SURLs to TURLs, use the value of this field as the endpoint port in the TURL for the GridFTP file transfer protocol.

The following line must always exist:

```
/cnafe /gpfs/cnaf NO_VO Y 3600 Permanent gridftp.cnaf.infn.it 2811
```

For the NON VOMS certificate (identified from BackEnd by 'NO\_VO' field), this information will be used for mapping the user into the specified directory.

*mapping.properties*

***THIS FILE WILL BE REMOVED IN NEXT VERSION OF STORM!!!***

Should be ignored; if it's not, please file a bug.

Please refer to <http://grid-it.cnaf.infn.it/storm/> and to the <http://www.egrid.it/sw/storm> for up-to-date information and latest errata.