

cvmfs at Fermilab: Release Manager's Guide

with notes on the OSG / OASIS implementation

Revised 1/30/2014 S. Fuess

Document Set

This document is part of a set of documents describing how cvmfs is implemented and used at Fermilab. The document set consists of:

- cvmfs at Fermilab : Overview
- cvmfs at Fermilab : User's Guide
- cvmfs at Fermilab : Release Manager's Guide <<< this document
- cvmfs at Fermilab : System Manager's Guide

This document set reflects version 2.1.14 of the cvmfs product.

Note: cvmfs is implemented elsewhere in different modes. Where useful, the differences in other implementations are noted.

References

The CERN VM File System (CernVM-FS, or cvmfs) was, as the name implies, originally developed at CERN for the function of distributing virtual machine (VM) images, and is a project within the larger CernVM toolkit. CernVM-FS was recognized as providing features as a stand-alone network file system. The home page is available at <http://cernvm.cern.ch/portal/filesystem>.

The OSG / OASIS implementation is documented at <https://twiki.opensciencegrid.org/bin/view/Documentation/Release3/InstallCvmfs> and <https://www.opensciencegrid.org/bin/view/ReleaseDocumentation/OasisUpdateMethod>.

Overview

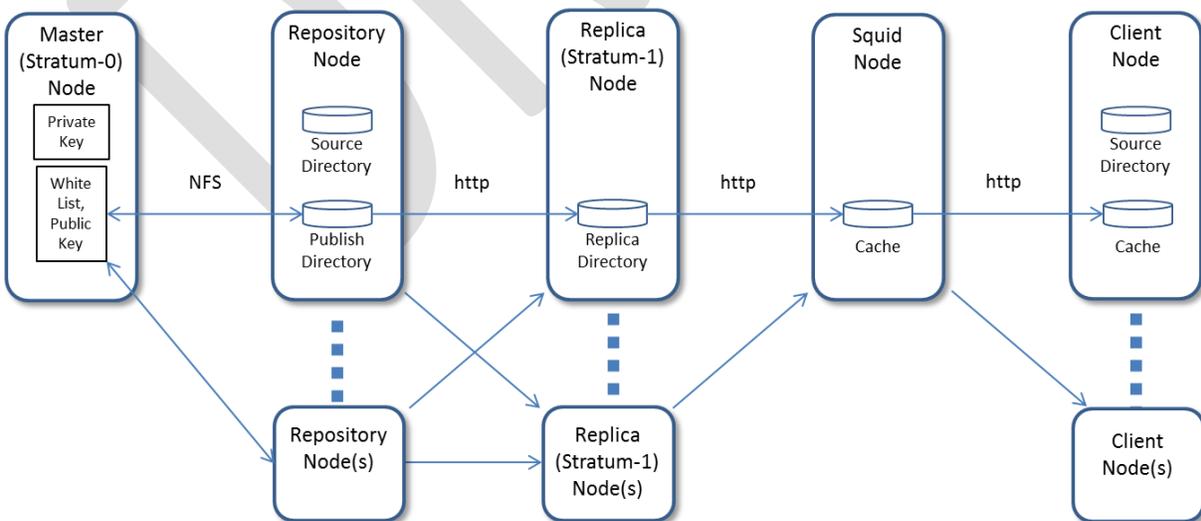


Figure 1 Elements of cvmfs file system

Figure 1 shows the elements within cvmfs and the path from the definitive source directory to the distributed copies of the source directory on the client nodes. Each of these elements is discussed in more detail in the following.

Note: The OSG configuration combines the *Repository* and *Publish* functions within a single node.

Instructions at Fermilab

Creating a Repository

- Request creation of a *Repository* node for the experiment / group. By convention at Fermilab, the *Repository* [name](#) is the same as the FQN *Repository* [node](#) name, and is noted in this document as <repo>.
- Request addition of Kerberos principal to .k5login of cvmfs user on the *Repository* [node](#)

Populating the Repository

- Transfer files to *Repository* [node](#) staging area located at

```
/srv/cvmfs/<repo>/staging
```

Where <repo> is the [name](#) of the repository.

- Example method of transferring files:
 - Generate a tar file elsewhere

```
tar -czf <source.tgz> /my_release_directory
```

- Copy the tar file, for example:

```
kinit <release manager principal>
```

```
scp <source.tgz> cvmfs@<repo>:/srv/cvmfs/<repo>/staging/<source.tgz>
```

- Extract files from staging area into *Repository* source area
 - Example method of copying files:

```
tar -zxf /srv/cvmfs/<repo>/staging/<source>.tgz /cvmfs/<repo>/<my_release_directory>
```

Publishing the Release

- Log in to the *Repository* [node](#) as user cvmfs, execute

```
cvmfs_publish
```

Note: [cvmfs_publish](#) is a Fermilab supplied script which performs the same action as an earlier version (2.0.8) of [cvmfs_server publish](#). Recent (2.1.14) versions of the [cvmfs_server](#) script provide additional options, for example to create named snapshots or discard changes before publishing.

The *published* directory is RW NFS exported to the *Master (stratum-0)* node which maintains a signed whitelist within. The *Repository* node makes the *published* directory available to *Replica (Stratum-1)* servers via a web server.

A cron script on connected *Replica (Stratum-1)* servers executes every 30 minutes to replicate the *published* files. The replication action can also be initiated manually by an administrator (as only root access is allowed to the *Replica* servers).

Instructions for OASIS

The *Publish* step is queued from the login node by the `osg-oasis-update` command. See the OSG / OASIS documentation at <https://www.opensciencegrid.org/bin/view/ReleaseDocumentation/OasisUpdateMethod>.

Fermilab or OASIS Methods : A Discussion

The Fermilab and the OSG / OASIS methods offer alternatives for locating the *Repository* containing the source of the distribution of files. The differences can be hidden from a typical user by the proper configuration of the *Client*. There are some philosophical and operational differences that are noteworthy:

- The Fermilab implementation offers an independent *Repository node* for each experiment's / user group's access and control. Access to the node is controlled by the usual Kerberos mechanism. The OASIS implementation utilizes a single login node shared between experiments / user groups. Access to the node is via `gsissh` utilizing a proxy registered with OSG.
- The Fermilab implementation allows independent control of the *Publish* step, which generates the condensed set of files to be distributed from the original source files. The timing of the *Publish* step is the choice of the experiment / user group owning the *Repository*. With the OASIS implementation, each experiment / user group does an independent *Publish* step, but only one such *Publish* can be active at any time – there is a locking mechanism preventing overlapping *Publish* operations.
- The OASIS implementation is meant to serve all of OSG. Hence the distribution of configuration files and security keys becomes part of the OSG distribution set. The Fermilab configuration parameters and security keys are not part of a wider distribution. If software distributions are meant to be used outside of Fermilab, using the OSG service solves the issue of distributing configuration and keys.
- The “split” or “merge” of multiple (Fermilab, OASIS) *cvms* distributions can occur at several spots in the *cvms* chain:
 - Fermilab and OASIS could maintain independent *Publish (Stratum-0)* servers, with a merging at any *Replica (Stratum-1)* servers.
 - Fermilab and OASIS could maintain independent *Replica (Stratum-1)* servers, with a merging at the *Squid Proxy* servers local to a site.

The principal reason for the creation of an independent Fermilab implementation is that the use of *cvms* in the Fermilab environment is new, and *cvms* is a young product. It is thought that knowledge of the usage patterns and the behavior of *cvms* components is better studied in an environment completely under the control of Fermilab system managers. The future *cvms* implementations may see some level of merging of the two methods.